

# NVIDIA Performance Primitives (NPP)

Version 4.0

October 12, 2011



# Contents

<b>1</b>	<b>NVIDIA Performance Primitives</b>	<b>1</b>
1.1	What is NPP? . . . . .	1
1.2	Documentation . . . . .	1
1.3	Technical Specifications . . . . .	1
1.4	Files . . . . .	2
1.4.1	Header Files . . . . .	2
1.4.2	Library Files . . . . .	2
1.5	Supported NVIDIA Hardware . . . . .	2
<b>2</b>	<b>General API Conventions</b>	<b>3</b>
2.1	Memory Management . . . . .	4
2.1.1	Scratch Buffer and Host Pointer . . . . .	4
2.2	Function Naming . . . . .	4
2.3	Integer Result Scaling . . . . .	5
<b>3</b>	<b>Signal-Processing Specific API Conventions</b>	<b>7</b>
3.1	Signal Data . . . . .	8
3.1.1	Parameter Names for Signal Data . . . . .	8
3.1.1.1	Source Signal Pointer . . . . .	8
3.1.1.2	Destination Signal Pointer . . . . .	8
3.1.1.3	In-Place Signal Pointer . . . . .	8
3.1.2	Signal Data Alignment Requirements . . . . .	9
3.1.3	Signal Data Related Error Codes . . . . .	9
3.2	Signal Length . . . . .	9
3.2.1	Length Related Error Codes . . . . .	9
<b>4</b>	<b>Imaging-Processing Specific API Conventions</b>	<b>11</b>
4.1	Function Naming . . . . .	12
4.2	Image Data . . . . .	12

4.2.1	Line Step	13
4.2.2	Parameter Names for Image Data	13
4.2.2.1	Passing Source-Image Data	13
4.2.2.2	Passing Destination-Image Data	14
4.2.2.3	Passing In-Place Image Data	14
4.2.2.4	Passing Mask-Image Data	14
4.2.3	Image Data Alignment Requirements	15
4.2.4	Image Data Related Error Codes	15
4.3	Region-of-Interest (ROI)	15
4.3.1	ROI Related Error Codes	16
4.4	Masked Operation	16
4.5	Channel-of-Interest API	16
4.5.1	Select-Channel Source-Image Pointer	16
4.5.2	Select-Channel Destination-Image Pointer	17
4.6	Geometric Transform API Specifics	17
4.6.1	Geometric Transforms and ROIs	17
4.6.2	Pixel Interpolation	17
4.6.3	Rotate specific Error Codes	18
<b>5</b>	<b>Module Index</b>	<b>19</b>
5.1	Modules	19
<b>6</b>	<b>Data Structure Index</b>	<b>21</b>
6.1	Data Structures	21
<b>7</b>	<b>Module Documentation</b>	<b>23</b>
7.1	NPP Core	23
7.1.1	Detailed Description	23
7.1.2	Function Documentation	24
7.1.2.1	nppGetGpuComputeCapability	24
7.1.2.2	nppGetGpuName	24
7.1.2.3	nppGetGpuNumSMs	24
7.1.2.4	nppGetLibVersion	24
7.1.2.5	nppGetMaxThreadsPerBlock	24
7.1.2.6	nppGetMaxThreadsPerSM	25
7.1.2.7	nppGetStream	25
7.1.2.8	nppSetStream	25
7.2	NPP Type Definitions and Constants	26



7.2.1	Define Documentation	29
7.2.1.1	NPP_MAX_16S	29
7.2.1.2	NPP_MAX_16U	29
7.2.1.3	NPP_MAX_32S	30
7.2.1.4	NPP_MAX_32U	30
7.2.1.5	NPP_MAX_64S	30
7.2.1.6	NPP_MAX_64U	30
7.2.1.7	NPP_MAX_8S	30
7.2.1.8	NPP_MAX_8U	30
7.2.1.9	NPP_MAXABS_32F	30
7.2.1.10	NPP_MAXABS_64F	30
7.2.1.11	NPP_MIN_16S	30
7.2.1.12	NPP_MIN_16U	30
7.2.1.13	NPP_MIN_32S	30
7.2.1.14	NPP_MIN_32U	31
7.2.1.15	NPP_MIN_64S	31
7.2.1.16	NPP_MIN_64U	31
7.2.1.17	NPP_MIN_8S	31
7.2.1.18	NPP_MIN_8U	31
7.2.1.19	NPP_MINABS_32F	31
7.2.1.20	NPP_MINABS_64F	31
7.2.2	Enumeration Type Documentation	31
7.2.2.1	NppCmpOp	31
7.2.2.2	NppGpuComputeCapability	31
7.2.2.3	NppHintAlgorithm	32
7.2.2.4	NppiAxis	32
7.2.2.5	NppiBorderType	32
7.2.2.6	NppiInterpolationMode	32
7.2.2.7	NppRoundMode	33
7.2.2.8	NppStatus	33
7.3	Basic NPP Data Types	35
7.3.1	Typedef Documentation	36
7.3.1.1	Npp16s	36
7.3.1.2	Npp16u	36
7.3.1.3	Npp32f	36
7.3.1.4	Npp32s	36

7.3.1.5	Npp32u	36
7.3.1.6	Npp64f	36
7.3.1.7	Npp64s	36
7.3.1.8	Npp64u	36
7.3.1.9	Npp8s	36
7.3.1.10	Npp8u	36
7.4	NPP Image Processing	37
7.4.1	Typedef Documentation	48
7.4.1.1	NppiGraphcutState	48
7.4.2	Enumeration Type Documentation	48
7.4.2.1	NppiAlphaOp	48
7.4.3	Function Documentation	48
7.4.3.1	nppiColorTwist32f_8u_AC4R	48
7.4.3.2	nppiColorTwist32f_8u_C3R	49
7.4.3.3	nppiColorTwist32f_8u_P3R	49
7.4.3.4	nppiDilate_8u_C1R	50
7.4.3.5	nppiDilate_8u_C4R	50
7.4.3.6	nppiErode_8u_C1R	51
7.4.3.7	nppiErode_8u_C4R	51
7.4.3.8	nppiEvenLevelsHost_32s	52
7.4.3.9	nppiFilter_8u_C1R	52
7.4.3.10	nppiFilter_8u_C4R	52
7.4.3.11	nppiFilterBox_8u_C1R	53
7.4.3.12	nppiFilterBox_8u_C4R	53
7.4.3.13	nppiFilterColumn_8u_C1R	54
7.4.3.14	nppiFilterColumn_8u_C4R	54
7.4.3.15	nppiFilterMax_8u_C1R	55
7.4.3.16	nppiFilterMax_8u_C4R	55
7.4.3.17	nppiFilterMin_8u_C1R	56
7.4.3.18	nppiFilterMin_8u_C4R	56
7.4.3.19	nppiFilterRow_8u_C1R	57
7.4.3.20	nppiFilterRow_8u_C4R	57
7.4.3.21	nppiGraphcut8_32s8u	58
7.4.3.22	nppiGraphcut8GetSize	59
7.4.3.23	nppiGraphcut8InitAlloc	59
7.4.3.24	nppiGraphcut_32s8u	59

7.4.3.25	<a href="#">nppiGraphcutFree</a>	60
7.4.3.26	<a href="#">nppiGraphcutGetSize</a>	61
7.4.3.27	<a href="#">nppiGraphcutInitAlloc</a>	61
7.4.3.28	<a href="#">nppiHistogramEven_16s_AC4R</a>	61
7.4.3.29	<a href="#">nppiHistogramEven_16s_C1R</a>	62
7.4.3.30	<a href="#">nppiHistogramEven_16s_C4R</a>	62
7.4.3.31	<a href="#">nppiHistogramEven_16u_AC4R</a>	63
7.4.3.32	<a href="#">nppiHistogramEven_16u_C1R</a>	63
7.4.3.33	<a href="#">nppiHistogramEven_16u_C4R</a>	64
7.4.3.34	<a href="#">nppiHistogramEven_8u_AC4R</a>	64
7.4.3.35	<a href="#">nppiHistogramEven_8u_C1R</a>	65
7.4.3.36	<a href="#">nppiHistogramEven_8u_C4R</a>	65
7.4.3.37	<a href="#">nppiHistogramEvenGetBufferSize_16s_AC4R</a>	65
7.4.3.38	<a href="#">nppiHistogramEvenGetBufferSize_16s_C1R</a>	66
7.4.3.39	<a href="#">nppiHistogramEvenGetBufferSize_16s_C4R</a>	66
7.4.3.40	<a href="#">nppiHistogramEvenGetBufferSize_16u_AC4R</a>	66
7.4.3.41	<a href="#">nppiHistogramEvenGetBufferSize_16u_C1R</a>	67
7.4.3.42	<a href="#">nppiHistogramEvenGetBufferSize_16u_C4R</a>	67
7.4.3.43	<a href="#">nppiHistogramEvenGetBufferSize_8u_AC4R</a>	67
7.4.3.44	<a href="#">nppiHistogramEvenGetBufferSize_8u_C1R</a>	67
7.4.3.45	<a href="#">nppiHistogramEvenGetBufferSize_8u_C4R</a>	68
7.4.3.46	<a href="#">nppiHistogramRange_16s_AC4R</a>	68
7.4.3.47	<a href="#">nppiHistogramRange_16s_C1R</a>	69
7.4.3.48	<a href="#">nppiHistogramRange_16s_C4R</a>	69
7.4.3.49	<a href="#">nppiHistogramRange_16u_AC4R</a>	69
7.4.3.50	<a href="#">nppiHistogramRange_16u_C1R</a>	70
7.4.3.51	<a href="#">nppiHistogramRange_16u_C4R</a>	70
7.4.3.52	<a href="#">nppiHistogramRange_32f_AC4R</a>	71
7.4.3.53	<a href="#">nppiHistogramRange_32f_C1R</a>	71
7.4.3.54	<a href="#">nppiHistogramRange_32f_C4R</a>	72
7.4.3.55	<a href="#">nppiHistogramRange_8u_AC4R</a>	72
7.4.3.56	<a href="#">nppiHistogramRange_8u_C1R</a>	73
7.4.3.57	<a href="#">nppiHistogramRange_8u_C4R</a>	73
7.4.3.58	<a href="#">nppiHistogramRangeGetBufferSize_16s_AC4R</a>	73
7.4.3.59	<a href="#">nppiHistogramRangeGetBufferSize_16s_C1R</a>	74
7.4.3.60	<a href="#">nppiHistogramRangeGetBufferSize_16s_C4R</a>	74

7.4.3.61	<a href="#">nppiHistogramRangeGetBufferSize_16u_AC4R</a>	74
7.4.3.62	<a href="#">nppiHistogramRangeGetBufferSize_16u_C1R</a>	75
7.4.3.63	<a href="#">nppiHistogramRangeGetBufferSize_16u_C4R</a>	75
7.4.3.64	<a href="#">nppiHistogramRangeGetBufferSize_32f_AC4R</a>	75
7.4.3.65	<a href="#">nppiHistogramRangeGetBufferSize_32f_C1R</a>	75
7.4.3.66	<a href="#">nppiHistogramRangeGetBufferSize_32f_C4R</a>	76
7.4.3.67	<a href="#">nppiHistogramRangeGetBufferSize_8u_AC4R</a>	76
7.4.3.68	<a href="#">nppiHistogramRangeGetBufferSize_8u_C1R</a>	76
7.4.3.69	<a href="#">nppiHistogramRangeGetBufferSize_8u_C4R</a>	77
7.4.3.70	<a href="#">nppiLUT_Linear_8u_AC4R</a>	77
7.4.3.71	<a href="#">nppiLUT_Linear_8u_C1R</a>	77
7.4.3.72	<a href="#">nppiLUT_Linear_8u_C3R</a>	78
7.4.3.73	<a href="#">nppiMagnitude_32fc32f_C1R</a>	78
7.4.3.74	<a href="#">nppiMagnitudeSqr_32fc32f_C1R</a>	79
7.4.3.75	<a href="#">nppiMean_StdDev_8u_C1R</a>	79
7.4.3.76	<a href="#">nppiMeanStdDev8uC1RGetBufferHostSize</a>	80
7.4.3.77	<a href="#">nppiMinMax_8u_C1R</a>	80
7.4.3.78	<a href="#">nppiMinMax_8u_C4R</a>	80
7.4.3.79	<a href="#">nppiMinMaxGetBufferSize_8u_C1R</a>	81
7.4.3.80	<a href="#">nppiMinMaxGetBufferSize_8u_C4R</a>	81
7.4.3.81	<a href="#">nppiNormDiff_Inf_8u_C1R</a>	81
7.4.3.82	<a href="#">nppiNormDiff_L1_8u_C1R</a>	82
7.4.3.83	<a href="#">nppiNormDiff_L2_8u_C1R</a>	82
7.4.3.84	<a href="#">nppiRectStdDev_32s32f_C1R</a>	82
7.4.3.85	<a href="#">nppiReductionGetBufferHostSize_8u_C1R</a>	83
7.4.3.86	<a href="#">nppiReductionGetBufferHostSize_8u_C4R</a>	83
7.4.3.87	<a href="#">nppiRGBToYCbCr420_8u_C3P3R</a>	83
7.4.3.88	<a href="#">nppiRGBToYCbCr422_8u_C3C2R</a>	84
7.4.3.89	<a href="#">nppiRGBToYCbCr_8u_AC4R</a>	84
7.4.3.90	<a href="#">nppiRGBToYCbCr_8u_C3R</a>	84
7.4.3.91	<a href="#">nppiRGBToYCbCr_8u_P3R</a>	85
7.4.3.92	<a href="#">nppiSqrIntegral_8u32s32f_C1R</a>	85
7.4.3.93	<a href="#">nppiSum_8u64s_C1R</a>	85
7.4.3.94	<a href="#">nppiSum_8u64s_C4R</a>	86
7.4.3.95	<a href="#">nppiSum_8u_C1R</a>	86
7.4.3.96	<a href="#">nppiSum_8u_C4R</a>	87

7.4.3.97	<a href="#">nppiSumWindowColumn_8u32f_C1R</a>	87
7.4.3.98	<a href="#">nppiSumWindowRow_8u32f_C1R</a>	87
7.4.3.99	<a href="#">nppiYCbCr420ToRGB_8u_P3C3R</a>	88
7.4.3.100	<a href="#">nppiYCbCr420ToYCbCr411_8u_P3P2R</a>	88
7.4.3.101	<a href="#">nppiYCbCr420ToYCbCr422_8u_P3R</a>	89
7.4.3.102	<a href="#">nppiYCbCr422ToRGB_8u_C2C3R</a>	89
7.4.3.103	<a href="#">nppiYCbCr422ToYCbCr411_8u_P3R</a>	89
7.4.3.104	<a href="#">nppiYCbCr422ToYCbCr420_8u_P3R</a>	90
7.4.3.105	<a href="#">nppiYCbCrToRGB_8u_AC4R</a>	90
7.4.3.106	<a href="#">nppiYCbCrToRGB_8u_C3R</a>	90
7.4.3.107	<a href="#">nppiYCbCrToRGB_8u_P3R</a>	91
7.5	Memory Management	92
7.5.1	Detailed Description	94
7.5.2	Function Documentation	94
7.5.2.1	<a href="#">nppiFree</a>	94
7.5.2.2	<a href="#">nppiMalloc_16s_C1</a>	94
7.5.2.3	<a href="#">nppiMalloc_16s_C2</a>	95
7.5.2.4	<a href="#">nppiMalloc_16s_C4</a>	95
7.5.2.5	<a href="#">nppiMalloc_16sc_C1</a>	95
7.5.2.6	<a href="#">nppiMalloc_16sc_C2</a>	95
7.5.2.7	<a href="#">nppiMalloc_16sc_C3</a>	96
7.5.2.8	<a href="#">nppiMalloc_16sc_C4</a>	96
7.5.2.9	<a href="#">nppiMalloc_16u_C1</a>	96
7.5.2.10	<a href="#">nppiMalloc_16u_C2</a>	97
7.5.2.11	<a href="#">nppiMalloc_16u_C3</a>	97
7.5.2.12	<a href="#">nppiMalloc_16u_C4</a>	97
7.5.2.13	<a href="#">nppiMalloc_32f_C1</a>	97
7.5.2.14	<a href="#">nppiMalloc_32f_C2</a>	98
7.5.2.15	<a href="#">nppiMalloc_32f_C3</a>	98
7.5.2.16	<a href="#">nppiMalloc_32f_C4</a>	98
7.5.2.17	<a href="#">nppiMalloc_32fc_C1</a>	99
7.5.2.18	<a href="#">nppiMalloc_32fc_C2</a>	99
7.5.2.19	<a href="#">nppiMalloc_32fc_C3</a>	99
7.5.2.20	<a href="#">nppiMalloc_32fc_C4</a>	99
7.5.2.21	<a href="#">nppiMalloc_32s_C1</a>	100
7.5.2.22	<a href="#">nppiMalloc_32s_C3</a>	100

7.5.2.23	<code>nppiMalloc_32s_C4</code>	100
7.5.2.24	<code>nppiMalloc_32sc_C1</code>	101
7.5.2.25	<code>nppiMalloc_32sc_C2</code>	101
7.5.2.26	<code>nppiMalloc_32sc_C3</code>	101
7.5.2.27	<code>nppiMalloc_32sc_C4</code>	101
7.5.2.28	<code>nppiMalloc_8u_C1</code>	102
7.5.2.29	<code>nppiMalloc_8u_C2</code>	102
7.5.2.30	<code>nppiMalloc_8u_C3</code>	102
7.5.2.31	<code>nppiMalloc_8u_C4</code>	103
7.6	Data-Exchange and Initialization	104
7.6.1	Detailed Description	119
7.6.2	Function Documentation	120
7.6.2.1	<code>nppiConvert_16s32f_C1R</code>	120
7.6.2.2	<code>nppiConvert_16s32s_C1R</code>	120
7.6.2.3	<code>nppiConvert_16s8u_AC4R</code>	120
7.6.2.4	<code>nppiConvert_16s8u_C1R</code>	121
7.6.2.5	<code>nppiConvert_16s8u_C4R</code>	121
7.6.2.6	<code>nppiConvert_16u32f_C1R</code>	121
7.6.2.7	<code>nppiConvert_16u32s_C1R</code>	122
7.6.2.8	<code>nppiConvert_16u8u_AC4R</code>	122
7.6.2.9	<code>nppiConvert_16u8u_C1R</code>	123
7.6.2.10	<code>nppiConvert_16u8u_C4R</code>	123
7.6.2.11	<code>nppiConvert_32f16s_C1R</code>	123
7.6.2.12	<code>nppiConvert_32f16u_C1R</code>	124
7.6.2.13	<code>nppiConvert_32f8u_C1R</code>	124
7.6.2.14	<code>nppiConvert_8u16s_AC4R</code>	125
7.6.2.15	<code>nppiConvert_8u16s_C1R</code>	125
7.6.2.16	<code>nppiConvert_8u16s_C4R</code>	125
7.6.2.17	<code>nppiConvert_8u16u_AC4R</code>	126
7.6.2.18	<code>nppiConvert_8u16u_C1R</code>	126
7.6.2.19	<code>nppiConvert_8u16u_C4R</code>	126
7.6.2.20	<code>nppiConvert_8u32f_C1R</code>	127
7.6.2.21	<code>nppiCopy_16s_AC4MR</code>	127
7.6.2.22	<code>nppiCopy_16s_AC4R</code>	128
7.6.2.23	<code>nppiCopy_16s_C1C3R</code>	128
7.6.2.24	<code>nppiCopy_16s_C1C4R</code>	128

7.6.2.25	nppiCopy_16s_C1MR . . . . .	129
7.6.2.26	nppiCopy_16s_C1R . . . . .	129
7.6.2.27	nppiCopy_16s_C3C1R . . . . .	129
7.6.2.28	nppiCopy_16s_C3CR . . . . .	130
7.6.2.29	nppiCopy_16s_C3MR . . . . .	130
7.6.2.30	nppiCopy_16s_C3P3R . . . . .	130
7.6.2.31	nppiCopy_16s_C4C1R . . . . .	131
7.6.2.32	nppiCopy_16s_C4CR . . . . .	131
7.6.2.33	nppiCopy_16s_C4MR . . . . .	131
7.6.2.34	nppiCopy_16s_C4P4R . . . . .	132
7.6.2.35	nppiCopy_16s_C4R . . . . .	132
7.6.2.36	nppiCopy_16s_P3C3R . . . . .	132
7.6.2.37	nppiCopy_16s_P4C4R . . . . .	133
7.6.2.38	nppiCopy_16sc_AC4R . . . . .	133
7.6.2.39	nppiCopy_16sc_C1R . . . . .	133
7.6.2.40	nppiCopy_16sc_C2R . . . . .	134
7.6.2.41	nppiCopy_16sc_C3R . . . . .	134
7.6.2.42	nppiCopy_16sc_C4R . . . . .	134
7.6.2.43	nppiCopy_16u_AC4MR . . . . .	135
7.6.2.44	nppiCopy_16u_AC4R . . . . .	135
7.6.2.45	nppiCopy_16u_C1C3R . . . . .	135
7.6.2.46	nppiCopy_16u_C1C4R . . . . .	136
7.6.2.47	nppiCopy_16u_C1MR . . . . .	136
7.6.2.48	nppiCopy_16u_C1R . . . . .	136
7.6.2.49	nppiCopy_16u_C3C1R . . . . .	137
7.6.2.50	nppiCopy_16u_C3CR . . . . .	137
7.6.2.51	nppiCopy_16u_C3MR . . . . .	137
7.6.2.52	nppiCopy_16u_C3P3R . . . . .	138
7.6.2.53	nppiCopy_16u_C4C1R . . . . .	138
7.6.2.54	nppiCopy_16u_C4CR . . . . .	138
7.6.2.55	nppiCopy_16u_C4MR . . . . .	139
7.6.2.56	nppiCopy_16u_C4P4R . . . . .	139
7.6.2.57	nppiCopy_16u_C4R . . . . .	139
7.6.2.58	nppiCopy_16u_P3C3R . . . . .	140
7.6.2.59	nppiCopy_16u_P4C4R . . . . .	140
7.6.2.60	nppiCopy_32f_AC4MR . . . . .	140

7.6.2.61	<a href="#">nppiCopy_32f_AC4R</a>	141
7.6.2.62	<a href="#">nppiCopy_32f_C1C3R</a>	141
7.6.2.63	<a href="#">nppiCopy_32f_C1C4R</a>	141
7.6.2.64	<a href="#">nppiCopy_32f_C1MR</a>	142
7.6.2.65	<a href="#">nppiCopy_32f_C1R</a>	142
7.6.2.66	<a href="#">nppiCopy_32f_C3C1R</a>	142
7.6.2.67	<a href="#">nppiCopy_32f_C3CR</a>	143
7.6.2.68	<a href="#">nppiCopy_32f_C3MR</a>	143
7.6.2.69	<a href="#">nppiCopy_32f_C3P3R</a>	143
7.6.2.70	<a href="#">nppiCopy_32f_C4C1R</a>	144
7.6.2.71	<a href="#">nppiCopy_32f_C4CR</a>	144
7.6.2.72	<a href="#">nppiCopy_32f_C4MR</a>	144
7.6.2.73	<a href="#">nppiCopy_32f_C4P4R</a>	145
7.6.2.74	<a href="#">nppiCopy_32f_C4R</a>	145
7.6.2.75	<a href="#">nppiCopy_32f_P3C3R</a>	145
7.6.2.76	<a href="#">nppiCopy_32f_P4C4R</a>	146
7.6.2.77	<a href="#">nppiCopy_32fc_AC4R</a>	146
7.6.2.78	<a href="#">nppiCopy_32fc_C1R</a>	146
7.6.2.79	<a href="#">nppiCopy_32fc_C2R</a>	147
7.6.2.80	<a href="#">nppiCopy_32fc_C3R</a>	147
7.6.2.81	<a href="#">nppiCopy_32fc_C4R</a>	147
7.6.2.82	<a href="#">nppiCopy_32s_AC4MR</a>	148
7.6.2.83	<a href="#">nppiCopy_32s_AC4R</a>	148
7.6.2.84	<a href="#">nppiCopy_32s_C1C3R</a>	148
7.6.2.85	<a href="#">nppiCopy_32s_C1C4R</a>	149
7.6.2.86	<a href="#">nppiCopy_32s_C1MR</a>	149
7.6.2.87	<a href="#">nppiCopy_32s_C1R</a>	149
7.6.2.88	<a href="#">nppiCopy_32s_C3C1R</a>	150
7.6.2.89	<a href="#">nppiCopy_32s_C3CR</a>	150
7.6.2.90	<a href="#">nppiCopy_32s_C3MR</a>	150
7.6.2.91	<a href="#">nppiCopy_32s_C3P3R</a>	151
7.6.2.92	<a href="#">nppiCopy_32s_C4C1R</a>	151
7.6.2.93	<a href="#">nppiCopy_32s_C4CR</a>	151
7.6.2.94	<a href="#">nppiCopy_32s_C4MR</a>	152
7.6.2.95	<a href="#">nppiCopy_32s_C4P4R</a>	152
7.6.2.96	<a href="#">nppiCopy_32s_C4R</a>	152



7.6.2.97	nppiCopy_32s_P3C3R	153
7.6.2.98	nppiCopy_32s_P4C4R	153
7.6.2.99	nppiCopy_32sc_AC4R	153
7.6.2.100	nppiCopy_32sc_C1R	154
7.6.2.101	nppiCopy_32sc_C2R	154
7.6.2.102	nppiCopy_32sc_C3R	154
7.6.2.103	nppiCopy_32sc_C4R	155
7.6.2.104	nppiCopy_8s_AC4R	155
7.6.2.105	nppiCopy_8s_C1R	155
7.6.2.106	nppiCopy_8s_C2R	156
7.6.2.107	nppiCopy_8s_C3R	156
7.6.2.108	nppiCopy_8s_C4R	156
7.6.2.109	nppiCopy_8u_AC4MR	157
7.6.2.110	nppiCopy_8u_AC4R	157
7.6.2.111	nppiCopy_8u_C1C3R	157
7.6.2.112	nppiCopy_8u_C1C4R	158
7.6.2.113	nppiCopy_8u_C1MR	158
7.6.2.114	nppiCopy_8u_C1R	158
7.6.2.115	nppiCopy_8u_C3C1R	159
7.6.2.116	nppiCopy_8u_C3CR	159
7.6.2.117	nppiCopy_8u_C3MR	159
7.6.2.118	nppiCopy_8u_C3P3R	160
7.6.2.119	nppiCopy_8u_C4C1R	160
7.6.2.120	nppiCopy_8u_C4CR	160
7.6.2.121	nppiCopy_8u_C4MR	161
7.6.2.122	nppiCopy_8u_C4P4R	161
7.6.2.123	nppiCopy_8u_C4R	161
7.6.2.124	nppiCopy_8u_P3C3R	162
7.6.2.125	nppiCopy_8u_P4C4R	162
7.6.2.126	nppiCopyConstBorder_32s_C1R	162
7.6.2.127	nppiCopyConstBorder_8u_AC4R	163
7.6.2.128	nppiCopyConstBorder_8u_C1R	163
7.6.2.129	nppiCopyConstBorder_8u_C4R	164
7.6.2.130	nppiSet_16s_AC4MR	164
7.6.2.131	nppiSet_16s_AC4R	165
7.6.2.132	nppiSet_16s_C1MR	165

7.6.2.133 nppiSet_16s_C1R . . . . .	165
7.6.2.134 nppiSet_16s_C2R . . . . .	166
7.6.2.135 nppiSet_16s_C4CR . . . . .	166
7.6.2.136 nppiSet_16s_C4MR . . . . .	167
7.6.2.137 nppiSet_16s_C4R . . . . .	167
7.6.2.138 nppiSet_16sc_AC4R . . . . .	167
7.6.2.139 nppiSet_16sc_C1R . . . . .	168
7.6.2.140 nppiSet_16sc_C2R . . . . .	168
7.6.2.141 nppiSet_16sc_C3R . . . . .	168
7.6.2.142 nppiSet_16sc_C4R . . . . .	169
7.6.2.143 nppiSet_16u_AC4MR . . . . .	169
7.6.2.144 nppiSet_16u_AC4R . . . . .	169
7.6.2.145 nppiSet_16u_C1MR . . . . .	170
7.6.2.146 nppiSet_16u_C1R . . . . .	170
7.6.2.147 nppiSet_16u_C2R . . . . .	170
7.6.2.148 nppiSet_16u_C4CR . . . . .	171
7.6.2.149 nppiSet_16u_C4MR . . . . .	171
7.6.2.150 nppiSet_16u_C4R . . . . .	171
7.6.2.151 nppiSet_32f_AC4MR . . . . .	172
7.6.2.152 nppiSet_32f_AC4R . . . . .	172
7.6.2.153 nppiSet_32f_C1MR . . . . .	173
7.6.2.154 nppiSet_32f_C1R . . . . .	173
7.6.2.155 nppiSet_32f_C4CR . . . . .	173
7.6.2.156 nppiSet_32f_C4MR . . . . .	174
7.6.2.157 nppiSet_32f_C4R . . . . .	174
7.6.2.158 nppiSet_32fc_AC4R . . . . .	174
7.6.2.159 nppiSet_32fc_C1R . . . . .	175
7.6.2.160 nppiSet_32fc_C2R . . . . .	175
7.6.2.161 nppiSet_32fc_C3R . . . . .	175
7.6.2.162 nppiSet_32fc_C4R . . . . .	176
7.6.2.163 nppiSet_32s_AC4MR . . . . .	176
7.6.2.164 nppiSet_32s_AC4R . . . . .	176
7.6.2.165 nppiSet_32s_C1MR . . . . .	177
7.6.2.166 nppiSet_32s_C1R . . . . .	177
7.6.2.167 nppiSet_32s_C4CR . . . . .	178
7.6.2.168 nppiSet_32s_C4MR . . . . .	178

7.6.2.169	<code>nppiSet_32s_C4R</code>	178
7.6.2.170	<code>nppiSet_32sc_AC4R</code>	179
7.6.2.171	<code>nppiSet_32sc_C1R</code>	179
7.6.2.172	<code>nppiSet_32sc_C2R</code>	179
7.6.2.173	<code>nppiSet_32sc_C3R</code>	180
7.6.2.174	<code>nppiSet_32sc_C4R</code>	180
7.6.2.175	<code>nppiSet_8s_AC4R</code>	180
7.6.2.176	<code>nppiSet_8s_C1R</code>	181
7.6.2.177	<code>nppiSet_8s_C2R</code>	181
7.6.2.178	<code>nppiSet_8s_C3R</code>	181
7.6.2.179	<code>nppiSet_8s_C4R</code>	182
7.6.2.180	<code>nppiSet_8u_AC4MR</code>	182
7.6.2.181	<code>nppiSet_8u_AC4R</code>	182
7.6.2.182	<code>nppiSet_8u_C1MR</code>	183
7.6.2.183	<code>nppiSet_8u_C1R</code>	183
7.6.2.184	<code>nppiSet_8u_C4CR</code>	183
7.6.2.185	<code>nppiSet_8u_C4MR</code>	184
7.6.2.186	<code>nppiSet_8u_C4R</code>	184
7.6.2.187	<code>nppiSwapChannels_8u_C4IR</code>	184
7.6.2.188	<code>nppiTranspose_8u_C1R</code>	185
7.7	Arithmetic and Logical Operations	186
7.7.1	Function Documentation	274
7.7.1.1	<code>nppiAbs_16s_AC4IR</code>	274
7.7.1.2	<code>nppiAbs_16s_AC4R</code>	274
7.7.1.3	<code>nppiAbs_16s_C1IR</code>	275
7.7.1.4	<code>nppiAbs_16s_C1R</code>	275
7.7.1.5	<code>nppiAbs_16s_C3IR</code>	275
7.7.1.6	<code>nppiAbs_16s_C3R</code>	276
7.7.1.7	<code>nppiAbs_16s_C4IR</code>	276
7.7.1.8	<code>nppiAbs_16s_C4R</code>	276
7.7.1.9	<code>nppiAbs_32f_AC4IR</code>	277
7.7.1.10	<code>nppiAbs_32f_AC4R</code>	277
7.7.1.11	<code>nppiAbs_32f_C1IR</code>	277
7.7.1.12	<code>nppiAbs_32f_C1R</code>	278
7.7.1.13	<code>nppiAbs_32f_C3IR</code>	278
7.7.1.14	<code>nppiAbs_32f_C3R</code>	278

7.7.1.15	<a href="#">nppiAbs_32f_C4IR</a>	279
7.7.1.16	<a href="#">nppiAbs_32f_C4R</a>	279
7.7.1.17	<a href="#">nppiAbsDiff_16u_C1R</a>	279
7.7.1.18	<a href="#">nppiAbsDiff_32f_C1R</a>	280
7.7.1.19	<a href="#">nppiAbsDiff_8u_C1R</a>	280
7.7.1.20	<a href="#">nppiAbsDiffC_16u_C1R</a>	280
7.7.1.21	<a href="#">nppiAbsDiffC_32f_C1R</a>	281
7.7.1.22	<a href="#">nppiAbsDiffC_8u_C1R</a>	281
7.7.1.23	<a href="#">nppiAdd_16s_AC4IRSfs</a>	282
7.7.1.24	<a href="#">nppiAdd_16s_AC4RSfs</a>	282
7.7.1.25	<a href="#">nppiAdd_16s_C1IRSfs</a>	282
7.7.1.26	<a href="#">nppiAdd_16s_C1RSfs</a>	283
7.7.1.27	<a href="#">nppiAdd_16s_C3IRSfs</a>	283
7.7.1.28	<a href="#">nppiAdd_16s_C3RSfs</a>	284
7.7.1.29	<a href="#">nppiAdd_16s_C4IRSfs</a>	284
7.7.1.30	<a href="#">nppiAdd_16s_C4RSfs</a>	284
7.7.1.31	<a href="#">nppiAdd_16sc_AC4IRSfs</a>	285
7.7.1.32	<a href="#">nppiAdd_16sc_AC4RSfs</a>	285
7.7.1.33	<a href="#">nppiAdd_16sc_C1IRSfs</a>	286
7.7.1.34	<a href="#">nppiAdd_16sc_C1RSfs</a>	286
7.7.1.35	<a href="#">nppiAdd_16sc_C3IRSfs</a>	286
7.7.1.36	<a href="#">nppiAdd_16sc_C3RSfs</a>	287
7.7.1.37	<a href="#">nppiAdd_16sc_C4IRSfs</a>	287
7.7.1.38	<a href="#">nppiAdd_16sc_C4RSfs</a>	288
7.7.1.39	<a href="#">nppiAdd_16u_AC4IRSfs</a>	288
7.7.1.40	<a href="#">nppiAdd_16u_AC4RSfs</a>	288
7.7.1.41	<a href="#">nppiAdd_16u_C1IRSfs</a>	289
7.7.1.42	<a href="#">nppiAdd_16u_C1RSfs</a>	289
7.7.1.43	<a href="#">nppiAdd_16u_C3IRSfs</a>	290
7.7.1.44	<a href="#">nppiAdd_16u_C3RSfs</a>	290
7.7.1.45	<a href="#">nppiAdd_16u_C4IRSfs</a>	291
7.7.1.46	<a href="#">nppiAdd_16u_C4RSfs</a>	291
7.7.1.47	<a href="#">nppiAdd_32f_AC4IR</a>	291
7.7.1.48	<a href="#">nppiAdd_32f_AC4R</a>	292
7.7.1.49	<a href="#">nppiAdd_32f_C1IR</a>	292
7.7.1.50	<a href="#">nppiAdd_32f_C1R</a>	293

7.7.1.51	<a href="#">nppiAdd_32f_C3IR</a>	293
7.7.1.52	<a href="#">nppiAdd_32f_C3R</a>	293
7.7.1.53	<a href="#">nppiAdd_32f_C4IR</a>	294
7.7.1.54	<a href="#">nppiAdd_32f_C4R</a>	294
7.7.1.55	<a href="#">nppiAdd_32fc_AC4IR</a>	295
7.7.1.56	<a href="#">nppiAdd_32fc_AC4R</a>	295
7.7.1.57	<a href="#">nppiAdd_32fc_C1IR</a>	295
7.7.1.58	<a href="#">nppiAdd_32fc_C1R</a>	296
7.7.1.59	<a href="#">nppiAdd_32fc_C3IR</a>	296
7.7.1.60	<a href="#">nppiAdd_32fc_C3R</a>	296
7.7.1.61	<a href="#">nppiAdd_32fc_C4IR</a>	297
7.7.1.62	<a href="#">nppiAdd_32fc_C4R</a>	297
7.7.1.63	<a href="#">nppiAdd_32s_AC4IRSfs</a>	298
7.7.1.64	<a href="#">nppiAdd_32s_AC4RSfs</a>	298
7.7.1.65	<a href="#">nppiAdd_32s_C1IRSfs</a>	298
7.7.1.66	<a href="#">nppiAdd_32s_C1R</a>	299
7.7.1.67	<a href="#">nppiAdd_32s_C1RSfs</a>	299
7.7.1.68	<a href="#">nppiAdd_32s_C3IRSfs</a>	300
7.7.1.69	<a href="#">nppiAdd_32s_C3RSfs</a>	300
7.7.1.70	<a href="#">nppiAdd_32s_C4IRSfs</a>	300
7.7.1.71	<a href="#">nppiAdd_32s_C4RSfs</a>	301
7.7.1.72	<a href="#">nppiAdd_32sc_AC4IRSfs</a>	301
7.7.1.73	<a href="#">nppiAdd_32sc_AC4RSfs</a>	302
7.7.1.74	<a href="#">nppiAdd_32sc_C1IRSfs</a>	302
7.7.1.75	<a href="#">nppiAdd_32sc_C1RSfs</a>	302
7.7.1.76	<a href="#">nppiAdd_32sc_C3IRSfs</a>	303
7.7.1.77	<a href="#">nppiAdd_32sc_C3RSfs</a>	303
7.7.1.78	<a href="#">nppiAdd_32sc_C4IRSfs</a>	304
7.7.1.79	<a href="#">nppiAdd_32sc_C4RSfs</a>	304
7.7.1.80	<a href="#">nppiAdd_8u_AC4IRSfs</a>	305
7.7.1.81	<a href="#">nppiAdd_8u_AC4RSfs</a>	305
7.7.1.82	<a href="#">nppiAdd_8u_C1IRSfs</a>	305
7.7.1.83	<a href="#">nppiAdd_8u_C1RSfs</a>	306
7.7.1.84	<a href="#">nppiAdd_8u_C3IRSfs</a>	306
7.7.1.85	<a href="#">nppiAdd_8u_C3RSfs</a>	307
7.7.1.86	<a href="#">nppiAdd_8u_C4IRSfs</a>	307

7.7.1.87	<a href="#">nppiAdd_8u_C4RSfs</a>	307
7.7.1.88	<a href="#">nppiAddC_16s_AC4IRSfs</a>	308
7.7.1.89	<a href="#">nppiAddC_16s_AC4RSfs</a>	308
7.7.1.90	<a href="#">nppiAddC_16s_C1IRSfs</a>	309
7.7.1.91	<a href="#">nppiAddC_16s_C1RSfs</a>	309
7.7.1.92	<a href="#">nppiAddC_16s_C3IRSfs</a>	309
7.7.1.93	<a href="#">nppiAddC_16s_C3RSfs</a>	310
7.7.1.94	<a href="#">nppiAddC_16s_C4IRSfs</a>	310
7.7.1.95	<a href="#">nppiAddC_16s_C4RSfs</a>	310
7.7.1.96	<a href="#">nppiAddC_16sc_AC4IRSfs</a>	311
7.7.1.97	<a href="#">nppiAddC_16sc_AC4RSfs</a>	311
7.7.1.98	<a href="#">nppiAddC_16sc_C1IRSfs</a>	312
7.7.1.99	<a href="#">nppiAddC_16sc_C1RSfs</a>	312
7.7.1.100	<a href="#">nppiAddC_16sc_C3IRSfs</a>	312
7.7.1.101	<a href="#">nppiAddC_16sc_C3RSfs</a>	313
7.7.1.102	<a href="#">nppiAddC_16sc_C4IRSfs</a>	313
7.7.1.103	<a href="#">nppiAddC_16sc_C4RSfs</a>	314
7.7.1.104	<a href="#">nppiAddC_16u_AC4IRSfs</a>	314
7.7.1.105	<a href="#">nppiAddC_16u_AC4RSfs</a>	314
7.7.1.106	<a href="#">nppiAddC_16u_C1IRSfs</a>	315
7.7.1.107	<a href="#">nppiAddC_16u_C1RSfs</a>	315
7.7.1.108	<a href="#">nppiAddC_16u_C3IRSfs</a>	316
7.7.1.109	<a href="#">nppiAddC_16u_C3RSfs</a>	316
7.7.1.110	<a href="#">nppiAddC_16u_C4IRSfs</a>	316
7.7.1.111	<a href="#">nppiAddC_16u_C4RSfs</a>	317
7.7.1.112	<a href="#">nppiAddC_32f_AC4IR</a>	317
7.7.1.113	<a href="#">nppiAddC_32f_AC4R</a>	317
7.7.1.114	<a href="#">nppiAddC_32f_C1IR</a>	318
7.7.1.115	<a href="#">nppiAddC_32f_C1R</a>	318
7.7.1.116	<a href="#">nppiAddC_32f_C3IR</a>	318
7.7.1.117	<a href="#">nppiAddC_32f_C3R</a>	319
7.7.1.118	<a href="#">nppiAddC_32f_C4IR</a>	319
7.7.1.119	<a href="#">nppiAddC_32f_C4R</a>	319
7.7.1.120	<a href="#">nppiAddC_32fc_AC4IR</a>	320
7.7.1.121	<a href="#">nppiAddC_32fc_AC4R</a>	320
7.7.1.122	<a href="#">nppiAddC_32fc_C1IR</a>	320

7.7.1.123 nppiAddC_32fc_C1R . . . . .	321
7.7.1.124 nppiAddC_32fc_C3IR . . . . .	321
7.7.1.125 nppiAddC_32fc_C3R . . . . .	321
7.7.1.126 nppiAddC_32fc_C4IR . . . . .	322
7.7.1.127 nppiAddC_32fc_C4R . . . . .	322
7.7.1.128 nppiAddC_32s_AC4IRSfs . . . . .	322
7.7.1.129 nppiAddC_32s_AC4RSfs . . . . .	323
7.7.1.130 nppiAddC_32s_C1IRSfs . . . . .	323
7.7.1.131 nppiAddC_32s_C1RSfs . . . . .	324
7.7.1.132 nppiAddC_32s_C3IRSfs . . . . .	324
7.7.1.133 nppiAddC_32s_C3RSfs . . . . .	324
7.7.1.134 nppiAddC_32s_C4IRSfs . . . . .	325
7.7.1.135 nppiAddC_32s_C4RSfs . . . . .	325
7.7.1.136 nppiAddC_32sc_AC4IRSfs . . . . .	326
7.7.1.137 nppiAddC_32sc_AC4RSfs . . . . .	326
7.7.1.138 nppiAddC_32sc_C1IRSfs . . . . .	326
7.7.1.139 nppiAddC_32sc_C1RSfs . . . . .	327
7.7.1.140 nppiAddC_32sc_C3IRSfs . . . . .	327
7.7.1.141 nppiAddC_32sc_C3RSfs . . . . .	328
7.7.1.142 nppiAddC_32sc_C4IRSfs . . . . .	328
7.7.1.143 nppiAddC_32sc_C4RSfs . . . . .	328
7.7.1.144 nppiAddC_8u_AC4IRSfs . . . . .	329
7.7.1.145 nppiAddC_8u_AC4RSfs . . . . .	329
7.7.1.146 nppiAddC_8u_C1IRSfs . . . . .	330
7.7.1.147 nppiAddC_8u_C1RSfs . . . . .	330
7.7.1.148 nppiAddC_8u_C3IRSfs . . . . .	330
7.7.1.149 nppiAddC_8u_C3RSfs . . . . .	331
7.7.1.150 nppiAddC_8u_C4IRSfs . . . . .	331
7.7.1.151 nppiAddC_8u_C4RSfs . . . . .	331
7.7.1.152 nppiAddProduct_16u32f_C1IMR . . . . .	332
7.7.1.153 nppiAddProduct_16u32f_C1IR . . . . .	332
7.7.1.154 nppiAddProduct_32f_C1IMR . . . . .	333
7.7.1.155 nppiAddProduct_32f_C1IR . . . . .	333
7.7.1.156 nppiAddProduct_8u32f_C1IMR . . . . .	333
7.7.1.157 nppiAddProduct_8u32f_C1IR . . . . .	334
7.7.1.158 nppiAddSquare_16u32f_C1IMR . . . . .	334

7.7.1.159 nppiAddSquare_16u32f_C1IR . . . . .	335
7.7.1.160 nppiAddSquare_32f_C1IMR . . . . .	335
7.7.1.161 nppiAddSquare_32f_C1IR . . . . .	336
7.7.1.162 nppiAddSquare_8u32f_C1IMR . . . . .	336
7.7.1.163 nppiAddSquare_8u32f_C1IR . . . . .	336
7.7.1.164 nppiAddWeighted_16u32f_C1IMR . . . . .	337
7.7.1.165 nppiAddWeighted_16u32f_C1IR . . . . .	337
7.7.1.166 nppiAddWeighted_32f_C1IMR . . . . .	337
7.7.1.167 nppiAddWeighted_32f_C1IR . . . . .	338
7.7.1.168 nppiAddWeighted_8u32f_C1IMR . . . . .	338
7.7.1.169 nppiAddWeighted_8u32f_C1IR . . . . .	339
7.7.1.170 nppiAlphaComp_16s_AC1R . . . . .	339
7.7.1.171 nppiAlphaComp_16u_AC1R . . . . .	339
7.7.1.172 nppiAlphaComp_16u_AC4R . . . . .	340
7.7.1.173 nppiAlphaComp_32f_AC1R . . . . .	340
7.7.1.174 nppiAlphaComp_32f_AC4R . . . . .	341
7.7.1.175 nppiAlphaComp_32s_AC1R . . . . .	341
7.7.1.176 nppiAlphaComp_32s_AC4R . . . . .	342
7.7.1.177 nppiAlphaComp_32u_AC1R . . . . .	342
7.7.1.178 nppiAlphaComp_32u_AC4R . . . . .	343
7.7.1.179 nppiAlphaComp_8s_AC1R . . . . .	343
7.7.1.180 nppiAlphaComp_8u_AC1R . . . . .	343
7.7.1.181 nppiAlphaComp_8u_AC4R . . . . .	344
7.7.1.182 nppiAlphaCompC_16s_C1R . . . . .	344
7.7.1.183 nppiAlphaCompC_16u_AC4R . . . . .	345
7.7.1.184 nppiAlphaCompC_16u_C1R . . . . .	345
7.7.1.185 nppiAlphaCompC_16u_C3R . . . . .	346
7.7.1.186 nppiAlphaCompC_16u_C4R . . . . .	346
7.7.1.187 nppiAlphaCompC_32f_C1R . . . . .	347
7.7.1.188 nppiAlphaCompC_32s_C1R . . . . .	347
7.7.1.189 nppiAlphaCompC_32u_C1R . . . . .	348
7.7.1.190 nppiAlphaCompC_8s_C1R . . . . .	348
7.7.1.191 nppiAlphaCompC_8u_AC4R . . . . .	349
7.7.1.192 nppiAlphaCompC_8u_C1R . . . . .	349
7.7.1.193 nppiAlphaCompC_8u_C3R . . . . .	350
7.7.1.194 nppiAlphaCompC_8u_C4R . . . . .	350



7.7.1.195 nppiAlphaPremul_16u_AC4IR . . . . .	351
7.7.1.196 nppiAlphaPremul_16u_AC4R . . . . .	351
7.7.1.197 nppiAlphaPremul_8u_AC4IR . . . . .	351
7.7.1.198 nppiAlphaPremul_8u_AC4R . . . . .	352
7.7.1.199 nppiAlphaPremulC_16u_AC4IR . . . . .	352
7.7.1.200 nppiAlphaPremulC_16u_AC4R . . . . .	352
7.7.1.201 nppiAlphaPremulC_16u_C1IR . . . . .	353
7.7.1.202 nppiAlphaPremulC_16u_C1R . . . . .	353
7.7.1.203 nppiAlphaPremulC_16u_C3IR . . . . .	353
7.7.1.204 nppiAlphaPremulC_16u_C3R . . . . .	354
7.7.1.205 nppiAlphaPremulC_16u_C4IR . . . . .	354
7.7.1.206 nppiAlphaPremulC_16u_C4R . . . . .	354
7.7.1.207 nppiAlphaPremulC_8u_AC4IR . . . . .	355
7.7.1.208 nppiAlphaPremulC_8u_AC4R . . . . .	355
7.7.1.209 nppiAlphaPremulC_8u_C1IR . . . . .	355
7.7.1.210 nppiAlphaPremulC_8u_C1R . . . . .	356
7.7.1.211 nppiAlphaPremulC_8u_C3IR . . . . .	356
7.7.1.212 nppiAlphaPremulC_8u_C3R . . . . .	356
7.7.1.213 nppiAlphaPremulC_8u_C4IR . . . . .	357
7.7.1.214 nppiAlphaPremulC_8u_C4R . . . . .	357
7.7.1.215 nppiAnd_16u_AC4IR . . . . .	357
7.7.1.216 nppiAnd_16u_AC4R . . . . .	358
7.7.1.217 nppiAnd_16u_C1IR . . . . .	358
7.7.1.218 nppiAnd_16u_C1R . . . . .	358
7.7.1.219 nppiAnd_16u_C3IR . . . . .	359
7.7.1.220 nppiAnd_16u_C3R . . . . .	359
7.7.1.221 nppiAnd_16u_C4IR . . . . .	360
7.7.1.222 nppiAnd_16u_C4R . . . . .	360
7.7.1.223 nppiAnd_32s_AC4IR . . . . .	360
7.7.1.224 nppiAnd_32s_AC4R . . . . .	361
7.7.1.225 nppiAnd_32s_C1IR . . . . .	361
7.7.1.226 nppiAnd_32s_C1R . . . . .	361
7.7.1.227 nppiAnd_32s_C3IR . . . . .	362
7.7.1.228 nppiAnd_32s_C3R . . . . .	362
7.7.1.229 nppiAnd_32s_C4IR . . . . .	363
7.7.1.230 nppiAnd_32s_C4R . . . . .	363

7.7.1.231 nppiAnd_8u_AC4IR . . . . .	363
7.7.1.232 nppiAnd_8u_AC4R . . . . .	364
7.7.1.233 nppiAnd_8u_C1IR . . . . .	364
7.7.1.234 nppiAnd_8u_C1R . . . . .	364
7.7.1.235 nppiAnd_8u_C3IR . . . . .	365
7.7.1.236 nppiAnd_8u_C3R . . . . .	365
7.7.1.237 nppiAnd_8u_C4IR . . . . .	366
7.7.1.238 nppiAnd_8u_C4R . . . . .	366
7.7.1.239 nppiAndC_16u_AC4IR . . . . .	366
7.7.1.240 nppiAndC_16u_AC4R . . . . .	367
7.7.1.241 nppiAndC_16u_C1IR . . . . .	367
7.7.1.242 nppiAndC_16u_C1R . . . . .	367
7.7.1.243 nppiAndC_16u_C3IR . . . . .	368
7.7.1.244 nppiAndC_16u_C3R . . . . .	368
7.7.1.245 nppiAndC_16u_C4IR . . . . .	368
7.7.1.246 nppiAndC_16u_C4R . . . . .	369
7.7.1.247 nppiAndC_32s_AC4IR . . . . .	369
7.7.1.248 nppiAndC_32s_AC4R . . . . .	369
7.7.1.249 nppiAndC_32s_C1IR . . . . .	370
7.7.1.250 nppiAndC_32s_C1R . . . . .	370
7.7.1.251 nppiAndC_32s_C3IR . . . . .	370
7.7.1.252 nppiAndC_32s_C3R . . . . .	371
7.7.1.253 nppiAndC_32s_C4IR . . . . .	371
7.7.1.254 nppiAndC_32s_C4R . . . . .	371
7.7.1.255 nppiAndC_8u_AC4IR . . . . .	372
7.7.1.256 nppiAndC_8u_AC4R . . . . .	372
7.7.1.257 nppiAndC_8u_C1IR . . . . .	372
7.7.1.258 nppiAndC_8u_C1R . . . . .	373
7.7.1.259 nppiAndC_8u_C3IR . . . . .	373
7.7.1.260 nppiAndC_8u_C3R . . . . .	373
7.7.1.261 nppiAndC_8u_C4IR . . . . .	374
7.7.1.262 nppiAndC_8u_C4R . . . . .	374
7.7.1.263 nppiDiv_16s_AC4IRSfs . . . . .	374
7.7.1.264 nppiDiv_16s_AC4RSfs . . . . .	375
7.7.1.265 nppiDiv_16s_C1IRSfs . . . . .	375
7.7.1.266 nppiDiv_16s_C1RSfs . . . . .	375

7.7.1.267 nppiDiv_16s_C3IRSfs . . . . .	376
7.7.1.268 nppiDiv_16s_C3RSfs . . . . .	376
7.7.1.269 nppiDiv_16s_C4IRSfs . . . . .	377
7.7.1.270 nppiDiv_16s_C4RSfs . . . . .	377
7.7.1.271 nppiDiv_16sc_AC4IRSfs . . . . .	377
7.7.1.272 nppiDiv_16sc_AC4RSfs . . . . .	378
7.7.1.273 nppiDiv_16sc_C1IRSfs . . . . .	378
7.7.1.274 nppiDiv_16sc_C1RSfs . . . . .	379
7.7.1.275 nppiDiv_16sc_C3IRSfs . . . . .	379
7.7.1.276 nppiDiv_16sc_C3RSfs . . . . .	379
7.7.1.277 nppiDiv_16sc_C4IRSfs . . . . .	380
7.7.1.278 nppiDiv_16sc_C4RSfs . . . . .	380
7.7.1.279 nppiDiv_16u_AC4IRSfs . . . . .	381
7.7.1.280 nppiDiv_16u_AC4RSfs . . . . .	381
7.7.1.281 nppiDiv_16u_C1IRSfs . . . . .	382
7.7.1.282 nppiDiv_16u_C1RSfs . . . . .	382
7.7.1.283 nppiDiv_16u_C3IRSfs . . . . .	382
7.7.1.284 nppiDiv_16u_C3RSfs . . . . .	383
7.7.1.285 nppiDiv_16u_C4IRSfs . . . . .	383
7.7.1.286 nppiDiv_16u_C4RSfs . . . . .	384
7.7.1.287 nppiDiv_32f_AC4IR . . . . .	384
7.7.1.288 nppiDiv_32f_AC4R . . . . .	384
7.7.1.289 nppiDiv_32f_C1IR . . . . .	385
7.7.1.290 nppiDiv_32f_C1R . . . . .	385
7.7.1.291 nppiDiv_32f_C3IR . . . . .	386
7.7.1.292 nppiDiv_32f_C3R . . . . .	386
7.7.1.293 nppiDiv_32f_C4IR . . . . .	386
7.7.1.294 nppiDiv_32f_C4R . . . . .	387
7.7.1.295 nppiDiv_32fc_AC4IR . . . . .	387
7.7.1.296 nppiDiv_32fc_AC4R . . . . .	387
7.7.1.297 nppiDiv_32fc_C1IR . . . . .	388
7.7.1.298 nppiDiv_32fc_C1R . . . . .	388
7.7.1.299 nppiDiv_32fc_C3IR . . . . .	389
7.7.1.300 nppiDiv_32fc_C3R . . . . .	389
7.7.1.301 nppiDiv_32fc_C4IR . . . . .	389
7.7.1.302 nppiDiv_32fc_C4R . . . . .	390

7.7.1.303 nppiDiv_32s_AC4IRSfs . . . . .	390
7.7.1.304 nppiDiv_32s_AC4RSfs . . . . .	390
7.7.1.305 nppiDiv_32s_C1IRSfs . . . . .	391
7.7.1.306 nppiDiv_32s_C1R . . . . .	391
7.7.1.307 nppiDiv_32s_C1RSfs . . . . .	392
7.7.1.308 nppiDiv_32s_C3IRSfs . . . . .	392
7.7.1.309 nppiDiv_32s_C3RSfs . . . . .	392
7.7.1.310 nppiDiv_32s_C4IRSfs . . . . .	393
7.7.1.311 nppiDiv_32s_C4RSfs . . . . .	393
7.7.1.312 nppiDiv_32sc_AC4IRSfs . . . . .	394
7.7.1.313 nppiDiv_32sc_AC4RSfs . . . . .	394
7.7.1.314 nppiDiv_32sc_C1IRSfs . . . . .	395
7.7.1.315 nppiDiv_32sc_C1RSfs . . . . .	395
7.7.1.316 nppiDiv_32sc_C3IRSfs . . . . .	395
7.7.1.317 nppiDiv_32sc_C3RSfs . . . . .	396
7.7.1.318 nppiDiv_32sc_C4IRSfs . . . . .	396
7.7.1.319 nppiDiv_32sc_C4RSfs . . . . .	397
7.7.1.320 nppiDiv_8u_AC4IRSfs . . . . .	397
7.7.1.321 nppiDiv_8u_AC4RSfs . . . . .	397
7.7.1.322 nppiDiv_8u_C1IRSfs . . . . .	398
7.7.1.323 nppiDiv_8u_C1RSfs . . . . .	398
7.7.1.324 nppiDiv_8u_C3IRSfs . . . . .	399
7.7.1.325 nppiDiv_8u_C3RSfs . . . . .	399
7.7.1.326 nppiDiv_8u_C4IRSfs . . . . .	399
7.7.1.327 nppiDiv_8u_C4RSfs . . . . .	400
7.7.1.328 nppiDiv_Round_16s_AC4IRSfs . . . . .	400
7.7.1.329 nppiDiv_Round_16s_AC4RSfs . . . . .	401
7.7.1.330 nppiDiv_Round_16s_C1IRSfs . . . . .	401
7.7.1.331 nppiDiv_Round_16s_C1RSfs . . . . .	402
7.7.1.332 nppiDiv_Round_16s_C3IRSfs . . . . .	402
7.7.1.333 nppiDiv_Round_16s_C3RSfs . . . . .	403
7.7.1.334 nppiDiv_Round_16s_C4IRSfs . . . . .	403
7.7.1.335 nppiDiv_Round_16s_C4RSfs . . . . .	404
7.7.1.336 nppiDiv_Round_16u_AC4IRSfs . . . . .	404
7.7.1.337 nppiDiv_Round_16u_AC4RSfs . . . . .	405
7.7.1.338 nppiDiv_Round_16u_C1IRSfs . . . . .	405

7.7.1.339 nppiDiv_Round_16u_C1RSfs . . . . .	406
7.7.1.340 nppiDiv_Round_16u_C3RSfs . . . . .	406
7.7.1.341 nppiDiv_Round_16u_C3RSfs . . . . .	407
7.7.1.342 nppiDiv_Round_16u_C4RSfs . . . . .	407
7.7.1.343 nppiDiv_Round_16u_C4RSfs . . . . .	408
7.7.1.344 nppiDiv_Round_8u_AC4RSfs . . . . .	408
7.7.1.345 nppiDiv_Round_8u_AC4RSfs . . . . .	409
7.7.1.346 nppiDiv_Round_8u_C1RSfs . . . . .	409
7.7.1.347 nppiDiv_Round_8u_C1RSfs . . . . .	410
7.7.1.348 nppiDiv_Round_8u_C3RSfs . . . . .	410
7.7.1.349 nppiDiv_Round_8u_C3RSfs . . . . .	411
7.7.1.350 nppiDiv_Round_8u_C4RSfs . . . . .	411
7.7.1.351 nppiDiv_Round_8u_C4RSfs . . . . .	412
7.7.1.352 nppiDivC_16s_AC4RSfs . . . . .	412
7.7.1.353 nppiDivC_16s_AC4RSfs . . . . .	412
7.7.1.354 nppiDivC_16s_C1RSfs . . . . .	413
7.7.1.355 nppiDivC_16s_C1RSfs . . . . .	413
7.7.1.356 nppiDivC_16s_C3RSfs . . . . .	414
7.7.1.357 nppiDivC_16s_C3RSfs . . . . .	414
7.7.1.358 nppiDivC_16s_C4RSfs . . . . .	414
7.7.1.359 nppiDivC_16s_C4RSfs . . . . .	415
7.7.1.360 nppiDivC_16sc_AC4RSfs . . . . .	415
7.7.1.361 nppiDivC_16sc_AC4RSfs . . . . .	415
7.7.1.362 nppiDivC_16sc_C1RSfs . . . . .	416
7.7.1.363 nppiDivC_16sc_C1RSfs . . . . .	416
7.7.1.364 nppiDivC_16sc_C3RSfs . . . . .	417
7.7.1.365 nppiDivC_16sc_C3RSfs . . . . .	417
7.7.1.366 nppiDivC_16sc_C4RSfs . . . . .	417
7.7.1.367 nppiDivC_16sc_C4RSfs . . . . .	418
7.7.1.368 nppiDivC_16u_AC4RSfs . . . . .	418
7.7.1.369 nppiDivC_16u_AC4RSfs . . . . .	419
7.7.1.370 nppiDivC_16u_C1RSfs . . . . .	419
7.7.1.371 nppiDivC_16u_C1RSfs . . . . .	419
7.7.1.372 nppiDivC_16u_C3RSfs . . . . .	420
7.7.1.373 nppiDivC_16u_C3RSfs . . . . .	420
7.7.1.374 nppiDivC_16u_C4RSfs . . . . .	421

7.7.1.375 nppiDivC_16u_C4RSfs . . . . .	421
7.7.1.376 nppiDivC_32f_AC4IR . . . . .	421
7.7.1.377 nppiDivC_32f_AC4R . . . . .	422
7.7.1.378 nppiDivC_32f_C1IR . . . . .	422
7.7.1.379 nppiDivC_32f_C1R . . . . .	422
7.7.1.380 nppiDivC_32f_C3IR . . . . .	423
7.7.1.381 nppiDivC_32f_C3R . . . . .	423
7.7.1.382 nppiDivC_32f_C4IR . . . . .	423
7.7.1.383 nppiDivC_32f_C4R . . . . .	424
7.7.1.384 nppiDivC_32fc_AC4IR . . . . .	424
7.7.1.385 nppiDivC_32fc_AC4R . . . . .	424
7.7.1.386 nppiDivC_32fc_C1IR . . . . .	425
7.7.1.387 nppiDivC_32fc_C1R . . . . .	425
7.7.1.388 nppiDivC_32fc_C3IR . . . . .	425
7.7.1.389 nppiDivC_32fc_C3R . . . . .	426
7.7.1.390 nppiDivC_32fc_C4IR . . . . .	426
7.7.1.391 nppiDivC_32fc_C4R . . . . .	426
7.7.1.392 nppiDivC_32s_AC4IRSfs . . . . .	427
7.7.1.393 nppiDivC_32s_AC4RSfs . . . . .	427
7.7.1.394 nppiDivC_32s_C1IRSfs . . . . .	428
7.7.1.395 nppiDivC_32s_C1RSfs . . . . .	428
7.7.1.396 nppiDivC_32s_C3IRSfs . . . . .	428
7.7.1.397 nppiDivC_32s_C3RSfs . . . . .	429
7.7.1.398 nppiDivC_32s_C4IRSfs . . . . .	429
7.7.1.399 nppiDivC_32s_C4RSfs . . . . .	429
7.7.1.400 nppiDivC_32sc_AC4IRSfs . . . . .	430
7.7.1.401 nppiDivC_32sc_AC4RSfs . . . . .	430
7.7.1.402 nppiDivC_32sc_C1IRSfs . . . . .	431
7.7.1.403 nppiDivC_32sc_C1RSfs . . . . .	431
7.7.1.404 nppiDivC_32sc_C3IRSfs . . . . .	431
7.7.1.405 nppiDivC_32sc_C3RSfs . . . . .	432
7.7.1.406 nppiDivC_32sc_C4IRSfs . . . . .	432
7.7.1.407 nppiDivC_32sc_C4RSfs . . . . .	433
7.7.1.408 nppiDivC_8u_AC4IRSfs . . . . .	433
7.7.1.409 nppiDivC_8u_AC4RSfs . . . . .	433
7.7.1.410 nppiDivC_8u_C1IRSfs . . . . .	434

7.7.1.411 nppiDivC_8u_C1RSfs . . . . .	434
7.7.1.412 nppiDivC_8u_C3RSfs . . . . .	435
7.7.1.413 nppiDivC_8u_C3RSfs . . . . .	435
7.7.1.414 nppiDivC_8u_C4RSfs . . . . .	435
7.7.1.415 nppiDivC_8u_C4RSfs . . . . .	436
7.7.1.416 nppiExp_16s_C1RSfs . . . . .	436
7.7.1.417 nppiExp_16s_C1RSfs . . . . .	436
7.7.1.418 nppiExp_16s_C3RSfs . . . . .	437
7.7.1.419 nppiExp_16s_C3RSfs . . . . .	437
7.7.1.420 nppiExp_16u_C1RSfs . . . . .	437
7.7.1.421 nppiExp_16u_C1RSfs . . . . .	438
7.7.1.422 nppiExp_16u_C3RSfs . . . . .	438
7.7.1.423 nppiExp_16u_C3RSfs . . . . .	439
7.7.1.424 nppiExp_32f_C1IR . . . . .	439
7.7.1.425 nppiExp_32f_C1R . . . . .	439
7.7.1.426 nppiExp_32f_C3IR . . . . .	440
7.7.1.427 nppiExp_32f_C3R . . . . .	440
7.7.1.428 nppiExp_8u_C1RSfs . . . . .	440
7.7.1.429 nppiExp_8u_C1RSfs . . . . .	441
7.7.1.430 nppiExp_8u_C3RSfs . . . . .	441
7.7.1.431 nppiExp_8u_C3RSfs . . . . .	441
7.7.1.432 nppiLn_16s_C1RSfs . . . . .	442
7.7.1.433 nppiLn_16s_C1RSfs . . . . .	442
7.7.1.434 nppiLn_16s_C3RSfs . . . . .	442
7.7.1.435 nppiLn_16s_C3RSfs . . . . .	443
7.7.1.436 nppiLn_16u_C1RSfs . . . . .	443
7.7.1.437 nppiLn_16u_C1RSfs . . . . .	443
7.7.1.438 nppiLn_16u_C3RSfs . . . . .	444
7.7.1.439 nppiLn_16u_C3RSfs . . . . .	444
7.7.1.440 nppiLn_32f_C1IR . . . . .	444
7.7.1.441 nppiLn_32f_C1R . . . . .	445
7.7.1.442 nppiLn_32f_C3IR . . . . .	445
7.7.1.443 nppiLn_32f_C3R . . . . .	445
7.7.1.444 nppiLn_8u_C1RSfs . . . . .	446
7.7.1.445 nppiLn_8u_C1RSfs . . . . .	446
7.7.1.446 nppiLn_8u_C3RSfs . . . . .	446

7.7.1.447 nppiLn_8u_C3RSfs . . . . .	447
7.7.1.448 nppiLShiftC_16u_AC4IR . . . . .	447
7.7.1.449 nppiLShiftC_16u_AC4R . . . . .	447
7.7.1.450 nppiLShiftC_16u_C1IR . . . . .	448
7.7.1.451 nppiLShiftC_16u_C1R . . . . .	448
7.7.1.452 nppiLShiftC_16u_C3IR . . . . .	448
7.7.1.453 nppiLShiftC_16u_C3R . . . . .	449
7.7.1.454 nppiLShiftC_16u_C4IR . . . . .	449
7.7.1.455 nppiLShiftC_16u_C4R . . . . .	449
7.7.1.456 nppiLShiftC_32s_AC4IR . . . . .	450
7.7.1.457 nppiLShiftC_32s_AC4R . . . . .	450
7.7.1.458 nppiLShiftC_32s_C1IR . . . . .	450
7.7.1.459 nppiLShiftC_32s_C1R . . . . .	451
7.7.1.460 nppiLShiftC_32s_C3IR . . . . .	451
7.7.1.461 nppiLShiftC_32s_C3R . . . . .	451
7.7.1.462 nppiLShiftC_32s_C4IR . . . . .	452
7.7.1.463 nppiLShiftC_32s_C4R . . . . .	452
7.7.1.464 nppiLShiftC_8u_AC4IR . . . . .	452
7.7.1.465 nppiLShiftC_8u_AC4R . . . . .	453
7.7.1.466 nppiLShiftC_8u_C1IR . . . . .	453
7.7.1.467 nppiLShiftC_8u_C1R . . . . .	453
7.7.1.468 nppiLShiftC_8u_C3IR . . . . .	454
7.7.1.469 nppiLShiftC_8u_C3R . . . . .	454
7.7.1.470 nppiLShiftC_8u_C4IR . . . . .	454
7.7.1.471 nppiLShiftC_8u_C4R . . . . .	455
7.7.1.472 nppiMul_16s_AC4IRSfs . . . . .	455
7.7.1.473 nppiMul_16s_AC4RSfs . . . . .	455
7.7.1.474 nppiMul_16s_C1IRSfs . . . . .	456
7.7.1.475 nppiMul_16s_C1RSfs . . . . .	456
7.7.1.476 nppiMul_16s_C3IRSfs . . . . .	457
7.7.1.477 nppiMul_16s_C3RSfs . . . . .	457
7.7.1.478 nppiMul_16s_C4IRSfs . . . . .	457
7.7.1.479 nppiMul_16s_C4RSfs . . . . .	458
7.7.1.480 nppiMul_16sc_AC4IRSfs . . . . .	458
7.7.1.481 nppiMul_16sc_AC4RSfs . . . . .	459
7.7.1.482 nppiMul_16sc_C1IRSfs . . . . .	459



7.7.1.483 nppiMul_16sc_C1RSfs . . . . .	459
7.7.1.484 nppiMul_16sc_C3RSfs . . . . .	460
7.7.1.485 nppiMul_16sc_C3RSfs . . . . .	460
7.7.1.486 nppiMul_16sc_C4RSfs . . . . .	461
7.7.1.487 nppiMul_16sc_C4RSfs . . . . .	461
7.7.1.488 nppiMul_16u_AC4RSfs . . . . .	462
7.7.1.489 nppiMul_16u_AC4RSfs . . . . .	462
7.7.1.490 nppiMul_16u_C1RSfs . . . . .	462
7.7.1.491 nppiMul_16u_C1RSfs . . . . .	463
7.7.1.492 nppiMul_16u_C3RSfs . . . . .	463
7.7.1.493 nppiMul_16u_C3RSfs . . . . .	464
7.7.1.494 nppiMul_16u_C4RSfs . . . . .	464
7.7.1.495 nppiMul_16u_C4RSfs . . . . .	464
7.7.1.496 nppiMul_32f_AC4IR . . . . .	465
7.7.1.497 nppiMul_32f_AC4R . . . . .	465
7.7.1.498 nppiMul_32f_C1IR . . . . .	466
7.7.1.499 nppiMul_32f_C1R . . . . .	466
7.7.1.500 nppiMul_32f_C3IR . . . . .	466
7.7.1.501 nppiMul_32f_C3R . . . . .	467
7.7.1.502 nppiMul_32f_C4IR . . . . .	467
7.7.1.503 nppiMul_32f_C4R . . . . .	467
7.7.1.504 nppiMul_32fc_AC4IR . . . . .	468
7.7.1.505 nppiMul_32fc_AC4R . . . . .	468
7.7.1.506 nppiMul_32fc_C1IR . . . . .	469
7.7.1.507 nppiMul_32fc_C1R . . . . .	469
7.7.1.508 nppiMul_32fc_C3IR . . . . .	469
7.7.1.509 nppiMul_32fc_C3R . . . . .	470
7.7.1.510 nppiMul_32fc_C4IR . . . . .	470
7.7.1.511 nppiMul_32fc_C4R . . . . .	470
7.7.1.512 nppiMul_32s_AC4RSfs . . . . .	471
7.7.1.513 nppiMul_32s_AC4RSfs . . . . .	471
7.7.1.514 nppiMul_32s_C1RSfs . . . . .	472
7.7.1.515 nppiMul_32s_C1R . . . . .	472
7.7.1.516 nppiMul_32s_C1RSfs . . . . .	473
7.7.1.517 nppiMul_32s_C3RSfs . . . . .	473
7.7.1.518 nppiMul_32s_C3RSfs . . . . .	473

7.7.1.519 nppiMul_32s_C4IRSfs . . . . .	474
7.7.1.520 nppiMul_32s_C4RSfs . . . . .	474
7.7.1.521 nppiMul_32sc_AC4IRSfs . . . . .	475
7.7.1.522 nppiMul_32sc_AC4RSfs . . . . .	475
7.7.1.523 nppiMul_32sc_C1IRSfs . . . . .	476
7.7.1.524 nppiMul_32sc_C1RSfs . . . . .	476
7.7.1.525 nppiMul_32sc_C3IRSfs . . . . .	476
7.7.1.526 nppiMul_32sc_C3RSfs . . . . .	477
7.7.1.527 nppiMul_32sc_C4IRSfs . . . . .	477
7.7.1.528 nppiMul_32sc_C4RSfs . . . . .	478
7.7.1.529 nppiMul_8u_AC4IRSfs . . . . .	478
7.7.1.530 nppiMul_8u_AC4RSfs . . . . .	478
7.7.1.531 nppiMul_8u_C1IRSfs . . . . .	479
7.7.1.532 nppiMul_8u_C1RSfs . . . . .	479
7.7.1.533 nppiMul_8u_C3IRSfs . . . . .	480
7.7.1.534 nppiMul_8u_C3RSfs . . . . .	480
7.7.1.535 nppiMul_8u_C4IRSfs . . . . .	480
7.7.1.536 nppiMul_8u_C4RSfs . . . . .	481
7.7.1.537 nppiMulC_16s_AC4IRSfs . . . . .	481
7.7.1.538 nppiMulC_16s_AC4RSfs . . . . .	482
7.7.1.539 nppiMulC_16s_C1IRSfs . . . . .	482
7.7.1.540 nppiMulC_16s_C1RSfs . . . . .	482
7.7.1.541 nppiMulC_16s_C3IRSfs . . . . .	483
7.7.1.542 nppiMulC_16s_C3RSfs . . . . .	483
7.7.1.543 nppiMulC_16s_C4IRSfs . . . . .	484
7.7.1.544 nppiMulC_16s_C4RSfs . . . . .	484
7.7.1.545 nppiMulC_16sc_AC4IRSfs . . . . .	484
7.7.1.546 nppiMulC_16sc_AC4RSfs . . . . .	485
7.7.1.547 nppiMulC_16sc_C1IRSfs . . . . .	485
7.7.1.548 nppiMulC_16sc_C1RSfs . . . . .	485
7.7.1.549 nppiMulC_16sc_C3IRSfs . . . . .	486
7.7.1.550 nppiMulC_16sc_C3RSfs . . . . .	486
7.7.1.551 nppiMulC_16sc_C4IRSfs . . . . .	487
7.7.1.552 nppiMulC_16sc_C4RSfs . . . . .	487
7.7.1.553 nppiMulC_16u_AC4IRSfs . . . . .	487
7.7.1.554 nppiMulC_16u_AC4RSfs . . . . .	488

7.7.1.555 nppiMulC_16u_C1IRSfs . . . . .	488
7.7.1.556 nppiMulC_16u_C1RSfs . . . . .	489
7.7.1.557 nppiMulC_16u_C3IRSfs . . . . .	489
7.7.1.558 nppiMulC_16u_C3RSfs . . . . .	489
7.7.1.559 nppiMulC_16u_C4IRSfs . . . . .	490
7.7.1.560 nppiMulC_16u_C4RSfs . . . . .	490
7.7.1.561 nppiMulC_32f_AC4IR . . . . .	491
7.7.1.562 nppiMulC_32f_AC4R . . . . .	491
7.7.1.563 nppiMulC_32f_C1IR . . . . .	491
7.7.1.564 nppiMulC_32f_C1R . . . . .	492
7.7.1.565 nppiMulC_32f_C3IR . . . . .	492
7.7.1.566 nppiMulC_32f_C3R . . . . .	492
7.7.1.567 nppiMulC_32f_C4IR . . . . .	493
7.7.1.568 nppiMulC_32f_C4R . . . . .	493
7.7.1.569 nppiMulC_32fc_AC4IR . . . . .	493
7.7.1.570 nppiMulC_32fc_AC4R . . . . .	494
7.7.1.571 nppiMulC_32fc_C1IR . . . . .	494
7.7.1.572 nppiMulC_32fc_C1R . . . . .	494
7.7.1.573 nppiMulC_32fc_C3IR . . . . .	495
7.7.1.574 nppiMulC_32fc_C3R . . . . .	495
7.7.1.575 nppiMulC_32fc_C4IR . . . . .	495
7.7.1.576 nppiMulC_32fc_C4R . . . . .	496
7.7.1.577 nppiMulC_32s_AC4IRSfs . . . . .	496
7.7.1.578 nppiMulC_32s_AC4RSfs . . . . .	497
7.7.1.579 nppiMulC_32s_C1IRSfs . . . . .	497
7.7.1.580 nppiMulC_32s_C1RSfs . . . . .	497
7.7.1.581 nppiMulC_32s_C3IRSfs . . . . .	498
7.7.1.582 nppiMulC_32s_C3RSfs . . . . .	498
7.7.1.583 nppiMulC_32s_C4IRSfs . . . . .	499
7.7.1.584 nppiMulC_32s_C4RSfs . . . . .	499
7.7.1.585 nppiMulC_32sc_AC4IRSfs . . . . .	499
7.7.1.586 nppiMulC_32sc_AC4RSfs . . . . .	500
7.7.1.587 nppiMulC_32sc_C1IRSfs . . . . .	500
7.7.1.588 nppiMulC_32sc_C1RSfs . . . . .	501
7.7.1.589 nppiMulC_32sc_C3IRSfs . . . . .	501
7.7.1.590 nppiMulC_32sc_C3RSfs . . . . .	501

7.7.1.591 nppiMulC_32sc_C4IRSfs . . . . .	502
7.7.1.592 nppiMulC_32sc_C4RSfs . . . . .	502
7.7.1.593 nppiMulC_8u_AC4IRSfs . . . . .	503
7.7.1.594 nppiMulC_8u_AC4RSfs . . . . .	503
7.7.1.595 nppiMulC_8u_C1IRSfs . . . . .	503
7.7.1.596 nppiMulC_8u_C1RSfs . . . . .	504
7.7.1.597 nppiMulC_8u_C3IRSfs . . . . .	504
7.7.1.598 nppiMulC_8u_C3RSfs . . . . .	504
7.7.1.599 nppiMulC_8u_C4IRSfs . . . . .	505
7.7.1.600 nppiMulC_8u_C4RSfs . . . . .	505
7.7.1.601 nppiMulCScale_16u_AC4IR . . . . .	506
7.7.1.602 nppiMulCScale_16u_AC4R . . . . .	506
7.7.1.603 nppiMulCScale_16u_C1IR . . . . .	506
7.7.1.604 nppiMulCScale_16u_C1R . . . . .	507
7.7.1.605 nppiMulCScale_16u_C3IR . . . . .	507
7.7.1.606 nppiMulCScale_16u_C3R . . . . .	507
7.7.1.607 nppiMulCScale_16u_C4IR . . . . .	508
7.7.1.608 nppiMulCScale_16u_C4R . . . . .	508
7.7.1.609 nppiMulCScale_8u_AC4IR . . . . .	508
7.7.1.610 nppiMulCScale_8u_AC4R . . . . .	509
7.7.1.611 nppiMulCScale_8u_C1IR . . . . .	509
7.7.1.612 nppiMulCScale_8u_C1R . . . . .	509
7.7.1.613 nppiMulCScale_8u_C3IR . . . . .	510
7.7.1.614 nppiMulCScale_8u_C3R . . . . .	510
7.7.1.615 nppiMulCScale_8u_C4IR . . . . .	510
7.7.1.616 nppiMulCScale_8u_C4R . . . . .	511
7.7.1.617 nppiMulScale_16u_AC4IR . . . . .	511
7.7.1.618 nppiMulScale_16u_AC4R . . . . .	511
7.7.1.619 nppiMulScale_16u_C1IR . . . . .	512
7.7.1.620 nppiMulScale_16u_C1R . . . . .	512
7.7.1.621 nppiMulScale_16u_C3IR . . . . .	513
7.7.1.622 nppiMulScale_16u_C3R . . . . .	513
7.7.1.623 nppiMulScale_16u_C4IR . . . . .	513
7.7.1.624 nppiMulScale_16u_C4R . . . . .	514
7.7.1.625 nppiMulScale_8u_AC4IR . . . . .	514
7.7.1.626 nppiMulScale_8u_AC4R . . . . .	514

7.7.1.627 nppiMulScale_8u_C1IR . . . . .	515
7.7.1.628 nppiMulScale_8u_C1R . . . . .	515
7.7.1.629 nppiMulScale_8u_C3IR . . . . .	516
7.7.1.630 nppiMulScale_8u_C3R . . . . .	516
7.7.1.631 nppiMulScale_8u_C4IR . . . . .	516
7.7.1.632 nppiMulScale_8u_C4R . . . . .	517
7.7.1.633 nppiNot_8u_AC4IR . . . . .	517
7.7.1.634 nppiNot_8u_AC4R . . . . .	517
7.7.1.635 nppiNot_8u_C1IR . . . . .	518
7.7.1.636 nppiNot_8u_C1R . . . . .	518
7.7.1.637 nppiNot_8u_C3IR . . . . .	518
7.7.1.638 nppiNot_8u_C3R . . . . .	519
7.7.1.639 nppiNot_8u_C4IR . . . . .	519
7.7.1.640 nppiNot_8u_C4R . . . . .	519
7.7.1.641 nppiOr_16u_AC4IR . . . . .	520
7.7.1.642 nppiOr_16u_AC4R . . . . .	520
7.7.1.643 nppiOr_16u_C1IR . . . . .	520
7.7.1.644 nppiOr_16u_C1R . . . . .	521
7.7.1.645 nppiOr_16u_C3IR . . . . .	521
7.7.1.646 nppiOr_16u_C3R . . . . .	521
7.7.1.647 nppiOr_16u_C4IR . . . . .	522
7.7.1.648 nppiOr_16u_C4R . . . . .	522
7.7.1.649 nppiOr_32s_AC4IR . . . . .	523
7.7.1.650 nppiOr_32s_AC4R . . . . .	523
7.7.1.651 nppiOr_32s_C1IR . . . . .	523
7.7.1.652 nppiOr_32s_C1R . . . . .	524
7.7.1.653 nppiOr_32s_C3IR . . . . .	524
7.7.1.654 nppiOr_32s_C3R . . . . .	524
7.7.1.655 nppiOr_32s_C4IR . . . . .	525
7.7.1.656 nppiOr_32s_C4R . . . . .	525
7.7.1.657 nppiOr_8u_AC4IR . . . . .	526
7.7.1.658 nppiOr_8u_AC4R . . . . .	526
7.7.1.659 nppiOr_8u_C1IR . . . . .	526
7.7.1.660 nppiOr_8u_C1R . . . . .	527
7.7.1.661 nppiOr_8u_C3IR . . . . .	527
7.7.1.662 nppiOr_8u_C3R . . . . .	527

7.7.1.663 nppiOr_8u_C4IR . . . . .	528
7.7.1.664 nppiOr_8u_C4R . . . . .	528
7.7.1.665 nppiOrC_16u_AC4IR . . . . .	529
7.7.1.666 nppiOrC_16u_AC4R . . . . .	529
7.7.1.667 nppiOrC_16u_C1IR . . . . .	529
7.7.1.668 nppiOrC_16u_C1R . . . . .	530
7.7.1.669 nppiOrC_16u_C3IR . . . . .	530
7.7.1.670 nppiOrC_16u_C3R . . . . .	530
7.7.1.671 nppiOrC_16u_C4IR . . . . .	531
7.7.1.672 nppiOrC_16u_C4R . . . . .	531
7.7.1.673 nppiOrC_32s_AC4IR . . . . .	531
7.7.1.674 nppiOrC_32s_AC4R . . . . .	532
7.7.1.675 nppiOrC_32s_C1IR . . . . .	532
7.7.1.676 nppiOrC_32s_C1R . . . . .	532
7.7.1.677 nppiOrC_32s_C3IR . . . . .	533
7.7.1.678 nppiOrC_32s_C3R . . . . .	533
7.7.1.679 nppiOrC_32s_C4IR . . . . .	533
7.7.1.680 nppiOrC_32s_C4R . . . . .	534
7.7.1.681 nppiOrC_8u_AC4IR . . . . .	534
7.7.1.682 nppiOrC_8u_AC4R . . . . .	534
7.7.1.683 nppiOrC_8u_C1IR . . . . .	535
7.7.1.684 nppiOrC_8u_C1R . . . . .	535
7.7.1.685 nppiOrC_8u_C3IR . . . . .	535
7.7.1.686 nppiOrC_8u_C3R . . . . .	536
7.7.1.687 nppiOrC_8u_C4IR . . . . .	536
7.7.1.688 nppiOrC_8u_C4R . . . . .	536
7.7.1.689 nppiRShiftC_16s_AC4IR . . . . .	537
7.7.1.690 nppiRShiftC_16s_AC4R . . . . .	537
7.7.1.691 nppiRShiftC_16s_C1IR . . . . .	537
7.7.1.692 nppiRShiftC_16s_C1R . . . . .	538
7.7.1.693 nppiRShiftC_16s_C3IR . . . . .	538
7.7.1.694 nppiRShiftC_16s_C3R . . . . .	538
7.7.1.695 nppiRShiftC_16s_C4IR . . . . .	539
7.7.1.696 nppiRShiftC_16s_C4R . . . . .	539
7.7.1.697 nppiRShiftC_16u_AC4IR . . . . .	539
7.7.1.698 nppiRShiftC_16u_AC4R . . . . .	540

7.7.1.699 nppiRShiftC_16u_C1IR . . . . .	540
7.7.1.700 nppiRShiftC_16u_C1R . . . . .	540
7.7.1.701 nppiRShiftC_16u_C3IR . . . . .	541
7.7.1.702 nppiRShiftC_16u_C3R . . . . .	541
7.7.1.703 nppiRShiftC_16u_C4IR . . . . .	541
7.7.1.704 nppiRShiftC_16u_C4R . . . . .	542
7.7.1.705 nppiRShiftC_32s_AC4IR . . . . .	542
7.7.1.706 nppiRShiftC_32s_AC4R . . . . .	542
7.7.1.707 nppiRShiftC_32s_C1IR . . . . .	543
7.7.1.708 nppiRShiftC_32s_C1R . . . . .	543
7.7.1.709 nppiRShiftC_32s_C3IR . . . . .	543
7.7.1.710 nppiRShiftC_32s_C3R . . . . .	544
7.7.1.711 nppiRShiftC_32s_C4IR . . . . .	544
7.7.1.712 nppiRShiftC_32s_C4R . . . . .	544
7.7.1.713 nppiRShiftC_8s_AC4IR . . . . .	545
7.7.1.714 nppiRShiftC_8s_AC4R . . . . .	545
7.7.1.715 nppiRShiftC_8s_C1IR . . . . .	545
7.7.1.716 nppiRShiftC_8s_C1R . . . . .	546
7.7.1.717 nppiRShiftC_8s_C3IR . . . . .	546
7.7.1.718 nppiRShiftC_8s_C3R . . . . .	546
7.7.1.719 nppiRShiftC_8s_C4IR . . . . .	547
7.7.1.720 nppiRShiftC_8s_C4R . . . . .	547
7.7.1.721 nppiRShiftC_8u_AC4IR . . . . .	547
7.7.1.722 nppiRShiftC_8u_AC4R . . . . .	548
7.7.1.723 nppiRShiftC_8u_C1IR . . . . .	548
7.7.1.724 nppiRShiftC_8u_C1R . . . . .	548
7.7.1.725 nppiRShiftC_8u_C3IR . . . . .	549
7.7.1.726 nppiRShiftC_8u_C3R . . . . .	549
7.7.1.727 nppiRShiftC_8u_C4IR . . . . .	549
7.7.1.728 nppiRShiftC_8u_C4R . . . . .	550
7.7.1.729 nppiSqr_16s_AC4IRSfs . . . . .	550
7.7.1.730 nppiSqr_16s_AC4RSfs . . . . .	550
7.7.1.731 nppiSqr_16s_C1IRSfs . . . . .	551
7.7.1.732 nppiSqr_16s_C1RSfs . . . . .	551
7.7.1.733 nppiSqr_16s_C3IRSfs . . . . .	551
7.7.1.734 nppiSqr_16s_C3RSfs . . . . .	552

7.7.1.735 nppiSqr_16s_C4IRSfs . . . . .	552
7.7.1.736 nppiSqr_16s_C4RSfs . . . . .	552
7.7.1.737 nppiSqr_16u_AC4IRSfs . . . . .	553
7.7.1.738 nppiSqr_16u_AC4RSfs . . . . .	553
7.7.1.739 nppiSqr_16u_C1IRSfs . . . . .	553
7.7.1.740 nppiSqr_16u_C1RSfs . . . . .	554
7.7.1.741 nppiSqr_16u_C3IRSfs . . . . .	554
7.7.1.742 nppiSqr_16u_C3RSfs . . . . .	554
7.7.1.743 nppiSqr_16u_C4IRSfs . . . . .	555
7.7.1.744 nppiSqr_16u_C4RSfs . . . . .	555
7.7.1.745 nppiSqr_32f_AC4IR . . . . .	555
7.7.1.746 nppiSqr_32f_AC4R . . . . .	556
7.7.1.747 nppiSqr_32f_C1IR . . . . .	556
7.7.1.748 nppiSqr_32f_C1R . . . . .	556
7.7.1.749 nppiSqr_32f_C3IR . . . . .	557
7.7.1.750 nppiSqr_32f_C3R . . . . .	557
7.7.1.751 nppiSqr_32f_C4IR . . . . .	557
7.7.1.752 nppiSqr_32f_C4R . . . . .	558
7.7.1.753 nppiSqr_8u_AC4IRSfs . . . . .	558
7.7.1.754 nppiSqr_8u_AC4RSfs . . . . .	558
7.7.1.755 nppiSqr_8u_C1IRSfs . . . . .	559
7.7.1.756 nppiSqr_8u_C1RSfs . . . . .	559
7.7.1.757 nppiSqr_8u_C3IRSfs . . . . .	559
7.7.1.758 nppiSqr_8u_C3RSfs . . . . .	560
7.7.1.759 nppiSqr_8u_C4IRSfs . . . . .	560
7.7.1.760 nppiSqr_8u_C4RSfs . . . . .	560
7.7.1.761 nppiSqrt_16s_AC4IRSfs . . . . .	561
7.7.1.762 nppiSqrt_16s_AC4RSfs . . . . .	561
7.7.1.763 nppiSqrt_16s_C1IRSfs . . . . .	561
7.7.1.764 nppiSqrt_16s_C1RSfs . . . . .	562
7.7.1.765 nppiSqrt_16s_C3IRSfs . . . . .	562
7.7.1.766 nppiSqrt_16s_C3RSfs . . . . .	562
7.7.1.767 nppiSqrt_16s_C4IRSfs . . . . .	563
7.7.1.768 nppiSqrt_16s_C4RSfs . . . . .	563
7.7.1.769 nppiSqrt_16u_AC4IRSfs . . . . .	563
7.7.1.770 nppiSqrt_16u_AC4RSfs . . . . .	564



7.7.1.771 nppiSqrt_16u_C1IRSfs . . . . .	564
7.7.1.772 nppiSqrt_16u_C1RSfs . . . . .	564
7.7.1.773 nppiSqrt_16u_C3IRSfs . . . . .	565
7.7.1.774 nppiSqrt_16u_C3RSfs . . . . .	565
7.7.1.775 nppiSqrt_16u_C4IRSfs . . . . .	565
7.7.1.776 nppiSqrt_16u_C4RSfs . . . . .	566
7.7.1.777 nppiSqrt_32f_AC4IR . . . . .	566
7.7.1.778 nppiSqrt_32f_AC4R . . . . .	566
7.7.1.779 nppiSqrt_32f_C1IR . . . . .	567
7.7.1.780 nppiSqrt_32f_C1R . . . . .	567
7.7.1.781 nppiSqrt_32f_C3IR . . . . .	567
7.7.1.782 nppiSqrt_32f_C3R . . . . .	568
7.7.1.783 nppiSqrt_32f_C4IR . . . . .	568
7.7.1.784 nppiSqrt_32f_C4R . . . . .	568
7.7.1.785 nppiSqrt_8u_AC4IRSfs . . . . .	569
7.7.1.786 nppiSqrt_8u_AC4RSfs . . . . .	569
7.7.1.787 nppiSqrt_8u_C1IRSfs . . . . .	569
7.7.1.788 nppiSqrt_8u_C1RSfs . . . . .	570
7.7.1.789 nppiSqrt_8u_C3IRSfs . . . . .	570
7.7.1.790 nppiSqrt_8u_C3RSfs . . . . .	570
7.7.1.791 nppiSqrt_8u_C4IRSfs . . . . .	571
7.7.1.792 nppiSqrt_8u_C4RSfs . . . . .	571
7.7.1.793 nppiSub_16s_AC4IRSfs . . . . .	571
7.7.1.794 nppiSub_16s_AC4RSfs . . . . .	572
7.7.1.795 nppiSub_16s_C1IRSfs . . . . .	572
7.7.1.796 nppiSub_16s_C1RSfs . . . . .	573
7.7.1.797 nppiSub_16s_C3IRSfs . . . . .	573
7.7.1.798 nppiSub_16s_C3RSfs . . . . .	573
7.7.1.799 nppiSub_16s_C4IRSfs . . . . .	574
7.7.1.800 nppiSub_16s_C4RSfs . . . . .	574
7.7.1.801 nppiSub_16sc_AC4IRSfs . . . . .	575
7.7.1.802 nppiSub_16sc_AC4RSfs . . . . .	575
7.7.1.803 nppiSub_16sc_C1IRSfs . . . . .	576
7.7.1.804 nppiSub_16sc_C1RSfs . . . . .	576
7.7.1.805 nppiSub_16sc_C3IRSfs . . . . .	576
7.7.1.806 nppiSub_16sc_C3RSfs . . . . .	577

7.7.1.807 nppiSub_16sc_C4IRSfs . . . . .	577
7.7.1.808 nppiSub_16sc_C4RSfs . . . . .	578
7.7.1.809 nppiSub_16u_AC4IRSfs . . . . .	578
7.7.1.810 nppiSub_16u_AC4RSfs . . . . .	578
7.7.1.811 nppiSub_16u_C1IRSfs . . . . .	579
7.7.1.812 nppiSub_16u_C1RSfs . . . . .	579
7.7.1.813 nppiSub_16u_C3IRSfs . . . . .	580
7.7.1.814 nppiSub_16u_C3RSfs . . . . .	580
7.7.1.815 nppiSub_16u_C4IRSfs . . . . .	581
7.7.1.816 nppiSub_16u_C4RSfs . . . . .	581
7.7.1.817 nppiSub_32f_AC4IR . . . . .	581
7.7.1.818 nppiSub_32f_AC4R . . . . .	582
7.7.1.819 nppiSub_32f_C1IR . . . . .	582
7.7.1.820 nppiSub_32f_C1R . . . . .	583
7.7.1.821 nppiSub_32f_C3IR . . . . .	583
7.7.1.822 nppiSub_32f_C3R . . . . .	583
7.7.1.823 nppiSub_32f_C4IR . . . . .	584
7.7.1.824 nppiSub_32f_C4R . . . . .	584
7.7.1.825 nppiSub_32fc_AC4IR . . . . .	585
7.7.1.826 nppiSub_32fc_AC4R . . . . .	585
7.7.1.827 nppiSub_32fc_C1IR . . . . .	585
7.7.1.828 nppiSub_32fc_C1R . . . . .	586
7.7.1.829 nppiSub_32fc_C3IR . . . . .	586
7.7.1.830 nppiSub_32fc_C3R . . . . .	587
7.7.1.831 nppiSub_32fc_C4IR . . . . .	587
7.7.1.832 nppiSub_32fc_C4R . . . . .	587
7.7.1.833 nppiSub_32s_AC4IRSfs . . . . .	588
7.7.1.834 nppiSub_32s_AC4RSfs . . . . .	588
7.7.1.835 nppiSub_32s_C1IRSfs . . . . .	589
7.7.1.836 nppiSub_32s_C1R . . . . .	589
7.7.1.837 nppiSub_32s_C1RSfs . . . . .	589
7.7.1.838 nppiSub_32s_C3IRSfs . . . . .	590
7.7.1.839 nppiSub_32s_C3RSfs . . . . .	590
7.7.1.840 nppiSub_32s_C4IRSfs . . . . .	591
7.7.1.841 nppiSub_32s_C4RSfs . . . . .	591
7.7.1.842 nppiSub_32sc_AC4IRSfs . . . . .	591

7.7.1.843 nppiSub_32sc_AC4RSfs . . . . .	592
7.7.1.844 nppiSub_32sc_C1IRSfs . . . . .	592
7.7.1.845 nppiSub_32sc_C1RSfs . . . . .	593
7.7.1.846 nppiSub_32sc_C3IRSfs . . . . .	593
7.7.1.847 nppiSub_32sc_C3RSfs . . . . .	593
7.7.1.848 nppiSub_32sc_C4IRSfs . . . . .	594
7.7.1.849 nppiSub_32sc_C4RSfs . . . . .	594
7.7.1.850 nppiSub_8u_AC4IRSfs . . . . .	595
7.7.1.851 nppiSub_8u_AC4RSfs . . . . .	595
7.7.1.852 nppiSub_8u_C1IRSfs . . . . .	596
7.7.1.853 nppiSub_8u_C1RSfs . . . . .	596
7.7.1.854 nppiSub_8u_C3IRSfs . . . . .	596
7.7.1.855 nppiSub_8u_C3RSfs . . . . .	597
7.7.1.856 nppiSub_8u_C4IRSfs . . . . .	597
7.7.1.857 nppiSub_8u_C4RSfs . . . . .	598
7.7.1.858 nppiSubC_16s_AC4IRSfs . . . . .	598
7.7.1.859 nppiSubC_16s_AC4RSfs . . . . .	598
7.7.1.860 nppiSubC_16s_C1IRSfs . . . . .	599
7.7.1.861 nppiSubC_16s_C1RSfs . . . . .	599
7.7.1.862 nppiSubC_16s_C3IRSfs . . . . .	600
7.7.1.863 nppiSubC_16s_C3RSfs . . . . .	600
7.7.1.864 nppiSubC_16s_C4IRSfs . . . . .	600
7.7.1.865 nppiSubC_16s_C4RSfs . . . . .	601
7.7.1.866 nppiSubC_16sc_AC4IRSfs . . . . .	601
7.7.1.867 nppiSubC_16sc_AC4RSfs . . . . .	601
7.7.1.868 nppiSubC_16sc_C1IRSfs . . . . .	602
7.7.1.869 nppiSubC_16sc_C1RSfs . . . . .	602
7.7.1.870 nppiSubC_16sc_C3IRSfs . . . . .	603
7.7.1.871 nppiSubC_16sc_C3RSfs . . . . .	603
7.7.1.872 nppiSubC_16sc_C4IRSfs . . . . .	603
7.7.1.873 nppiSubC_16sc_C4RSfs . . . . .	604
7.7.1.874 nppiSubC_16u_AC4IRSfs . . . . .	604
7.7.1.875 nppiSubC_16u_AC4RSfs . . . . .	605
7.7.1.876 nppiSubC_16u_C1IRSfs . . . . .	605
7.7.1.877 nppiSubC_16u_C1RSfs . . . . .	605
7.7.1.878 nppiSubC_16u_C3IRSfs . . . . .	606

7.7.1.879 nppiSubC_16u_C3RSfs . . . . .	606
7.7.1.880 nppiSubC_16u_C4IRSfs . . . . .	607
7.7.1.881 nppiSubC_16u_C4RSfs . . . . .	607
7.7.1.882 nppiSubC_32f_AC4IR . . . . .	607
7.7.1.883 nppiSubC_32f_AC4R . . . . .	608
7.7.1.884 nppiSubC_32f_C1IR . . . . .	608
7.7.1.885 nppiSubC_32f_C1R . . . . .	608
7.7.1.886 nppiSubC_32f_C3IR . . . . .	609
7.7.1.887 nppiSubC_32f_C3R . . . . .	609
7.7.1.888 nppiSubC_32f_C4IR . . . . .	609
7.7.1.889 nppiSubC_32f_C4R . . . . .	610
7.7.1.890 nppiSubC_32fc_AC4IR . . . . .	610
7.7.1.891 nppiSubC_32fc_AC4R . . . . .	610
7.7.1.892 nppiSubC_32fc_C1IR . . . . .	611
7.7.1.893 nppiSubC_32fc_C1R . . . . .	611
7.7.1.894 nppiSubC_32fc_C3IR . . . . .	611
7.7.1.895 nppiSubC_32fc_C3R . . . . .	612
7.7.1.896 nppiSubC_32fc_C4IR . . . . .	612
7.7.1.897 nppiSubC_32fc_C4R . . . . .	612
7.7.1.898 nppiSubC_32s_AC4IRSfs . . . . .	613
7.7.1.899 nppiSubC_32s_AC4RSfs . . . . .	613
7.7.1.900 nppiSubC_32s_C1IRSfs . . . . .	614
7.7.1.901 nppiSubC_32s_C1RSfs . . . . .	614
7.7.1.902 nppiSubC_32s_C3IRSfs . . . . .	614
7.7.1.903 nppiSubC_32s_C3RSfs . . . . .	615
7.7.1.904 nppiSubC_32s_C4IRSfs . . . . .	615
7.7.1.905 nppiSubC_32s_C4RSfs . . . . .	615
7.7.1.906 nppiSubC_32sc_AC4IRSfs . . . . .	616
7.7.1.907 nppiSubC_32sc_AC4RSfs . . . . .	616
7.7.1.908 nppiSubC_32sc_C1IRSfs . . . . .	617
7.7.1.909 nppiSubC_32sc_C1RSfs . . . . .	617
7.7.1.910 nppiSubC_32sc_C3IRSfs . . . . .	617
7.7.1.911 nppiSubC_32sc_C3RSfs . . . . .	618
7.7.1.912 nppiSubC_32sc_C4IRSfs . . . . .	618
7.7.1.913 nppiSubC_32sc_C4RSfs . . . . .	619
7.7.1.914 nppiSubC_8u_AC4IRSfs . . . . .	619

7.7.1.915 nppiSubC_8u_AC4RSfs . . . . .	619
7.7.1.916 nppiSubC_8u_C1IRSfs . . . . .	620
7.7.1.917 nppiSubC_8u_C1RSfs . . . . .	620
7.7.1.918 nppiSubC_8u_C3IRSfs . . . . .	621
7.7.1.919 nppiSubC_8u_C3RSfs . . . . .	621
7.7.1.920 nppiSubC_8u_C4IRSfs . . . . .	621
7.7.1.921 nppiSubC_8u_C4RSfs . . . . .	622
7.7.1.922 nppiXor_16u_AC4IR . . . . .	622
7.7.1.923 nppiXor_16u_AC4R . . . . .	622
7.7.1.924 nppiXor_16u_C1IR . . . . .	623
7.7.1.925 nppiXor_16u_C1R . . . . .	623
7.7.1.926 nppiXor_16u_C3IR . . . . .	624
7.7.1.927 nppiXor_16u_C3R . . . . .	624
7.7.1.928 nppiXor_16u_C4IR . . . . .	624
7.7.1.929 nppiXor_16u_C4R . . . . .	625
7.7.1.930 nppiXor_32s_AC4IR . . . . .	625
7.7.1.931 nppiXor_32s_AC4R . . . . .	625
7.7.1.932 nppiXor_32s_C1IR . . . . .	626
7.7.1.933 nppiXor_32s_C1R . . . . .	626
7.7.1.934 nppiXor_32s_C3IR . . . . .	627
7.7.1.935 nppiXor_32s_C3R . . . . .	627
7.7.1.936 nppiXor_32s_C4IR . . . . .	627
7.7.1.937 nppiXor_32s_C4R . . . . .	628
7.7.1.938 nppiXor_8u_AC4IR . . . . .	628
7.7.1.939 nppiXor_8u_AC4R . . . . .	628
7.7.1.940 nppiXor_8u_C1IR . . . . .	629
7.7.1.941 nppiXor_8u_C1R . . . . .	629
7.7.1.942 nppiXor_8u_C3IR . . . . .	630
7.7.1.943 nppiXor_8u_C3R . . . . .	630
7.7.1.944 nppiXor_8u_C4IR . . . . .	630
7.7.1.945 nppiXor_8u_C4R . . . . .	631
7.7.1.946 nppiXorC_16u_AC4IR . . . . .	631
7.7.1.947 nppiXorC_16u_AC4R . . . . .	631
7.7.1.948 nppiXorC_16u_C1IR . . . . .	632
7.7.1.949 nppiXorC_16u_C1R . . . . .	632
7.7.1.950 nppiXorC_16u_C3IR . . . . .	632

7.7.1.951	<a href="#">nppiXorC_16u_C3R</a>	633
7.7.1.952	<a href="#">nppiXorC_16u_C4IR</a>	633
7.7.1.953	<a href="#">nppiXorC_16u_C4R</a>	633
7.7.1.954	<a href="#">nppiXorC_32s_AC4IR</a>	634
7.7.1.955	<a href="#">nppiXorC_32s_AC4R</a>	634
7.7.1.956	<a href="#">nppiXorC_32s_C1IR</a>	634
7.7.1.957	<a href="#">nppiXorC_32s_C1R</a>	635
7.7.1.958	<a href="#">nppiXorC_32s_C3IR</a>	635
7.7.1.959	<a href="#">nppiXorC_32s_C3R</a>	635
7.7.1.960	<a href="#">nppiXorC_32s_C4IR</a>	636
7.7.1.961	<a href="#">nppiXorC_32s_C4R</a>	636
7.7.1.962	<a href="#">nppiXorC_8u_AC4IR</a>	636
7.7.1.963	<a href="#">nppiXorC_8u_AC4R</a>	637
7.7.1.964	<a href="#">nppiXorC_8u_C1IR</a>	637
7.7.1.965	<a href="#">nppiXorC_8u_C1R</a>	637
7.7.1.966	<a href="#">nppiXorC_8u_C3IR</a>	638
7.7.1.967	<a href="#">nppiXorC_8u_C3R</a>	638
7.7.1.968	<a href="#">nppiXorC_8u_C4IR</a>	638
7.7.1.969	<a href="#">nppiXorC_8u_C4R</a>	639
7.8	<a href="#">Threshold and Compare Operations</a>	640
7.8.1	<a href="#">Detailed Description</a>	640
7.8.2	<a href="#">Function Documentation</a>	640
7.8.2.1	<a href="#">nppiCompare_32f_C1R</a>	640
7.8.2.2	<a href="#">nppiCompare_8u_AC4R</a>	641
7.8.2.3	<a href="#">nppiCompare_8u_C4R</a>	641
7.8.2.4	<a href="#">nppiThreshold_32f_C1R</a>	642
7.8.2.5	<a href="#">nppiThreshold_8u_AC4R</a>	642
7.9	<a href="#">Compression</a>	644
7.9.1	<a href="#">Detailed Description</a>	644
7.9.2	<a href="#">Function Documentation</a>	644
7.9.2.1	<a href="#">nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R</a>	644
7.9.2.2	<a href="#">nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R</a>	645
7.9.2.3	<a href="#">nppiQuantFwdRawTableInit_JPEG_8u</a>	645
7.9.2.4	<a href="#">nppiQuantFwdTableInit_JPEG_8u16u</a>	646
7.9.2.5	<a href="#">nppiQuantInvTableInit_JPEG_8u16u</a>	646
7.10	<a href="#">Geometric Transforms</a>	647

7.10.1 Detailed Description . . . . .	665
7.10.2 Function Documentation . . . . .	665
7.10.2.1 nppiGetAffineBound . . . . .	665
7.10.2.2 nppiGetAffineQuad . . . . .	666
7.10.2.3 nppiGetAffineTransform . . . . .	666
7.10.2.4 nppiGetPerspectiveBound . . . . .	666
7.10.2.5 nppiGetPerspectiveQuad . . . . .	667
7.10.2.6 nppiGetPerspectiveTransform . . . . .	667
7.10.2.7 nppiGetRotateBound . . . . .	668
7.10.2.8 nppiGetRotateQuad . . . . .	668
7.10.2.9 nppiMirror_16u_AC4R . . . . .	668
7.10.2.10 nppiMirror_16u_C1R . . . . .	669
7.10.2.11 nppiMirror_16u_C3R . . . . .	669
7.10.2.12 nppiMirror_16u_C4R . . . . .	670
7.10.2.13 nppiMirror_32f_AC4R . . . . .	670
7.10.2.14 nppiMirror_32f_C1R . . . . .	670
7.10.2.15 nppiMirror_32f_C3R . . . . .	671
7.10.2.16 nppiMirror_32f_C4R . . . . .	671
7.10.2.17 nppiMirror_32s_AC4R . . . . .	672
7.10.2.18 nppiMirror_32s_C1R . . . . .	672
7.10.2.19 nppiMirror_32s_C3R . . . . .	672
7.10.2.20 nppiMirror_32s_C4R . . . . .	673
7.10.2.21 nppiMirror_8u_AC4R . . . . .	673
7.10.2.22 nppiMirror_8u_C1R . . . . .	674
7.10.2.23 nppiMirror_8u_C3R . . . . .	674
7.10.2.24 nppiMirror_8u_C4R . . . . .	674
7.10.2.25 nppiResize_8u_C1R . . . . .	675
7.10.2.26 nppiResize_8u_C4R . . . . .	675
7.10.2.27 nppiRotate_16u_AC4R . . . . .	676
7.10.2.28 nppiRotate_16u_C1R . . . . .	677
7.10.2.29 nppiRotate_16u_C3R . . . . .	677
7.10.2.30 nppiRotate_16u_C4R . . . . .	678
7.10.2.31 nppiRotate_32f_AC4R . . . . .	678
7.10.2.32 nppiRotate_32f_C1R . . . . .	679
7.10.2.33 nppiRotate_32f_C3R . . . . .	679
7.10.2.34 nppiRotate_32f_C4R . . . . .	680

7.10.2.35 nppiRotate_8u_AC4R . . . . .	680
7.10.2.36 nppiRotate_8u_C1R . . . . .	681
7.10.2.37 nppiRotate_8u_C3R . . . . .	681
7.10.2.38 nppiRotate_8u_C4R . . . . .	682
7.10.2.39 nppiWarpAffine_16u_AC4R . . . . .	682
7.10.2.40 nppiWarpAffine_16u_C1R . . . . .	682
7.10.2.41 nppiWarpAffine_16u_C3R . . . . .	683
7.10.2.42 nppiWarpAffine_16u_C4R . . . . .	684
7.10.2.43 nppiWarpAffine_16u_P3R . . . . .	684
7.10.2.44 nppiWarpAffine_16u_P4R . . . . .	684
7.10.2.45 nppiWarpAffine_32f_AC4R . . . . .	684
7.10.2.46 nppiWarpAffine_32f_C1R . . . . .	684
7.10.2.47 nppiWarpAffine_32f_C3R . . . . .	685
7.10.2.48 nppiWarpAffine_32f_C4R . . . . .	686
7.10.2.49 nppiWarpAffine_32f_P3R . . . . .	686
7.10.2.50 nppiWarpAffine_32f_P4R . . . . .	686
7.10.2.51 nppiWarpAffine_32s_AC4R . . . . .	686
7.10.2.52 nppiWarpAffine_32s_C1R . . . . .	686
7.10.2.53 nppiWarpAffine_32s_C3R . . . . .	687
7.10.2.54 nppiWarpAffine_32s_C4R . . . . .	688
7.10.2.55 nppiWarpAffine_32s_P3R . . . . .	688
7.10.2.56 nppiWarpAffine_32s_P4R . . . . .	688
7.10.2.57 nppiWarpAffine_8u_AC4R . . . . .	688
7.10.2.58 nppiWarpAffine_8u_C1R . . . . .	688
7.10.2.59 nppiWarpAffine_8u_C3R . . . . .	690
7.10.2.60 nppiWarpAffine_8u_C4R . . . . .	690
7.10.2.61 nppiWarpAffine_8u_P3R . . . . .	690
7.10.2.62 nppiWarpAffine_8u_P4R . . . . .	690
7.10.2.63 nppiWarpAffineBack_16u_AC4R . . . . .	690
7.10.2.64 nppiWarpAffineBack_16u_C1R . . . . .	691
7.10.2.65 nppiWarpAffineBack_16u_C3R . . . . .	692
7.10.2.66 nppiWarpAffineBack_16u_C4R . . . . .	692
7.10.2.67 nppiWarpAffineBack_16u_P3R . . . . .	692
7.10.2.68 nppiWarpAffineBack_16u_P4R . . . . .	692
7.10.2.69 nppiWarpAffineBack_32f_AC4R . . . . .	693
7.10.2.70 nppiWarpAffineBack_32f_C1R . . . . .	693



7.10.2.71	nppiWarpAffineBack_32f_C3R . . . . .	694
7.10.2.72	nppiWarpAffineBack_32f_C4R . . . . .	694
7.10.2.73	nppiWarpAffineBack_32f_P3R . . . . .	694
7.10.2.74	nppiWarpAffineBack_32f_P4R . . . . .	694
7.10.2.75	nppiWarpAffineBack_32s_AC4R . . . . .	695
7.10.2.76	nppiWarpAffineBack_32s_C1R . . . . .	695
7.10.2.77	nppiWarpAffineBack_32s_C3R . . . . .	696
7.10.2.78	nppiWarpAffineBack_32s_C4R . . . . .	696
7.10.2.79	nppiWarpAffineBack_32s_P3R . . . . .	696
7.10.2.80	nppiWarpAffineBack_32s_P4R . . . . .	696
7.10.2.81	nppiWarpAffineBack_8u_AC4R . . . . .	697
7.10.2.82	nppiWarpAffineBack_8u_C1R . . . . .	697
7.10.2.83	nppiWarpAffineBack_8u_C3R . . . . .	698
7.10.2.84	nppiWarpAffineBack_8u_C4R . . . . .	698
7.10.2.85	nppiWarpAffineBack_8u_P3R . . . . .	698
7.10.2.86	nppiWarpAffineBack_8u_P4R . . . . .	699
7.10.2.87	nppiWarpAffineQuad_16u_AC4R . . . . .	699
7.10.2.88	nppiWarpAffineQuad_16u_C1R . . . . .	699
7.10.2.89	nppiWarpAffineQuad_16u_C3R . . . . .	700
7.10.2.90	nppiWarpAffineQuad_16u_C4R . . . . .	700
7.10.2.91	nppiWarpAffineQuad_16u_P3R . . . . .	700
7.10.2.92	nppiWarpAffineQuad_16u_P4R . . . . .	701
7.10.2.93	nppiWarpAffineQuad_32f_AC4R . . . . .	701
7.10.2.94	nppiWarpAffineQuad_32f_C1R . . . . .	701
7.10.2.95	nppiWarpAffineQuad_32f_C3R . . . . .	702
7.10.2.96	nppiWarpAffineQuad_32f_C4R . . . . .	702
7.10.2.97	nppiWarpAffineQuad_32f_P3R . . . . .	702
7.10.2.98	nppiWarpAffineQuad_32f_P4R . . . . .	702
7.10.2.99	nppiWarpAffineQuad_32s_AC4R . . . . .	703
7.10.2.100	nppiWarpAffineQuad_32s_C1R . . . . .	703
7.10.2.101	nppiWarpAffineQuad_32s_C3R . . . . .	704
7.10.2.102	nppiWarpAffineQuad_32s_C4R . . . . .	704
7.10.2.103	nppiWarpAffineQuad_32s_P3R . . . . .	704
7.10.2.104	nppiWarpAffineQuad_32s_P4R . . . . .	704
7.10.2.105	nppiWarpAffineQuad_8u_AC4R . . . . .	704
7.10.2.106	nppiWarpAffineQuad_8u_C1R . . . . .	705

7.10.2.107nppiWarpAffineQuad_8u_C3R . . . . .	706
7.10.2.108nppiWarpAffineQuad_8u_C4R . . . . .	706
7.10.2.109nppiWarpAffineQuad_8u_P3R . . . . .	706
7.10.2.110nppiWarpAffineQuad_8u_P4R . . . . .	706
7.10.2.111nppiWarpPerspective_16u_AC4R . . . . .	706
7.10.2.112nppiWarpPerspective_16u_C1R . . . . .	707
7.10.2.113nppiWarpPerspective_16u_C3R . . . . .	708
7.10.2.114nppiWarpPerspective_16u_C4R . . . . .	708
7.10.2.115nppiWarpPerspective_16u_P3R . . . . .	708
7.10.2.116nppiWarpPerspective_16u_P4R . . . . .	708
7.10.2.117nppiWarpPerspective_32f_AC4R . . . . .	709
7.10.2.118nppiWarpPerspective_32f_C1R . . . . .	709
7.10.2.119nppiWarpPerspective_32f_C3R . . . . .	710
7.10.2.120nppiWarpPerspective_32f_C4R . . . . .	710
7.10.2.121nppiWarpPerspective_32f_P3R . . . . .	710
7.10.2.122nppiWarpPerspective_32f_P4R . . . . .	710
7.10.2.123nppiWarpPerspective_32s_AC4R . . . . .	711
7.10.2.124nppiWarpPerspective_32s_C1R . . . . .	711
7.10.2.125nppiWarpPerspective_32s_C3R . . . . .	712
7.10.2.126nppiWarpPerspective_32s_C4R . . . . .	712
7.10.2.127nppiWarpPerspective_32s_P3R . . . . .	712
7.10.2.128nppiWarpPerspective_32s_P4R . . . . .	712
7.10.2.129nppiWarpPerspective_8u_AC4R . . . . .	713
7.10.2.130nppiWarpPerspective_8u_C1R . . . . .	713
7.10.2.131nppiWarpPerspective_8u_C3R . . . . .	714
7.10.2.132nppiWarpPerspective_8u_C4R . . . . .	714
7.10.2.133nppiWarpPerspective_8u_P3R . . . . .	714
7.10.2.134nppiWarpPerspective_8u_P4R . . . . .	715
7.10.2.135nppiWarpPerspectiveBack_16u_AC4R . . . . .	715
7.10.2.136nppiWarpPerspectiveBack_16u_C1R . . . . .	715
7.10.2.137nppiWarpPerspectiveBack_16u_C3R . . . . .	716
7.10.2.138nppiWarpPerspectiveBack_16u_C4R . . . . .	716
7.10.2.139nppiWarpPerspectiveBack_16u_P3R . . . . .	717
7.10.2.140nppiWarpPerspectiveBack_16u_P4R . . . . .	717
7.10.2.141nppiWarpPerspectiveBack_32f_AC4R . . . . .	717
7.10.2.142nppiWarpPerspectiveBack_32f_C1R . . . . .	717

7.10.2.143nppiWarpPerspectiveBack_32f_C3R . . . . .	718
7.10.2.144nppiWarpPerspectiveBack_32f_C4R . . . . .	718
7.10.2.145nppiWarpPerspectiveBack_32f_P3R . . . . .	719
7.10.2.146nppiWarpPerspectiveBack_32f_P4R . . . . .	719
7.10.2.147nppiWarpPerspectiveBack_32s_AC4R . . . . .	719
7.10.2.148nppiWarpPerspectiveBack_32s_C1R . . . . .	719
7.10.2.149nppiWarpPerspectiveBack_32s_C3R . . . . .	720
7.10.2.150nppiWarpPerspectiveBack_32s_C4R . . . . .	720
7.10.2.151nppiWarpPerspectiveBack_32s_P3R . . . . .	721
7.10.2.152nppiWarpPerspectiveBack_32s_P4R . . . . .	721
7.10.2.153nppiWarpPerspectiveBack_8u_AC4R . . . . .	721
7.10.2.154nppiWarpPerspectiveBack_8u_C1R . . . . .	721
7.10.2.155nppiWarpPerspectiveBack_8u_C3R . . . . .	722
7.10.2.156nppiWarpPerspectiveBack_8u_C4R . . . . .	723
7.10.2.157nppiWarpPerspectiveBack_8u_P3R . . . . .	723
7.10.2.158nppiWarpPerspectiveBack_8u_P4R . . . . .	723
7.10.2.159nppiWarpPerspectiveQuad_16u_AC4R . . . . .	723
7.10.2.160nppiWarpPerspectiveQuad_16u_C1R . . . . .	723
7.10.2.161nppiWarpPerspectiveQuad_16u_C3R . . . . .	724
7.10.2.162nppiWarpPerspectiveQuad_16u_C4R . . . . .	725
7.10.2.163nppiWarpPerspectiveQuad_16u_P3R . . . . .	725
7.10.2.164nppiWarpPerspectiveQuad_16u_P4R . . . . .	725
7.10.2.165nppiWarpPerspectiveQuad_32f_AC4R . . . . .	725
7.10.2.166nppiWarpPerspectiveQuad_32f_C1R . . . . .	725
7.10.2.167nppiWarpPerspectiveQuad_32f_C3R . . . . .	726
7.10.2.168nppiWarpPerspectiveQuad_32f_C4R . . . . .	727
7.10.2.169nppiWarpPerspectiveQuad_32f_P3R . . . . .	727
7.10.2.170nppiWarpPerspectiveQuad_32f_P4R . . . . .	727
7.10.2.171nppiWarpPerspectiveQuad_32s_AC4R . . . . .	727
7.10.2.172nppiWarpPerspectiveQuad_32s_C1R . . . . .	727
7.10.2.173nppiWarpPerspectiveQuad_32s_C3R . . . . .	728
7.10.2.174nppiWarpPerspectiveQuad_32s_C4R . . . . .	728
7.10.2.175nppiWarpPerspectiveQuad_32s_P3R . . . . .	729
7.10.2.176nppiWarpPerspectiveQuad_32s_P4R . . . . .	729
7.10.2.177nppiWarpPerspectiveQuad_8u_AC4R . . . . .	729
7.10.2.178nppiWarpPerspectiveQuad_8u_C1R . . . . .	729

7.10.2.179	npplWarpPerspectiveQuad_8u_C3R	730
7.10.2.180	npplWarpPerspectiveQuad_8u_C4R	730
7.10.2.181	npplWarpPerspectiveQuad_8u_P3R	731
7.10.2.182	npplWarpPerspectiveQuad_8u_P4R	731
7.11	NPP Signal Processing	732
7.11.1	Function Documentation	770
7.11.1.1	nppl10Log10_32s_ISfs	770
7.11.1.2	nppl10Log10_32s_Sfs	770
7.11.1.3	npplAbs_16s	770
7.11.1.4	npplAbs_16s_I	771
7.11.1.5	npplAbs_32f	771
7.11.1.6	npplAbs_32f_I	771
7.11.1.7	npplAbs_32s	771
7.11.1.8	npplAbs_32s_I	772
7.11.1.9	npplAbs_64f	772
7.11.1.10	npplAbs_64f_I	772
7.11.1.11	npplAdd_16s	772
7.11.1.12	npplAdd_16s32f	773
7.11.1.13	npplAdd_16s32s_I	773
7.11.1.14	npplAdd_16s_I	773
7.11.1.15	npplAdd_16s_ISfs	774
7.11.1.16	npplAdd_16s_Sfs	774
7.11.1.17	npplAdd_16sc_ISfs	774
7.11.1.18	npplAdd_16sc_Sfs	775
7.11.1.19	npplAdd_16u	775
7.11.1.20	npplAdd_16u_ISfs	775
7.11.1.21	npplAdd_16u_Sfs	776
7.11.1.22	npplAdd_32f	776
7.11.1.23	npplAdd_32f_I	776
7.11.1.24	npplAdd_32fc	777
7.11.1.25	npplAdd_32fc_I	777
7.11.1.26	npplAdd_32s_ISfs	777
7.11.1.27	npplAdd_32s_Sfs	778
7.11.1.28	npplAdd_32sc_ISfs	778
7.11.1.29	npplAdd_32sc_Sfs	778
7.11.1.30	npplAdd_32u	779

7.11.1.31 nppsAdd_64f . . . . .	779
7.11.1.32 nppsAdd_64f_I . . . . .	779
7.11.1.33 nppsAdd_64fc . . . . .	780
7.11.1.34 nppsAdd_64fc_I . . . . .	780
7.11.1.35 nppsAdd_64s_Sfs . . . . .	780
7.11.1.36 nppsAdd_8u16u . . . . .	781
7.11.1.37 nppsAdd_8u_ISfs . . . . .	781
7.11.1.38 nppsAdd_8u_Sfs . . . . .	781
7.11.1.39 nppsAddC_16s_ISfs . . . . .	782
7.11.1.40 nppsAddC_16s_Sfs . . . . .	782
7.11.1.41 nppsAddC_16sc_ISfs . . . . .	782
7.11.1.42 nppsAddC_16sc_Sfs . . . . .	783
7.11.1.43 nppsAddC_16u_ISfs . . . . .	783
7.11.1.44 nppsAddC_16u_Sfs . . . . .	783
7.11.1.45 nppsAddC_32f . . . . .	784
7.11.1.46 nppsAddC_32f_I . . . . .	784
7.11.1.47 nppsAddC_32fc . . . . .	784
7.11.1.48 nppsAddC_32fc_I . . . . .	785
7.11.1.49 nppsAddC_32s_ISfs . . . . .	785
7.11.1.50 nppsAddC_32s_Sfs . . . . .	785
7.11.1.51 nppsAddC_32sc_ISfs . . . . .	786
7.11.1.52 nppsAddC_32sc_Sfs . . . . .	786
7.11.1.53 nppsAddC_64f . . . . .	786
7.11.1.54 nppsAddC_64f_I . . . . .	787
7.11.1.55 nppsAddC_64fc . . . . .	787
7.11.1.56 nppsAddC_64fc_I . . . . .	787
7.11.1.57 nppsAddC_8u_ISfs . . . . .	787
7.11.1.58 nppsAddC_8u_Sfs . . . . .	788
7.11.1.59 nppsAddProduct_16s32s_Sfs . . . . .	788
7.11.1.60 nppsAddProduct_16s_Sfs . . . . .	789
7.11.1.61 nppsAddProduct_32f . . . . .	789
7.11.1.62 nppsAddProduct_32fc . . . . .	789
7.11.1.63 nppsAddProduct_32s_Sfs . . . . .	790
7.11.1.64 nppsAddProduct_64f . . . . .	790
7.11.1.65 nppsAddProduct_64fc . . . . .	790
7.11.1.66 nppsAddProductC_16s_ISfs . . . . .	791

7.11.1.67 nppsAddProductC_16s_Sfs . . . . .	791
7.11.1.68 nppsAddProductC_16sc_ISfs . . . . .	791
7.11.1.69 nppsAddProductC_16sc_Sfs . . . . .	792
7.11.1.70 nppsAddProductC_16u_ISfs . . . . .	792
7.11.1.71 nppsAddProductC_16u_Sfs . . . . .	792
7.11.1.72 nppsAddProductC_32f . . . . .	793
7.11.1.73 nppsAddProductC_32f_I . . . . .	793
7.11.1.74 nppsAddProductC_32fc . . . . .	793
7.11.1.75 nppsAddProductC_32fc_I . . . . .	794
7.11.1.76 nppsAddProductC_32s_ISfs . . . . .	794
7.11.1.77 nppsAddProductC_32s_Sfs . . . . .	794
7.11.1.78 nppsAddProductC_32sc_ISfs . . . . .	795
7.11.1.79 nppsAddProductC_32sc_Sfs . . . . .	795
7.11.1.80 nppsAddProductC_64f . . . . .	795
7.11.1.81 nppsAddProductC_64f_I . . . . .	796
7.11.1.82 nppsAddProductC_64fc . . . . .	796
7.11.1.83 nppsAddProductC_64fc_I . . . . .	796
7.11.1.84 nppsAddProductC_8u_ISfs . . . . .	797
7.11.1.85 nppsAddProductC_8u_Sfs . . . . .	797
7.11.1.86 nppsAnd_16u . . . . .	797
7.11.1.87 nppsAnd_16u_I . . . . .	798
7.11.1.88 nppsAnd_32u . . . . .	798
7.11.1.89 nppsAnd_32u_I . . . . .	798
7.11.1.90 nppsAnd_8u . . . . .	798
7.11.1.91 nppsAnd_8u_I . . . . .	799
7.11.1.92 nppsAndC_16u . . . . .	799
7.11.1.93 nppsAndC_16u_I . . . . .	799
7.11.1.94 nppsAndC_32u . . . . .	800
7.11.1.95 nppsAndC_32u_I . . . . .	800
7.11.1.96 nppsAndC_8u . . . . .	800
7.11.1.97 nppsAndC_8u_I . . . . .	801
7.11.1.98 nppsArctan_32f . . . . .	801
7.11.1.99 nppsArctan_32f_I . . . . .	801
7.11.1.100 nppsArctan_64f . . . . .	801
7.11.1.101 nppsArctan_64f_I . . . . .	802
7.11.1.102 nppsCauchy_32f_I . . . . .	802

7.11.1.103nppsCauchyD_32f_I . . . . .	802
7.11.1.104nppsCauchyDD2_32f_I . . . . .	802
7.11.1.105nppsCopy_16s . . . . .	803
7.11.1.106nppsCopy_16sc . . . . .	803
7.11.1.107nppsCopy_32f . . . . .	803
7.11.1.108nppsCopy_32fc . . . . .	804
7.11.1.109nppsCopy_32s . . . . .	804
7.11.1.110nppsCopy_32sc . . . . .	804
7.11.1.111nppsCopy_64fc . . . . .	804
7.11.1.112nppsCopy_64s . . . . .	805
7.11.1.113nppsCopy_64sc . . . . .	805
7.11.1.114nppsCopy_8u . . . . .	805
7.11.1.115nppsCubrt_32f . . . . .	806
7.11.1.116nppsCubrt_32s16s_Sfs . . . . .	806
7.11.1.117nppsDiv_16s_ISfs . . . . .	806
7.11.1.118nppsDiv_16s_Sfs . . . . .	807
7.11.1.119nppsDiv_16sc_ISfs . . . . .	807
7.11.1.120nppsDiv_16sc_Sfs . . . . .	807
7.11.1.121nppsDiv_16u_ISfs . . . . .	808
7.11.1.122nppsDiv_16u_Sfs . . . . .	808
7.11.1.123nppsDiv_32f . . . . .	808
7.11.1.124nppsDiv_32f_I . . . . .	809
7.11.1.125nppsDiv_32fc . . . . .	809
7.11.1.126nppsDiv_32fc_I . . . . .	809
7.11.1.127nppsDiv_32s16s_Sfs . . . . .	809
7.11.1.128nppsDiv_32s_ISfs . . . . .	810
7.11.1.129nppsDiv_32s_Sfs . . . . .	810
7.11.1.130nppsDiv_64f . . . . .	811
7.11.1.131nppsDiv_64f_I . . . . .	811
7.11.1.132nppsDiv_64fc . . . . .	811
7.11.1.133nppsDiv_64fc_I . . . . .	812
7.11.1.134nppsDiv_8u_ISfs . . . . .	812
7.11.1.135nppsDiv_8u_Sfs . . . . .	812
7.11.1.136nppsDiv_Round_16s_ISfs . . . . .	813
7.11.1.137nppsDiv_Round_16s_Sfs . . . . .	813
7.11.1.138nppsDiv_Round_16u_ISfs . . . . .	813

7.11.1.139nppsDiv_Round_16u_Sfs . . . . .	814
7.11.1.140nppsDiv_Round_8u_ISfs . . . . .	814
7.11.1.141nppsDiv_Round_8u_Sfs . . . . .	814
7.11.1.142nppsDivC_16s_ISfs . . . . .	815
7.11.1.143nppsDivC_16s_Sfs . . . . .	815
7.11.1.144nppsDivC_16sc_ISfs . . . . .	815
7.11.1.145nppsDivC_16sc_Sfs . . . . .	816
7.11.1.146nppsDivC_16u_ISfs . . . . .	816
7.11.1.147nppsDivC_16u_Sfs . . . . .	816
7.11.1.148nppsDivC_32f . . . . .	817
7.11.1.149nppsDivC_32f_I . . . . .	817
7.11.1.150nppsDivC_32fc . . . . .	817
7.11.1.151nppsDivC_32fc_I . . . . .	818
7.11.1.152nppsDivC_32s_ISfs . . . . .	818
7.11.1.153nppsDivC_32s_Sfs . . . . .	818
7.11.1.154nppsDivC_32sc_ISfs . . . . .	819
7.11.1.155nppsDivC_32sc_Sfs . . . . .	819
7.11.1.156nppsDivC_64f . . . . .	819
7.11.1.157nppsDivC_64f_I . . . . .	820
7.11.1.158nppsDivC_64fc . . . . .	820
7.11.1.159nppsDivC_64fc_I . . . . .	820
7.11.1.160nppsDivC_8u_ISfs . . . . .	820
7.11.1.161nppsDivC_8u_Sfs . . . . .	821
7.11.1.162nppsDivCRev_16s . . . . .	821
7.11.1.163nppsDivCRev_16s_I . . . . .	822
7.11.1.164nppsDivCRev_16u . . . . .	822
7.11.1.165nppsDivCRev_16u_I . . . . .	822
7.11.1.166nppsDivCRev_32f . . . . .	822
7.11.1.167nppsDivCRev_32f_I . . . . .	823
7.11.1.168nppsDivCRev_32s . . . . .	823
7.11.1.169nppsDivCRev_32s_I . . . . .	823
7.11.1.170nppsDivCRev_64f . . . . .	824
7.11.1.171nppsDivCRev_64f_I . . . . .	824
7.11.1.172nppsDivCRev_8u . . . . .	824
7.11.1.173nppsDivCRev_8u_I . . . . .	825
7.11.1.174nppsExp_16s_ISfs . . . . .	825



7.11.1.175	nppsExp_16s_Sfs	825
7.11.1.176	nppsExp_32f	825
7.11.1.177	nppsExp_32f64f	826
7.11.1.178	nppsExp_32f_I	826
7.11.1.179	nppsExp_32s_ISfs	826
7.11.1.180	nppsExp_32s_Sfs	827
7.11.1.181	nppsExp_64f	827
7.11.1.182	nppsExp_64f_I	827
7.11.1.183	nppsExp_64s_ISfs	827
7.11.1.184	nppsExp_64s_Sfs	828
7.11.1.185	nppsFree	828
7.11.1.186	nppsIntegral_32s	828
7.11.1.187	nppsIntegralGetBufferSize_32s	828
7.11.1.188	nppsLn_16s_ISfs	828
7.11.1.189	nppsLn_16s_Sfs	829
7.11.1.190	nppsLn_32f	829
7.11.1.191	nppsLn_32f_I	829
7.11.1.192	nppsLn_32s16s_Sfs	829
7.11.1.193	nppsLn_32s_ISfs	830
7.11.1.194	nppsLn_32s_Sfs	830
7.11.1.195	nppsLn_64f	830
7.11.1.196	nppsLn_64f32f	831
7.11.1.197	nppsLn_64f_I	831
7.11.1.198	nppsLShiftC_16s	831
7.11.1.199	nppsLShiftC_16s_I	831
7.11.1.200	nppsLShiftC_16u	832
7.11.1.201	nppsLShiftC_16u_I	832
7.11.1.202	nppsLShiftC_32s	832
7.11.1.203	nppsLShiftC_32s_I	833
7.11.1.204	nppsLShiftC_32u	833
7.11.1.205	nppsLShiftC_32u_I	833
7.11.1.206	nppsLShiftC_8u	833
7.11.1.207	nppsLShiftC_8u_I	834
7.11.1.208	nppsMalloc_16s	834
7.11.1.209	nppsMalloc_16sc	834
7.11.1.210	nppsMalloc_16u	835

7.11.1.21	lnppsMalloc_32f	835
7.11.1.212	nppsMalloc_32fc	835
7.11.1.213	nppsMalloc_32s	835
7.11.1.214	nppsMalloc_32sc	836
7.11.1.215	nppsMalloc_32u	836
7.11.1.216	nppsMalloc_64f	836
7.11.1.217	nppsMalloc_64fc	836
7.11.1.218	nppsMalloc_64s	837
7.11.1.219	nppsMalloc_64sc	837
7.11.1.220	nppsMalloc_8u	837
7.11.1.22	lnppsMax_16s	837
7.11.1.222	nppsMax_32f	838
7.11.1.223	nppsMax_32s	838
7.11.1.224	nppsMax_64f	838
7.11.1.225	nppsMaxGetBufferSize_16s	839
7.11.1.226	nppsMaxGetBufferSize_32f	839
7.11.1.227	nppsMaxGetBufferSize_32s	839
7.11.1.228	nppsMaxGetBufferSize_64f	839
7.11.1.229	nppsMin_16s	840
7.11.1.230	nppsMin_32f	840
7.11.1.23	lnppsMin_32s	840
7.11.1.232	nppsMin_64f	841
7.11.1.233	nppsMinGetBufferSize_16s	841
7.11.1.234	nppsMinGetBufferSize_32f	841
7.11.1.235	nppsMinGetBufferSize_32s	842
7.11.1.236	nppsMinGetBufferSize_64f	842
7.11.1.237	nppsMinMax_16s	842
7.11.1.238	nppsMinMax_16u	843
7.11.1.239	nppsMinMax_32f	843
7.11.1.240	nppsMinMax_32s	843
7.11.1.24	lnppsMinMax_32u	844
7.11.1.242	nppsMinMax_64f	844
7.11.1.243	nppsMinMax_8u	844
7.11.1.244	nppsMinMaxGetBufferSize_16s	845
7.11.1.245	nppsMinMaxGetBufferSize_16u	845
7.11.1.246	nppsMinMaxGetBufferSize_32f	845

7.11.1.247nppsMinMaxGetBufferSize_32s . . . . .	845
7.11.1.248nppsMinMaxGetBufferSize_32u . . . . .	846
7.11.1.249nppsMinMaxGetBufferSize_64f . . . . .	846
7.11.1.250nppsMinMaxGetBufferSize_8u . . . . .	846
7.11.1.251nppsMul_16s . . . . .	846
7.11.1.252nppsMul_16s32f . . . . .	847
7.11.1.253nppsMul_16s32s_Sfs . . . . .	847
7.11.1.254nppsMul_16s_I . . . . .	847
7.11.1.255nppsMul_16s_ISfs . . . . .	848
7.11.1.256nppsMul_16s_Sfs . . . . .	848
7.11.1.257nppsMul_16sc_ISfs . . . . .	848
7.11.1.258nppsMul_16sc_Sfs . . . . .	849
7.11.1.259nppsMul_16u16s_Sfs . . . . .	849
7.11.1.260nppsMul_16u_ISfs . . . . .	849
7.11.1.261nppsMul_16u_Sfs . . . . .	850
7.11.1.262nppsMul_32f . . . . .	850
7.11.1.263nppsMul_32f32fc . . . . .	850
7.11.1.264nppsMul_32f32fc_I . . . . .	851
7.11.1.265nppsMul_32f_I . . . . .	851
7.11.1.266nppsMul_32fc . . . . .	851
7.11.1.267nppsMul_32fc_I . . . . .	852
7.11.1.268nppsMul_32s32sc_ISfs . . . . .	852
7.11.1.269nppsMul_32s32sc_Sfs . . . . .	852
7.11.1.270nppsMul_32s_ISfs . . . . .	853
7.11.1.271nppsMul_32s_Sfs . . . . .	853
7.11.1.272nppsMul_32sc_ISfs . . . . .	853
7.11.1.273nppsMul_32sc_Sfs . . . . .	854
7.11.1.274nppsMul_64f . . . . .	854
7.11.1.275nppsMul_64f_I . . . . .	854
7.11.1.276nppsMul_64fc . . . . .	855
7.11.1.277nppsMul_64fc_I . . . . .	855
7.11.1.278nppsMul_8u16u . . . . .	855
7.11.1.279nppsMul_8u_ISfs . . . . .	856
7.11.1.280nppsMul_8u_Sfs . . . . .	856
7.11.1.281nppsMul_Low_32s_Sfs . . . . .	856
7.11.1.282nppsMulC_16s_ISfs . . . . .	857

7.11.1.283nppsMulC_16s_Sfs . . . . .	857
7.11.1.284nppsMulC_16sc_ISfs . . . . .	857
7.11.1.285nppsMulC_16sc_Sfs . . . . .	858
7.11.1.286nppsMulC_16u_ISfs . . . . .	858
7.11.1.287nppsMulC_16u_Sfs . . . . .	858
7.11.1.288nppsMulC_32f . . . . .	859
7.11.1.289nppsMulC_32f16s_Sfs . . . . .	859
7.11.1.290nppsMulC_32f_I . . . . .	859
7.11.1.291nppsMulC_32fc . . . . .	860
7.11.1.292nppsMulC_32fc_I . . . . .	860
7.11.1.293nppsMulC_32s_ISfs . . . . .	860
7.11.1.294nppsMulC_32s_Sfs . . . . .	861
7.11.1.295nppsMulC_32sc_ISfs . . . . .	861
7.11.1.296nppsMulC_32sc_Sfs . . . . .	861
7.11.1.297nppsMulC_64f . . . . .	862
7.11.1.298nppsMulC_64f64s_ISfs . . . . .	862
7.11.1.299nppsMulC_64f_I . . . . .	862
7.11.1.300nppsMulC_64fc . . . . .	863
7.11.1.301nppsMulC_64fc_I . . . . .	863
7.11.1.302nppsMulC_8u_ISfs . . . . .	863
7.11.1.303nppsMulC_8u_Sfs . . . . .	864
7.11.1.304nppsMulC_Low_32f16s . . . . .	864
7.11.1.305nppsNormalize_16s_Sfs . . . . .	864
7.11.1.306nppsNormalize_16sc_Sfs . . . . .	865
7.11.1.307nppsNormalize_32f . . . . .	865
7.11.1.308nppsNormalize_32fc . . . . .	865
7.11.1.309nppsNormalize_64f . . . . .	866
7.11.1.310nppsNormalize_64fc . . . . .	866
7.11.1.311nppsNot_16u . . . . .	866
7.11.1.312nppsNot_16u_I . . . . .	867
7.11.1.313nppsNot_32u . . . . .	867
7.11.1.314nppsNot_32u_I . . . . .	867
7.11.1.315nppsNot_8u . . . . .	867
7.11.1.316nppsNot_8u_I . . . . .	868
7.11.1.317nppsOr_16u . . . . .	868
7.11.1.318nppsOr_16u_I . . . . .	868

7.11.1.319nppsOr_32u . . . . .	868
7.11.1.320nppsOr_32u_I . . . . .	869
7.11.1.321nppsOr_8u . . . . .	869
7.11.1.322nppsOr_8u_I . . . . .	869
7.11.1.323nppsOrC_16u . . . . .	870
7.11.1.324nppsOrC_16u_I . . . . .	870
7.11.1.325nppsOrC_32u . . . . .	870
7.11.1.326nppsOrC_32u_I . . . . .	871
7.11.1.327nppsOrC_8u . . . . .	871
7.11.1.328nppsOrC_8u_I . . . . .	871
7.11.1.329nppsRShiftC_16s . . . . .	871
7.11.1.330nppsRShiftC_16s_I . . . . .	872
7.11.1.331nppsRShiftC_16u . . . . .	872
7.11.1.332nppsRShiftC_16u_I . . . . .	872
7.11.1.333nppsRShiftC_32s . . . . .	873
7.11.1.334nppsRShiftC_32s_I . . . . .	873
7.11.1.335nppsRShiftC_32u . . . . .	873
7.11.1.336nppsRShiftC_32u_I . . . . .	874
7.11.1.337nppsRShiftC_8u . . . . .	874
7.11.1.338nppsRShiftC_8u_I . . . . .	874
7.11.1.339nppsSet_16s . . . . .	874
7.11.1.340nppsSet_16sc . . . . .	875
7.11.1.341nppsSet_32f . . . . .	875
7.11.1.342nppsSet_32fc . . . . .	875
7.11.1.343nppsSet_32s . . . . .	876
7.11.1.344nppsSet_32sc . . . . .	876
7.11.1.345nppsSet_64f . . . . .	876
7.11.1.346nppsSet_64fc . . . . .	876
7.11.1.347nppsSet_64s . . . . .	877
7.11.1.348nppsSet_64sc . . . . .	877
7.11.1.349nppsSet_8u . . . . .	877
7.11.1.350nppsSqr_16s_ISfs . . . . .	878
7.11.1.351nppsSqr_16s_Sfs . . . . .	878
7.11.1.352nppsSqr_16sc_ISfs . . . . .	878
7.11.1.353nppsSqr_16sc_Sfs . . . . .	878
7.11.1.354nppsSqr_16u_ISfs . . . . .	879

7.11.1.355nppsSqr_16u_Sfs . . . . .	879
7.11.1.356nppsSqr_32f . . . . .	879
7.11.1.357nppsSqr_32f_I . . . . .	880
7.11.1.358nppsSqr_32fc . . . . .	880
7.11.1.359nppsSqr_32fc_I . . . . .	880
7.11.1.360nppsSqr_64f . . . . .	880
7.11.1.361nppsSqr_64f_I . . . . .	881
7.11.1.362nppsSqr_64fc . . . . .	881
7.11.1.363nppsSqr_64fc_I . . . . .	881
7.11.1.364nppsSqr_8u_ISfs . . . . .	881
7.11.1.365nppsSqr_8u_Sfs . . . . .	882
7.11.1.366nppsSqrt_16s_ISfs . . . . .	882
7.11.1.367nppsSqrt_16s_Sfs . . . . .	882
7.11.1.368nppsSqrt_16sc_ISfs . . . . .	883
7.11.1.369nppsSqrt_16sc_Sfs . . . . .	883
7.11.1.370nppsSqrt_16u_ISfs . . . . .	883
7.11.1.371nppsSqrt_16u_Sfs . . . . .	883
7.11.1.372nppsSqrt_32f . . . . .	884
7.11.1.373nppsSqrt_32f_I . . . . .	884
7.11.1.374nppsSqrt_32fc . . . . .	884
7.11.1.375nppsSqrt_32fc_I . . . . .	885
7.11.1.376nppsSqrt_32s16s_Sfs . . . . .	885
7.11.1.377nppsSqrt_64f . . . . .	885
7.11.1.378nppsSqrt_64f_I . . . . .	885
7.11.1.379nppsSqrt_64fc . . . . .	886
7.11.1.380nppsSqrt_64fc_I . . . . .	886
7.11.1.381nppsSqrt_64s16s_Sfs . . . . .	886
7.11.1.382nppsSqrt_64s_ISfs . . . . .	886
7.11.1.383nppsSqrt_64s_Sfs . . . . .	887
7.11.1.384nppsSqrt_8u_ISfs . . . . .	887
7.11.1.385nppsSqrt_8u_Sfs . . . . .	887
7.11.1.386nppsSub_16s . . . . .	888
7.11.1.387nppsSub_16s32f . . . . .	888
7.11.1.388nppsSub_16s_I . . . . .	888
7.11.1.389nppsSub_16s_ISfs . . . . .	889
7.11.1.390nppsSub_16s_Sfs . . . . .	889

7.11.1.391nppsSub_16sc_ISfs . . . . .	889
7.11.1.392nppsSub_16sc_Sfs . . . . .	890
7.11.1.393nppsSub_16u_ISfs . . . . .	890
7.11.1.394nppsSub_16u_Sfs . . . . .	890
7.11.1.395nppsSub_32f . . . . .	891
7.11.1.396nppsSub_32f_I . . . . .	891
7.11.1.397nppsSub_32fc . . . . .	891
7.11.1.398nppsSub_32fc_I . . . . .	892
7.11.1.399nppsSub_32s_ISfs . . . . .	892
7.11.1.400nppsSub_32s_Sfs . . . . .	892
7.11.1.401nppsSub_32sc_ISfs . . . . .	893
7.11.1.402nppsSub_32sc_Sfs . . . . .	893
7.11.1.403nppsSub_64f . . . . .	893
7.11.1.404nppsSub_64f_I . . . . .	894
7.11.1.405nppsSub_64fc . . . . .	894
7.11.1.406nppsSub_64fc_I . . . . .	894
7.11.1.407nppsSub_8u_ISfs . . . . .	894
7.11.1.408nppsSub_8u_Sfs . . . . .	895
7.11.1.409nppsSubC_16s_ISfs . . . . .	895
7.11.1.410nppsSubC_16s_Sfs . . . . .	896
7.11.1.411nppsSubC_16sc_ISfs . . . . .	896
7.11.1.412nppsSubC_16sc_Sfs . . . . .	896
7.11.1.413nppsSubC_16u_ISfs . . . . .	897
7.11.1.414nppsSubC_16u_Sfs . . . . .	897
7.11.1.415nppsSubC_32f . . . . .	897
7.11.1.416nppsSubC_32f_I . . . . .	898
7.11.1.417nppsSubC_32fc . . . . .	898
7.11.1.418nppsSubC_32fc_I . . . . .	898
7.11.1.419nppsSubC_32s_ISfs . . . . .	898
7.11.1.420nppsSubC_32s_Sfs . . . . .	899
7.11.1.421nppsSubC_32sc_ISfs . . . . .	899
7.11.1.422nppsSubC_32sc_Sfs . . . . .	900
7.11.1.423nppsSubC_64f . . . . .	900
7.11.1.424nppsSubC_64f_I . . . . .	900
7.11.1.425nppsSubC_64fc . . . . .	901
7.11.1.426nppsSubC_64fc_I . . . . .	901

7.11.1.427nppsSubC_8u_ISfs . . . . .	901
7.11.1.428nppsSubC_8u_Sfs . . . . .	902
7.11.1.429nppsSubCRev_16s_ISfs . . . . .	902
7.11.1.430nppsSubCRev_16s_Sfs . . . . .	902
7.11.1.431nppsSubCRev_16sc_ISfs . . . . .	903
7.11.1.432nppsSubCRev_16sc_Sfs . . . . .	903
7.11.1.433nppsSubCRev_16u_ISfs . . . . .	903
7.11.1.434nppsSubCRev_16u_Sfs . . . . .	904
7.11.1.435nppsSubCRev_32f . . . . .	904
7.11.1.436nppsSubCRev_32f_I . . . . .	904
7.11.1.437nppsSubCRev_32fc . . . . .	905
7.11.1.438nppsSubCRev_32fc_I . . . . .	905
7.11.1.439nppsSubCRev_32s_ISfs . . . . .	905
7.11.1.440nppsSubCRev_32s_Sfs . . . . .	906
7.11.1.441nppsSubCRev_32sc_ISfs . . . . .	906
7.11.1.442nppsSubCRev_32sc_Sfs . . . . .	906
7.11.1.443nppsSubCRev_64f . . . . .	907
7.11.1.444nppsSubCRev_64f_I . . . . .	907
7.11.1.445nppsSubCRev_64fc . . . . .	907
7.11.1.446nppsSubCRev_64fc_I . . . . .	908
7.11.1.447nppsSubCRev_8u_ISfs . . . . .	908
7.11.1.448nppsSubCRev_8u_Sfs . . . . .	908
7.11.1.449nppsSum_16s32s_Sfs . . . . .	909
7.11.1.450nppsSum_16s_Sfs . . . . .	909
7.11.1.451nppsSum_16sc32sc_Sfs . . . . .	909
7.11.1.452nppsSum_16sc_Sfs . . . . .	910
7.11.1.453nppsSum_32f . . . . .	910
7.11.1.454nppsSum_32fc . . . . .	910
7.11.1.455nppsSum_32s_Sfs . . . . .	911
7.11.1.456nppsSum_64f . . . . .	911
7.11.1.457nppsSum_64fc . . . . .	911
7.11.1.458nppsSumGetBufferSize_16s32s_Sfs . . . . .	912
7.11.1.459nppsSumGetBufferSize_16s_Sfs . . . . .	912
7.11.1.460nppsSumGetBufferSize_16sc32sc_Sfs . . . . .	912
7.11.1.461nppsSumGetBufferSize_16sc_Sfs . . . . .	912
7.11.1.462nppsSumGetBufferSize_32f . . . . .	913



7.11.1.463	nppsSumGetBufferSize_32fc	913
7.11.1.464	nppsSumGetBufferSize_32s_Sfs	913
7.11.1.465	nppsSumGetBufferSize_64f	914
7.11.1.466	nppsSumGetBufferSize_64fc	914
7.11.1.467	nppsXor_16u	914
7.11.1.468	nppsXor_16u_I	915
7.11.1.469	nppsXor_32u	915
7.11.1.470	nppsXor_32u_I	915
7.11.1.471	nppsXor_8u	915
7.11.1.472	nppsXor_8u_I	916
7.11.1.473	nppsXorC_16u	916
7.11.1.474	nppsXorC_16u_I	916
7.11.1.475	nppsXorC_32u	917
7.11.1.476	nppsXorC_32u_I	917
7.11.1.477	nppsXorC_8u	917
7.11.1.478	nppsXorC_8u_I	918
7.11.1.479	nppsZero_16s	918
7.11.1.480	nppsZero_16sc	918
7.11.1.481	nppsZero_32f	918
7.11.1.482	nppsZero_32fc	919
7.11.1.483	nppsZero_32s	919
7.11.1.484	nppsZero_32sc	919
7.11.1.485	nppsZero_64f	919
7.11.1.486	nppsZero_64fc	920
7.11.1.487	nppsZero_64s	920
7.11.1.488	nppsZero_64sc	920
7.11.1.489	nppsZero_8u	920
<b>8</b>	<b>Data Structure Documentation</b>	<b>921</b>
8.1	Npp16sc Struct Reference	921
8.1.1	Detailed Description	921
8.1.2	Field Documentation	921
8.1.2.1	im	921
8.1.2.2	re	921
8.2	Npp32fc Struct Reference	922
8.2.1	Detailed Description	922
8.2.2	Field Documentation	922

8.2.2.1	im	922
8.2.2.2	re	922
8.3	Npp32sc Struct Reference	923
8.3.1	Detailed Description	923
8.3.2	Field Documentation	923
8.3.2.1	im	923
8.3.2.2	re	923
8.4	Npp64fc Struct Reference	924
8.4.1	Detailed Description	924
8.4.2	Field Documentation	924
8.4.2.1	im	924
8.4.2.2	re	924
8.5	Npp64sc Struct Reference	925
8.5.1	Detailed Description	925
8.5.2	Field Documentation	925
8.5.2.1	im	925
8.5.2.2	re	925
8.6	NppiHaarBuffer Struct Reference	926
8.6.1	Field Documentation	926
8.6.1.1	haarBuffer	926
8.6.1.2	haarBufferSize	926
8.7	NppiHaarClassifier_32f Struct Reference	927
8.7.1	Field Documentation	927
8.7.1.1	classifiers	927
8.7.1.2	classifierSize	927
8.7.1.3	classifierStep	927
8.7.1.4	counterDevice	927
8.7.1.5	numClassifiers	927
8.8	NppiPoint Struct Reference	928
8.8.1	Detailed Description	928
8.8.2	Field Documentation	928
8.8.2.1	x	928
8.8.2.2	y	928
8.9	NppiRect Struct Reference	929
8.9.1	Detailed Description	929
8.9.2	Field Documentation	929

---

8.9.2.1	height	929
8.9.2.2	width	929
8.9.2.3	x	929
8.9.2.4	y	929
8.10	NppiSize Struct Reference	930
8.10.1	Detailed Description	930
8.10.2	Field Documentation	930
8.10.2.1	height	930
8.10.2.2	width	930
8.11	NppLibraryVersion Struct Reference	931
8.11.1	Field Documentation	931
8.11.1.1	build	931
8.11.1.2	major	931
8.11.1.3	minor	931



# Chapter 1

## NVIDIA Performance Primitives

### 1.1 What is NPP?

NVIDIA NPP is a library of functions for performing CUDA accelerated processing. The initial set of functionality in the library focuses on imaging and video processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

NPP can be used in one of two ways:

- A stand-alone library for adding GPU acceleration to an application with minimal effort. Using this route allows developers to add GPU acceleration to their applications in a matter of hours.
- A cooperative library for interoperating with a developer's GPU code efficiently.

Either route allows developers to harness the massive compute resources of NVIDIA GPUs, while simultaneously reducing development times.

### 1.2 Documentation

- [General API Conventions](#)
- [Signal-Processing Specific API Conventions](#)
- [Imaging-Processing Specific API Conventions](#)

### 1.3 Technical Specifications

Supported Platforms:

- Microsoft Windows 7 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Microsoft Windows XP (64-bit and 32-bit)
- Linux (Centos & Ubuntu) (64-bit and 32-bit)
- Mac OS X

## 1.4 Files

NPP is comprises the following files:

### 1.4.1 Header Files

- [npp.h](#)
- [nppcore.h](#)
- [nppdefs.h](#)
- [nppi.h](#)
- [npps.h](#)
- [nppversion.h](#)

All those header files are located in the CUDA Toolkit's

`/include/`

directory.

### 1.4.2 Library Files

On the Windows platform the NPP stub library is found in the CUDA Toolkit's library directory:

`/lib/npp.lib`

The matching DLL is located in the CUDA Toolkit's binary directory:

```
/bin/npp32_32_7.dll      // Dynamic library for 32-bit Windows.  
/bin/npp64_32_7.dll     // Dynamic library for 64-bit Windows.
```

On Linux and Mac platforms the dynamic libraries are located in the lib directory

```
/lib/libnpp32.so.3.2.9   // NPP 32-bit dynamic library for Linux  
/lib/libnpp64.so.3.2.9  // NPP 64-bit dynamic library for Linux  
  
/lib/libnpp32.3.2.dylib  // NPP 32-bit dynamic library for Mac  
/lib/libnpp64.3.2.dylib  // NPP 64-bit dynamic library for Mac
```

## 1.5 Supported NVIDIA Hardware

NPP runs on all CUDA capable NVIDIA hardware. For details please see [http://www.nvidia.com/object/cuda\\_learn\\_products.html](http://www.nvidia.com/object/cuda_learn_products.html)

## **Chapter 2**

# **General API Conventions**

## 2.1 Memory Management

The design of all the NPP functions follows the same guidelines as other NVIDIA CUDA libraries like cuFFT and cuBLAS. That is that all pointer arguments in those APIs are device pointers.

This convention enables the individual developer to make smart choices about memory management that minimize the number of memory transfers. It also allows the user the maximum flexibility regarding which of the various memory transfer mechanisms offered by the CUDA runtime is used, e.g. synchronous or asynchronous memory transfers, zero-copy and pinned memory, etc.

The most basic steps involved in using NPP for processing data is as follows:

1. Transfer input data from the host to device using

```
cudaMemcpy(...)
```

2. Process data using one or several NPP functions or custom CUDA kernels
3. Transfer the result data from the device to the host using

```
cudaMemcpy(...)
```

### 2.1.1 Scratch Buffer and Host Pointer

Some primitives of NPP require additional device memory buffers (scratch buffers) for calculations, e.g. signal and image reductions (Sum, Max, Min, MinMax, etc.). In order to give the NPP user maximum control regarding memory allocations and performance, it is the user's responsibility to allocate and delete those temporary buffers. For one this has the benefit that the library will not allocate memory unbeknownst to the user. It also allows developers who invoke the same primitive repeatedly to allocate the scratch only once, improving performance and potential device-memory fragmentation.

Scratch-buffer memory is unstructured and may be passed to the primitive in uninitialized form. This allows for reuse of the same scratch buffers with any primitive require scratch memory, as long as it is sufficiently sized.

The minimum scratch-buffer size for a given primitive (e.g. [nppsSum\\_32f\(\)](#)) can be obtained by a companion function (e.g. [nppsSumGetBufferSize\\_32f\(\)](#)). The buffer size is returned via a host pointer as allocation of the scratch-buffer is performed via CUDA runtime host code.

An example to invoke Sum primitive and allocate and free the necessary scratch memory:

```
...
// Compute the appropriate size of the scratch-memory buffer
int nBufferSize;
nppsSumGetBufferSize_32f(nLength, &nBufferSize);
// Allocate the scratch buffer
Npp8u * pDeviceBuffer;
cudaMalloc((void **)&pDeviceBuffer, nBufferSize);
// Call the primitive with the scratch buffer
nppsSum_32f(pSrc, nLength, pSum, nppAlgHintNone, pDeviceBuffer);
// Free the scratch buffer
cudaFree(pDeviceBuffer);
...
```

## 2.2 Function Naming

Since NPP is a C API and therefore does not allow for function overloading for different data-types the NPP naming convention addresses the need to differentiate between different flavors of the same algorithm



or primitive function but for various data types. This disambiguation of different flavors of a primitive is done via a suffix containing data type and other disambiguating information.

In addition to the flavor suffix, all NPP functions are prefixed with by the letters "npp". Primitives belonging to NPP's image-processing module add the letter "i" to the npp prefix, i.e. are prefixed by "nppi". Similarly signal-processing primitives are prefixed with "npps".

The general naming scheme is:

`npp<module info><PrimitiveName>_<data-type info>[_<additional flavor info>](<parameter list>)`

The data-type information uses the same names as the [Basic NPP Data Types](#). For example the data-type information "8u" would imply that the primitive operates on [Npp8u](#) data.

If a primitive consumes different type data from what it produces, both types will be listed in the order of consumed to produced data type.

Details about the "additional flavor information" is provided for each of the NPP modules, since each problem domain uses different flavor information suffixes.

## 2.3 Integer Result Scaling

NPP signal processing and imaging primitives often operate on integer data. This integer data is a usually a fixed point fractional representation of some physical magnitue (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend produce results exceeding the original fixed-point range if treated as regular integers.

In cases where the results exceed the original range, these functions clamp the result values back to the valid range. E.g. the maximum positive value for a 16-bit unsigned integer is 32767. A multiplication operation of  $4 * 10000 = 40000$  would exceed this range. The result would be clamped to be 32767.

To avoid the level of lost information due to clamping most integer primitives allow for result scaling. Primitives with result scaling have the "Sfs" suffix in their name and provide a parameter "nScaleFactor" that controls the amount of scaling. Before the results of an operation are clamped to the valid output-data range by multiplying them with  $2^{-nScaleFactor}$ .

Example: The primitive [nppsSqr\\_8u\\_Sfs\(\)](#) computes the square of 8-bit unsigned sample values in a signal (1D array of values). The maximum value of a 8-bit value is 255. The square of  $255^2 = 65025$  which would be clamped to 255 if no result scaling is performed. In order to map the maximum value of 255 to 255 in the result, one would specify an integer result scaling factor of 8, i.e. multiply each result with  $2^{-8} = \frac{1}{2^8} = \frac{1}{256}$ . The final result for a signal value of 255 being squared and scaled would be:

$$255^2 \cdot 2^{-8} = 254.00390625$$

which would be rounded to a final result of 254.

A medium gray value of 128 would result in

$$128^2 \cdot 2^{-8} = 64$$



## **Chapter 3**

# **Signal-Processing Specific API Conventions**

## 3.1 Signal Data

Signal data is passed to and from NPPS primitives via a pointer to the signal's data type.

The general idea behind this fairly low-level way of passing signal data is ease-of-adoption into existing software projects:

- Passing the data pointer rather than a higher-level signal struct allows for easy adoption by not requiring a specific signal representation (that could include total signal size offset, or other additional information). This avoids awkward packing and unpacking of signal data from the host application to an NPP specific signal representation.

### 3.1.1 Parameter Names for Signal Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

Those are signals consumed by the algorithm.

#### 3.1.1.1 Source Signal Pointer

The source signal data is generally passed via a pointer named

`pSrc`

The source signal pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppsPrimitive_32s(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pSrc1, pSrc2, ...`

#### 3.1.1.2 Destination Signal Pointer

The destination signal data is generally passed via a pointer named

`pDst`

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pDst1, pDst2, ...`

#### 3.1.1.3 In-Place Signal Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place signal data are called:

`pSrcDst`

### 3.1.2 Signal Data Alignment Requirements

NPP requires signal sample data to be naturally aligned, i.e. any pointer

```
NppType * p;
```

to a sample in a signal needs to fulfill:

```
assert(p % sizeof(p) == 0);
```

### 3.1.3 Signal Data Related Error Codes

All NPPI primitives operating on signal data validate the signal-data pointer for proper alignment and test that the point is not null.

Failed validation results in one of the following error codes being returned and the primitive not being executed:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if the image-data pointer is 0 (NULL).
- [NPP\\_ALIGNMENT\\_ERROR](#) if the signal-data pointer address is not a multiple of the signal's data-type size.

## 3.2 Signal Length

The vast majority of NPPS functions take a

```
nLength
```

parameter that tells the primitive how many of the signal's samples starting from the given data pointer are to be processed.

### 3.2.1 Length Related Error Codes

All NPPS primitives taking a length parameter validate this input.

Failed validation results in the following error code being returned and the primitive not being executed:

- [NPP\\_SIZE\\_ERROR](#) is returned if the length is negative.



## **Chapter 4**

# **Imaging-Processing Specific API Conventions**

## 4.1 Function Naming

Image processing related functions use a number of suffixes to indicate various different flavors of a primitive beyond just different data types. The flavor suffix uses the following abbreviations:

- "A" if the image is a 4 channel image this indicates the result alpha channel is not affected by the primitive.
- "Cn" the image consists of n channel packed pixels, where n can be 1, 2, 3 or 4.
- "Pn" the image consists of n separate image planes, where n can be 1, 2, 3 or 4.
- "C" (following the channel information) indicates that the primitive only operates on one of the color channels, the "channel-of-interest". All other output channels are not affected by the primitive.
- "I" indicates that the primitive works "in-place". In this case the image-data pointer is usually named "pSrcDst" to indicate that the image data serves as source and destination at the same time.
- "M" indicates "masked operation". These types of primitives have an additional "mask image" as input. Each pixel in the destination image corresponds to a pixel in the mask image. Only pixels with a corresponding non-zero mask pixel are being processed.
- "R" indicates the primitive operates only on a rectangular "region-of-interest" or "ROI". All ROI primitives take an additional input parameter of type [NppiSize](#), which specifies the width and height of the rectangular region that the primitive should process. For details on how primitives operate on ROIs see: [Region-of-Interest \(ROI\)](#).
- "Sfs" indicates the result values are processed by fixed scaling and saturation before they're written out.

The suffixes above always appear in alphabetical order. E.g. a 4 channel primitive not affecting the alpha channel with masked operation, in place and with scaling/saturation and ROI would have the postfix: "AC4IMRSfs".

## 4.2 Image Data

Image data is passed to and from NPPI primitives via a pair of parameters:

1. A pointer to the image's underlying data type.
2. A line step in bytes (also sometimes called line stride).

The general idea behind this fairly low-level way of passing image data is ease-of-adoption into existing software projects:

- Passing a raw pointer to the underlying pixel data type, rather than structured (by color) channel pixel data allows usage of the function in a wide variety of situations avoiding risky type cast or expensive image data copies.
- Passing the data pointer and line step individually rather than a higher-level image struct again allows for easy adoption by not requiring a specific image representation and thus avoiding awkward packing and unpacking of image data from the host application to an NPP specific image representation.



### 4.2.1 Line Step

The line step (also called "line stride" or "row step") allows lines of oddly sized images to start on well-aligned addresses by adding a number of unused bytes at the ends of the lines. This type of line padding has been common practice in digital image processing for a long time and is not particular to GPU image processing.

The line step is the number of bytes in a line **including the padding**. An other way to interpret this number is to say that it is the number of bytes between the first pixel of successive rows in the image, or generally the number of bytes between two neighboring pixels in any column of pixels.

The general reason for the existence of the line step it is that uniformly aligned rows of pixel enable optimizations of memory-access patterns.

Even though all functions in NPP will work with arbitrarily aligned images, best performance can only be achieved with well aligned image data. Any image data allocated with the NPP image allocators or the 2D memory allocators in the CUDA runtime, is well aligned.

Particularly on older CUDA capable GPUs it is likely that the performance decrease for misaligned data is substantial (orders of magnitude).

All image data passed to NPPI primitives requires a line step to be provided. It is important to keep in mind that this line step is always specified in terms of bytes, not pixels.

### 4.2.2 Parameter Names for Image Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

#### 4.2.2.1 Passing Source-Image Data

Those are images consumed by the algorithm.

##### 4.2.2.1.1 Source-Image Pointer

The source image data is generally passed via a pointer named

```
pSrc
```

The source image pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppiPrimitive_32s_C1R(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

##### 4.2.2.1.2 Source-Image Line Step

The source-image line step is the number of bytes between successive rows in the image. The source-image line step parameter is

```
nSrcStep
```

or in the case of multiple source images

```
nSrcStep1, nSrcStep2, ...
```

#### **4.2.2.2 Passing Destination-Image Data**

Those are images produced by the algorithm.

##### **4.2.2.2.1 Destination-Image Pointer**

The destination image data is generally passed via a pointer named

```
pDst
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

```
pDst1, pDst2, ...
```

##### **4.2.2.2.2 Destination-Image Line Step**

The destination-image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

```
nDstStep1, nDstStep2, ...
```

#### **4.2.2.3 Passing In-Place Image Data**

##### **4.2.2.3.1 In-Place Image Pointer**

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place image data are called:

```
pSrcDst
```

##### **4.2.2.3.2 In-Place-Image Line Step**

The in-place line step parameter is

```
nSrcDstStep
```

#### **4.2.2.4 Passing Mask-Image Data**

Some image processing primitives have variants supporting [Masked Operation](#).

#### 4.2.2.4.1 Mask-Image Pointer

The mask-image data is generally passed via a pointer named

`pMask`

#### 4.2.2.4.2 Mask-Image Line Step

The mask-image line step parameter is

`nMaskStep`

### 4.2.3 Image Data Alignment Requirements

NPP requires pixel data to adhere to certain alignment constraints: For 2 and 4 channel images the following alignment requirement holds: `data_pointer % (#channels * sizeof(channel type)) == 0`. E.g. a 4 channel image with underlying type `Npp8u` (8-bit unsigned) would require all pixels to fall on addresses that are multiples of 4 (4 channels \* 1 byte size).

As a logical consequence of all pixels being aligned to their natural size the image line steps of 2 and 4 channel images also need to be multiples of the pixel size.

1 and 3 channel images only require that pixel pointers are aligned to the underlying data type, i.e. `pData % sizeof(data type) == 0`. And consequentially line steps are also held to this requirement.

### 4.2.4 Image Data Related Error Codes

All NPPI primitives operating on image data validate the image-data pointer for proper alignment and test that the point is not null. They also validate the line stride for proper alignment and guard against the step being less or equal to 0. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_STEP_ERROR` is returned if the data step is 0 or negative.
- `NPP_NOT_EVEN_STEP_ERROR` is returned if the line step is not a multiple of the pixel size for 2 and 4 channel images.
- `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL).
- `NPP_ALIGNMENT_ERROR` if the image-data pointer address is not a multiple of the pixel size for 2 and 4 channel images.

## 4.3 Region-of-Interest (ROI)

In practice processing a rectangular sub-region of an image is often more common than processing complete images. The vast majority of NPP's image-processing primitives allow for processing of such sub regions also referred to as regions-of-interest or ROIs.

All primitives supporting ROI processing are marked by a "R" in their name suffix. Where possible, the ROI a primitive operates on is passed as a single `NppiSize` struct, which provides the with and height of the ROI. This raises the obvious question how the primitive knows where in the image this rectangle of (width, height) is located. The "start pixel" of the ROI is implicitly given by the image-data pointer. I.e. instead

of explicitly passing a pixel coordinate for the upper-right corner of the ROI the primitive's user needs to perform the necessary offset computation on the image data pointers, such that the pointers passed to the primitive thus point to the start of the ROI.

In practice this means that for an image (pSrc, nSrcStep) and the start-pixel of the ROI being given by (xROI, yROI), one would pass

```
pSrcOffset = pSrc + yROI * nSrcStep + xROI * PixelSize;
```

as the image-data source to the primitive. PixelSize is typically computed as

```
PixelSize = NumberOfColorChannels * sizeof(PixelDataType).
```

E.g. for a primitive like `nppiSet_16s_C4R()` we would have

- `NumberOfColorChannels == 4;`
- `sizeof(Npp16s) == 2;`
- and thus `PixelSize = 4 * 2 = 8;`

### 4.3.1 ROI Related Error Codes

All NPPI primitives operating on ROIs of image data validate the ROI size and image's step size. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_SIZE_ERROR` is returned if either the ROI width or ROI height are negative.
- `NPP_STEP_ERROR` is returned if the ROI width exceeds the image's line step. In mathematical terms  $(\text{widthROI} * \text{PixelSize}) > \text{nLinStep}$  indicates an error.

## 4.4 Masked Operation

Some primitive support masked operation. An "M" in the suffix of those variants indicates masked operation. Primitives supporting masked operation consume an additional input image provided via a [Mask-Image Pointer](#) and [Mask-Image Line Step](#). The mask image is interpreted by these primitives as a boolean image. The values of type `Npp8u` are interpreted as boolean values where a values of 0 indicates false, any non-zero values true.

Unless otherwise indicated the operation is only performed on pixels where its spatially corresponding mask pixel is true (non-zero). E.g. a masked copy operation would only copy those pixels in the ROI that have corresponding non-zero mask pixels.

## 4.5 Channel-of-Interest API

Some primitives allow restricting operations to a single channel of interest within a multi-channel image. These primitives are suffixed with the letter "C" (after the channel information, e.g. `nppiCopy_8u_C3CR(...)`). The channel-of-interest is generally selected by offsetting the image-data pointer to point directly to the channel- of-interest rather than the base of the first pixel in the ROI.

### 4.5.1 Select-Channel Source-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the source image. E.g. if pSrc is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy

primitive one could copy the second channel of this source image into the first channel of a destination image given by pDst by offsetting the pointer by two data items:

```
nppiCopy_8u_C3CR(pSrc + 2, nSrcStep, pDst, nDstStep, oSizeROI);
```

### 4.5.2 Select-Channel Destination-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the destination image. E.g. if pDst is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy data into the second channel of this destination image from the first channel of a source image given by pSrc by offsetting the destination pointer by two data items:

```
nppiCopy_8u_C3CR(pSrc, nSrcStep, pDst + 2, nDstStep, oSizeROI);
```

## 4.6 Geometric Transform API Specifics

This section covers some of the unique API features common to the geometric transform primitives.

### 4.6.1 Geometric Transforms and ROIs

Geometric transforms operate on source and destination ROIs. The way these ROIs affect the processing of pixels differs from other (non geometric) image-processing primitives: Only pixels in the intersection of the destination ROI and the transformed source ROI are being processed.

The typical processing proceeds as follows:

1. Transform the rectangular source ROI (given in source image coordinates) into the destination image space. This yields a quadrilateral.
2. Write only pixels in the intersection of the transformed source ROI and the destination ROI.

### 4.6.2 Pixel Interpolation

The majority of image geometry transform operation need to perform a resampling of the source image as source and destination pixels are not coincident.

NPP supports the following pixel interpolation modes (in order from fastest to slowest and lowest to highest quality):

- nearest neighbor
- linear interpolation
- cubic convolution
- supersampling
- interpolation using Lanczos window function

### 4.6.3 Rotate specific Error Codes

- [NPP\\_INTERPOLATION\\_ERROR](#) if eInterpolation has an illegal value.
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1.
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) indicates an error condition if srcROIrect has no intersection with the source image.
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) indicates a warning that no operation is performed if the transformed source ROI does not intersect the destination ROI.

# Chapter 5

## Module Index

### 5.1 Modules

Here is a list of all modules:

NPP Core . . . . .	23
NPP Type Definitions and Constants . . . . .	26
Basic NPP Data Types . . . . .	35
NPP Image Processing . . . . .	37
Memory Management . . . . .	92
Data-Exchange and Initialization . . . . .	104
Arithmetic and Logical Operations . . . . .	186
Threshold and Compare Operations . . . . .	640
Compression . . . . .	644
Geometric Transforms . . . . .	647
NPP Signal Processing . . . . .	732





## Chapter 6

# Data Structure Index

### 6.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">Npp16sc</a> (Complex Number This struct represents a short complex number ) . . . . .	921
<a href="#">Npp32fc</a> (Complex Number This struct represents a single floating-point complex number ) . . .	922
<a href="#">Npp32sc</a> (Complex Number This struct represents a signed int complex number ) . . . . .	923
<a href="#">Npp64fc</a> (Complex Number This struct represents a double floating-point complex number ) . .	924
<a href="#">Npp64sc</a> (Complex Number This struct represents a long long complex number ) . . . . .	925
<a href="#">NppiHaarBuffer</a> . . . . .	926
<a href="#">NppiHaarClassifier_32f</a> . . . . .	927
<a href="#">NppiPoint</a> (2D Point ) . . . . .	928
<a href="#">NppiRect</a> (2D Rectangle This struct contains position and size information of a rectangle in two space ) . . . . .	929
<a href="#">NppiSize</a> (2D Size This struct typically represents the size of a a rectangular region in two space )	930
<a href="#">NppLibraryVersion</a> . . . . .	931



# Chapter 7

## Module Documentation

### 7.1 NPP Core

Basic functions for library management, in particular library version and device property query functions.

#### Functions

- `const NppLibraryVersion * nppGetLibVersion` (void)  
*Get the NPP library version.*
- `NppGpuComputeCapability nppGetGpuComputeCapability` (void)  
*What CUDA compute model is supported by the default CUDA device?*
- `int nppGetGpuNumSMs` (void)  
*Get the number of Streaming Multiprocessors (SM) on the default CUDA device.*
- `int nppGetMaxThreadsPerBlock` (void)  
*Get the maximum number of threads per block on the default CUDA device.*
- `int nppGetMaxThreadsPerSM` (void)  
*Get the maximum number of threads per SM for the 1st GPU.*
- `const char * nppGetGpuName` (void)  
*Get the name of the default CUDA device.*
- `cudaStream_t nppGetStream` (void)  
*Get the NPP CUDA stream.*
- `void nppSetStream` (cudaStream\_t hStream)  
*Set the NPP CUDA stream.*

#### 7.1.1 Detailed Description

Basic functions for library management, in particular library version and device property query functions.

## 7.1.2 Function Documentation

### 7.1.2.1 `NppGpuComputeCapability nppGetGpuComputeCapability (void)`

What CUDA compute model is supported by the default CUDA device?

Before trying to call any NPP functions, the user should make a call this function to ensure that the current machine has a CUDA capable device.

**Returns:**

An enum value representing if a CUDA capable device was found and what level of compute capabilities it supports.

### 7.1.2.2 `const char* nppGetGpuName (void)`

Get the name of the default CUDA device.

**Returns:**

Name string of the graphics-card/compute device in a system.

### 7.1.2.3 `int nppGetGpuNumSMs (void)`

Get the number of Streaming Multiprocessors (SM) on the default CUDA device.

**Returns:**

Number of SMs of the default CUDA device.

### 7.1.2.4 `const NppLibraryVersion* nppGetLibVersion (void)`

Get the NPP library version.

**Returns:**

A struct containing separate values for major and minor revision and build number.

### 7.1.2.5 `int nppGetMaxThreadsPerBlock (void)`

Get the maximum number of threads per block on the default CUDA device.

**Returns:**

Maximum number of threads per block on the default CUDA device.

**7.1.2.6 int nppGetMaxThreadsPerSM (void)**

Get the maximum number of threads per SM for the 1st GPU.

**Returns:**

Maximum number of threads per SM for the 1st GPU

**7.1.2.7 cudaStream\_t nppGetStream (void)**

Get the NPP CUDA stream.

NPP enables concurrent device tasks via a global stream state variable. The NPP stream by default is set to stream 0, i.e. non-concurrent mode. A user can set the NPP stream to any valid CUDA stream. All CUDA commands issued by NPP (e.g. kernels launched by the NPP library) are then issued to that NPP stream.

**7.1.2.8 void nppSetStream (cudaStream\_t *hStream*)**

Set the NPP CUDA stream.

**See also:**

[nppGetStream\(\)](#)

## 7.2 NPP Type Definitions and Constants

### Data Structures

- struct [NppLibraryVersion](#)
- struct [NppiPoint](#)  
*2D Point*
- struct [NppiSize](#)  
*2D Size This struct typically represents the size of a rectangular region in two space.*
- struct [NppiRect](#)  
*2D Rectangle This struct contains position and size information of a rectangle in two space.*
- struct [NppiHaarClassifier\\_32f](#)
- struct [NppiHaarBuffer](#)

### Modules

- [Basic NPP Data Types](#)

### Defines

- #define [NPP\\_MIN\\_8U](#) ( 0 )  
*Minimum 8-bit unsigned integer.*
- #define [NPP\\_MAX\\_8U](#) ( 255 )  
*Maximum 8-bit unsigned integer.*
- #define [NPP\\_MIN\\_16U](#) ( 0 )  
*Minimum 16-bit unsigned integer.*
- #define [NPP\\_MAX\\_16U](#) ( 65535 )  
*Maximum 16-bit unsigned integer.*
- #define [NPP\\_MIN\\_32U](#) ( 0 )  
*Minimum 32-bit unsigned integer.*
- #define [NPP\\_MAX\\_32U](#) ( 4294967295 )  
*Maximum 32-bit unsigned integer.*
- #define [NPP\\_MIN\\_64U](#) ( 0 )  
*Minimum 64-bit unsigned integer.*
- #define [NPP\\_MAX\\_64U](#) ( 18446744073709551615ULL )  
*Maximum 64-bit unsigned integer.*
- #define [NPP\\_MIN\\_8S](#) (-128 )  
*Minimum 8-bit signed integer.*

- #define `NPP_MAX_8S` ( 127 )  
*Maximum 8-bit signed integer.*
- #define `NPP_MIN_16S` (-32768 )  
*Minimum 16-bit signed integer.*
- #define `NPP_MAX_16S` ( 32767 )  
*Maximum 16-bit signed integer.*
- #define `NPP_MIN_32S` (-2147483647 - 1 )  
*Minimum 32-bit signed integer.*
- #define `NPP_MAX_32S` ( 2147483647 )  
*Maximum 32-bit signed integer.*
- #define `NPP_MAX_64S` ( 9223372036854775807LL )  
*Minimum 64-bit signed integer.*
- #define `NPP_MIN_64S` (-9223372036854775807LL - 1)  
*Maximum 64-bit signed integer.*
- #define `NPP_MINABS_32F` ( 1.175494351e-38f )  
*Smallest positive 32-bit floating point value.*
- #define `NPP_MAXABS_32F` ( 3.402823466e+38f )  
*Largest positive 32-bit floating point value.*
- #define `NPP_MINABS_64F` ( 2.2250738585072014e-308 )  
*Smallest positive 64-bit floating point value.*
- #define `NPP_MAXABS_64F` ( 1.7976931348623158e+308 )  
*Largest positive 64-bit floating point value.*

## Enumerations

- enum `NppiInterpolationMode` {  
`NPPI_INTER_UNDEFINED` = 0,  
`NPPI_INTER_NN` = 1,  
`NPPI_INTER_LINEAR` = 2,  
`NPPI_INTER_CUBIC` = 4,  
`NPPI_INTER_SUPER` = 8,  
`NPPI_INTER_LANCZOS` = 16,  
`NPPI_SMOOTH_EDGE` = (1 << 31) }  
*Filtering methods.*

```

• enum NppStatus {
    NPP_NOT_SUPPORTED_MODE_ERROR = -9999,
    NPP_ROUND_MODE_NOT_SUPPORTED_ERROR = -213,
    NPP_RESIZE_NO_OPERATION_ERROR = -50,
    NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY = -27,
    NPP_BAD_ARG_ERROR = -26,
    NPP_LUT_NUMBER_OF_LEVELS_ERROR = -25,
    NPP_TEXTURE_BIND_ERROR = -24,
    NPP_COEFF_ERROR = -23,
    NPP_RECT_ERROR = -22,
    NPP_QUAD_ERROR = -21,
    NPP_WRONG_INTERSECTION_ROI_ERROR = -20,
    NPP_NOT_EVEN_STEP_ERROR = -19,
    NPP_INTERPOLATION_ERROR = -18,
    NPP_RESIZE_FACTOR_ERROR = -17,
    NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR = -16,
    NPP_MEMFREE_ERR = -15,
    NPP_MEMSET_ERR = -14,
    NPP_MEMCPY_ERROR = -13,
    NPP_MEM_ALLOC_ERR = -12,
    NPP_HISTO_NUMBER_OF_LEVELS_ERROR = -11,
    NPP_MIRROR_FLIP_ERR = -10,
    NPP_INVALID_INPUT = -9,
    NPP_ALIGNMENT_ERROR = -8,
    NPP_STEP_ERROR = -7,
    NPP_SIZE_ERROR = -6,
    NPP_POINTER_ERROR = -5,
    NPP_NULL_POINTER_ERROR = -4,
    NPP_CUDA_KERNEL_EXECUTION_ERROR = -3,
    NPP_NOT_IMPLEMENTED_ERROR = -2,
    NPP_ERROR = -1,
    NPP_NO_ERROR = 0,
    NPP_SUCCESS = NPP_NO_ERROR,
    NPP_WARNING = 1,
    NPP_WRONG_INTERSECTION_QUAD_WARNING = 2,
    NPP_MISALIGNED_DST_ROI_WARNING = 3,
    NPP_AFFINE_QUAD_INCORRECT_WARNING = 4,
    NPP_DOUBLE_SIZE_WARNING = 5,
    NPP_ODD_ROI_WARNING = 6,
    NPP_WRONG_INTERSECTION_ROI_WARNING = 29 }

```

*Error Status Codes.*



- enum `NppGpuComputeCapability` {  
`NPP_CUDA_UNKNOWN_VERSION` = -1,  
`NPP_CUDA_NOT_CAPABLE`,  
`NPP_CUDA_1_0`,  
`NPP_CUDA_1_1`,  
`NPP_CUDA_1_2`,  
`NPP_CUDA_1_3`,  
`NPP_CUDA_2_0` }
- enum `NppiAxis` {  
`NPP_HORIZONTAL_AXIS`,  
`NPP_VERTICAL_AXIS`,  
`NPP_BOTH_AXIS` }
- enum `NppCmpOp` {  
`NPP_CMP_LESS`,  
`NPP_CMP_LESS_EQ`,  
`NPP_CMP_EQ`,  
`NPP_CMP_GREATER_EQ`,  
`NPP_CMP_GREATER` }
- enum `NppRoundMode` {  
`NPP_RND_ZERO`,  
`NPP_RND_NEAR`,  
`NPP_RND_FINANCIAL` }
- enum `NppiBorderType` {  
`NPP_BORDER_UNDEFINED` = -1,  
`NPP_BORDER_NONE` = `NPP_BORDER_UNDEFINED`,  
`NPP_BORDER_CONSTANT` = 0,  
`NPP_BORDER_REPLICATE` = 1,  
`NPP_BORDER_WRAP` = 2 }
- enum `NppHintAlgorithm` {  
`nppAlgHintNone`,  
`nppAlgHintFast`,  
`nppAlgHintAccurate` }

### 7.2.1 Define Documentation

#### 7.2.1.1 #define NPP\_MAX\_16S ( 32767 )

Maximum 16-bit signed integer.

#### 7.2.1.2 #define NPP\_MAX\_16U ( 65535 )

Maximum 16-bit unsigned integer.

**7.2.1.3 #define NPP\_MAX\_32S ( 2147483647 )**

Maximum 32-bit signed integer.

**7.2.1.4 #define NPP\_MAX\_32U ( 4294967295 )**

Maximum 32-bit unsigned integer.

**7.2.1.5 #define NPP\_MAX\_64S ( 9223372036854775807LL )**

Minimum 64-bit signed integer.

**7.2.1.6 #define NPP\_MAX\_64U ( 18446744073709551615ULL )**

Maximum 64-bit unsigned integer.

**7.2.1.7 #define NPP\_MAX\_8S ( 127 )**

Maximum 8-bit signed integer.

**7.2.1.8 #define NPP\_MAX\_8U ( 255 )**

Maximum 8-bit unsigned integer.

**7.2.1.9 #define NPP\_MAXABS\_32F ( 3.402823466e+38f )**

Largest positive 32-bit floating point value.

**7.2.1.10 #define NPP\_MAXABS\_64F ( 1.7976931348623158e+308 )**

Largest positive 64-bit floating point value.

**7.2.1.11 #define NPP\_MIN\_16S (-32768 )**

Minimum 16-bit signed integer.

**7.2.1.12 #define NPP\_MIN\_16U ( 0 )**

Minimum 16-bit unsigned integer.

**7.2.1.13 #define NPP\_MIN\_32S (-2147483647 - 1 )**

Minimum 32-bit signed integer.

**7.2.1.14 #define NPP\_MIN\_32U ( 0 )**

Minimum 32-bit unsigned integer.

**7.2.1.15 #define NPP\_MIN\_64S (-9223372036854775807LL - 1)**

Maximum 64-bit signed integer.

**7.2.1.16 #define NPP\_MIN\_64U ( 0 )**

Minimum 64-bit unsigned integer.

**7.2.1.17 #define NPP\_MIN\_8S (-128 )**

Minimum 8-bit signed integer.

**7.2.1.18 #define NPP\_MIN\_8U ( 0 )**

Minimum 8-bit unsigned integer.

**7.2.1.19 #define NPP\_MINABS\_32F ( 1.175494351e-38f )**

Smallest positive 32-bit floating point value.

**7.2.1.20 #define NPP\_MINABS\_64F ( 2.2250738585072014e-308 )**

Smallest positive 64-bit floating point value.

**7.2.2 Enumeration Type Documentation****7.2.2.1 enum NppCmpOp**

**Enumerator:**

*NPP\_CMP\_LESS*

*NPP\_CMP\_LESS\_EQ*

*NPP\_CMP\_EQ*

*NPP\_CMP\_GREATER\_EQ*

*NPP\_CMP\_GREATER*

**7.2.2.2 enum NppGpuComputeCapability**

**Enumerator:**

*NPP\_CUDA\_UNKNOWN\_VERSION* Indicates that the compute-capability query failed.

*NPP\_CUDA\_NOT\_CAPABLE* Indicates that no CUDA capable device was found on machine.

*NPP\_CUDA\_1\_0* Indicates that CUDA 1.0 capable device is default device on machine.

*NPP\_CUDA\_1\_1* Indicates that CUDA 1.1 capable device.

*NPP\_CUDA\_1\_2* Indicates that CUDA 1.2 capable device.

*NPP\_CUDA\_1\_3* Indicates that CUDA 1.3 capable device.

*NPP\_CUDA\_2\_0* Indicates that CUDA 2.0 or better is default device on machine.

#### 7.2.2.3 enum NppHintAlgorithm

Enumerator:

*nppAlgHintNone*

*nppAlgHintFast*

*nppAlgHintAccurate*

#### 7.2.2.4 enum NppiAxis

Enumerator:

*NPP\_HORIZONTAL\_AXIS*

*NPP\_VERTICAL\_AXIS*

*NPP\_BOTH\_AXIS*

#### 7.2.2.5 enum NppiBorderType

Enumerator:

*NPP\_BORDER\_UNDEFINED*

*NPP\_BORDER\_NONE*

*NPP\_BORDER\_CONSTANT*

*NPP\_BORDER\_REPLICATE*

*NPP\_BORDER\_WRAP*

#### 7.2.2.6 enum NppiInterpolationMode

Filtering methods.

Enumerator:

*NPPI\_INTER\_UNDEFINED*

*NPPI\_INTER\_NN* Nearest neighbor filtering.

*NPPI\_INTER\_LINEAR* Linear interpolation.

*NPPI\_INTER\_CUBIC* Cubic interpolation.

*NPPI\_INTER\_SUPER* Super sampling.

*NPPI\_INTER\_LANCZOS* Lanczos filtering.

*NPPI\_SMOOTH\_EDGE* Smooth edge filtering.

### 7.2.2.7 enum NppRoundMode

Enumerator:

*NPP\_RND\_ZERO*  
*NPP\_RND\_NEAR*  
*NPP\_RND\_FINANCIAL*

### 7.2.2.8 enum NppStatus

Error Status Codes.

Almost all NPP function return error-status information using these return codes. Negative return codes indicate errors, positive return codes indicate warnings, a return code of 0 indicates success.

Enumerator:

*NPP\_NOT\_SUPPORTED\_MODE\_ERROR*  
*NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ERROR*  
*NPP\_RESIZE\_NO\_OPERATION\_ERROR*  
*NPP\_NOT\_SUFFICIENT\_COMPUTE\_CAPABILITY*  
*NPP\_BAD\_ARG\_ERROR*  
*NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR*  
*NPP\_TEXTURE\_BIND\_ERROR*  
*NPP\_COEFF\_ERROR*  
*NPP\_RECT\_ERROR*  
*NPP\_QUAD\_ERROR*  
*NPP\_WRONG\_INTERSECTION\_ROI\_ERROR*  
*NPP\_NOT\_EVEN\_STEP\_ERROR*  
*NPP\_INTERPOLATION\_ERROR*  
*NPP\_RESIZE\_FACTOR\_ERROR*  
*NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR*  
*NPP\_MEMFREE\_ERR*  
*NPP\_MEMSET\_ERR*  
*NPP\_MEMCPY\_ERROR*  
*NPP\_MEM\_ALLOC\_ERR*  
*NPP\_HISTO\_NUMBER\_OF\_LEVELS\_ERROR*  
*NPP\_MIRROR\_FLIP\_ERR*  
*NPP\_INVALID\_INPUT*  
*NPP\_ALIGNMENT\_ERROR*  
*NPP\_STEP\_ERROR* Step is less or equal zero.  
*NPP\_SIZE\_ERROR*  
*NPP\_POINTER\_ERROR*  
*NPP\_NULL\_POINTER\_ERROR*  
*NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR*

***NPP\_NOT\_IMPLEMENTED\_ERROR***

***NPP\_ERROR***

***NPP\_NO\_ERROR*** Error free operation.

***NPP\_SUCCESS*** Successful operation (same as *NPP\_NO\_ERROR*).

***NPP\_WARNING***

***NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING***

***NPP\_MISALIGNED\_DST\_ROI\_WARNING*** Speed reduction due to uncoalesced memory accesses warning.

***NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING*** Indicates that the quadrangle passed to one of affine warping functions doesn't have necessary properties. First 3 vertices are used, the fourth vertex discarded.

***NPP\_DOUBLE\_SIZE\_WARNING*** Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

***NPP\_ODD\_ROI\_WARNING*** Indicates that for 422/411/420 sampling the ROI width/height was forced to even value.

***NPP\_WRONG\_INTERSECTION\_ROI\_WARNING*** ROI doesn't intersect source or destination ROI/image. No operation performed.

## 7.3 Basic NPP Data Types

### Data Structures

- struct [Npp16sc](#)  
*Complex Number This struct represents a short complex number.*
- struct [Npp32sc](#)  
*Complex Number This struct represents a signed int complex number.*
- struct [Npp32fc](#)  
*Complex Number This struct represents a single floating-point complex number.*
- struct [Npp64sc](#)  
*Complex Number This struct represents a long long complex number.*
- struct [Npp64fc](#)  
*Complex Number This struct represents a double floating-point complex number.*

### Typedefs

- typedef unsigned char [Npp8u](#)  
*8-bit unsigned chars*
- typedef signed char [Npp8s](#)  
*8-bit signed chars*
- typedef unsigned short [Npp16u](#)  
*16-bit unsigned integers*
- typedef short [Npp16s](#)  
*16-bit signed integers*
- typedef unsigned int [Npp32u](#)  
*32-bit unsigned integers*
- typedef int [Npp32s](#)  
*32-bit signed integers*
- typedef unsigned long long [Npp64u](#)  
*64-bit unsigned integers*
- typedef long long [Npp64s](#)  
*64-bit signed integers*
- typedef float [Npp32f](#)  
*32-bit (IEEE) floating-point numbers*

- typedef double [Npp64f](#)  
*64-bit floating-point numbers*

### 7.3.1 Typedef Documentation

#### 7.3.1.1 typedef short Npp16s

16-bit signed integers

#### 7.3.1.2 typedef unsigned short Npp16u

16-bit unsigned integers

#### 7.3.1.3 typedef float Npp32f

32-bit (IEEE) floating-point numbers

#### 7.3.1.4 typedef int Npp32s

32-bit signed integers

#### 7.3.1.5 typedef unsigned int Npp32u

32-bit unsigned integers

#### 7.3.1.6 typedef double Npp64f

64-bit floating-point numbers

#### 7.3.1.7 typedef long long Npp64s

64-bit signed integers

#### 7.3.1.8 typedef unsigned long long Npp64u

64-bit unsigned integers

#### 7.3.1.9 typedef signed char Npp8s

8-bit signed chars

#### 7.3.1.10 typedef unsigned char Npp8u

8-bit unsigned chars



## 7.4 NPP Image Processing

### Modules

- [Memory Management](#)

*Routines for allocating and deallocating pitched image storage.*

- [Data-Exchange and Initialization](#)

*Primitives for initialization, copying and converting image data.*

- [Arithmetic and Logical Operations](#)

- [Threshold and Compare Operations](#)

*Methods for pixel-wise threshold and compare operations.*

- [Compression](#)

*Image compression primitives.*

- [Geometric Transforms](#)

*Routines manipulating an image's geometry.*

### Enumerations

- enum [NppiAlphaOp](#) {  
[NPPI\\_OP\\_ALPHA\\_OVER](#),  
[NPPI\\_OP\\_ALPHA\\_IN](#),  
[NPPI\\_OP\\_ALPHA\\_OUT](#),  
[NPPI\\_OP\\_ALPHA\\_ATOP](#),  
[NPPI\\_OP\\_ALPHA\\_XOR](#),  
[NPPI\\_OP\\_ALPHA\\_PLUS](#),  
[NPPI\\_OP\\_ALPHA\\_OVER\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_IN\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_OUT\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_ATOP\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_XOR\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_PLUS\\_PREMUL](#),  
[NPPI\\_OP\\_ALPHA\\_PREMUL](#) }

### Functions

- [NppStatus nppiSqrIntegral\\_8u32s32f\\_C1R](#) ([Npp8u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [Npp32f](#) \*pSqr, int nSqrStep, [NppiSize](#) oSrcROI, [Npp32s](#) val, [Npp32f](#) valSqr, [Npp32s](#) integralImageNewHeight)

*SqrIntegral Transforms an image to integral and integral of pixel squares representation.*

- **NppStatus** **nppiRectStdDev\_32s32f\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, const **Npp64f** \*pSqr, int nSqrStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** rect)

*RectStdDev Computes the standard deviation of integral images.*

## Image Labeling Techniques

- typedef struct **NppiGraphcutState** **NppiGraphcutState**
- **NppStatus** **nppiGraphcutGetSize** (**NppiSize** oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.*
- **NppStatus** **nppiGraphcut8GetSize** (**NppiSize** oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.*
- **NppStatus** **nppiGraphcutInitAlloc** (**NppiSize** oSize, **NppiGraphcutState** \*\*ppState, **Npp8u** \*pDeviceMem)  
*Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.*
- **NppStatus** **nppiGraphcut8InitAlloc** (**NppiSize** oSize, **NppiGraphcutState** \*\*ppState, **Npp8u** \*pDeviceMem)  
*Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.*
- **NppStatus** **nppiGraphcutFree** (**NppiGraphcutState** \*pState)  
*Frees the additional resources of the graph-cut state structure.*
- **NppStatus** **nppiGraphcut\_32s8u** (**Npp32s** \*pTerminals, **Npp32s** \*pLeftTransposed, **Npp32s** \*pRightTransposed, **Npp32s** \*pTop, **Npp32s** \*pBottom, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** \*pLabel, int nLabelStep, **NppiGraphcutState** \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*
- **NppStatus** **nppiGraphcut8\_32s8u** (**Npp32s** \*pTerminals, **Npp32s** \*pLeftTransposed, **Npp32s** \*pRightTransposed, **Npp32s** \*pTop, **Npp32s** \*pTopLeft, **Npp32s** \*pTopRight, **Npp32s** \*pBottom, **Npp32s** \*pBottomLeft, **Npp32s** \*pBottomRight, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** \*pLabel, int nLabelStep, **NppiGraphcutState** \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*

## Mean\_StdDev

Computes the mean and standard deviation of image pixel values

- **NppStatus** **nppiMeanStdDev8uC1RGetBufferHostSize** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for mean and standard deviation of image.*
- **NppStatus** **nppiMean\_StdDev\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pMean, **Npp64f** \*pStdDev)  
*8-bit unsigned mean standard deviation.*

## NormDiff

Norm of pixel differences between two images.

- `NppStatus nppiNormDiff_L1_8u_C1R` (const `Npp8u` \*pSrc1, int nSrcStep1, const `Npp8u` \*pSrc2, int nSrcStep2, `NppiSize` oSizeROI, `Npp64f` \*pRetVal)  
*8-bit unsigned L1 norm of pixel differences.*
- `NppStatus nppiNormDiff_L2_8u_C1R` (const `Npp8u` \*pSrc1, int nSrcStep1, const `Npp8u` \*pSrc2, int nSrcStep2, `NppiSize` oSizeROI, `Npp64f` \*pRetVal)  
*8-bit unsigned L2 norm of pixel differences.*
- `NppStatus nppiNormDiff_Inf_8u_C1R` (const `Npp8u` \*pSrc1, int nSrcStep1, const `Npp8u` \*pSrc2, int nSrcStep2, `NppiSize` oSizeROI, `Npp64f` \*pRetVal)  
*8-bit unsigned Infinity Norm of pixel differences.*

## 1D Linear Filter

1D mask Linear Convolution Filter, with rescaling, for 8 bit images.

- `NppStatus nppiFilterColumn_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)  
*8-bit unsigned 1D (column) image convolution.*
- `NppStatus nppiFilterColumn_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)  
*4 channel 8-bit unsigned 1D (column) image convolution.*
- `NppStatus nppiFilterRow_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)  
*8-bit unsigned 1D (row) image convolution.*
- `NppStatus nppiFilterRow_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)  
*4 channel 8-bit unsigned 1D (row) image convolution.*

## 1D Window Sum

1D mask Window Sum for 8 bit images.

- `NppStatus nppiSumWindowColumn_8u32f_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*8-bit unsigned 1D (column) sum to 32f.*

- `NppStatus nppiSumWindowRow_8u32f_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*8-bit unsigned 1D (row) sum to 32f.*

## 2D Morphology Filter

Image dilate and erod operations.

- `NppStatus nppiDilate_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*8-bit unsigned image dilation.*

- `NppStatus nppiDilate_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*4 channel 8-bit unsigned image dilation.*

- `NppStatus nppiErode_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*8-bit unsigned image erosion.*

- `NppStatus nppiErode_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*4 channel 8-bit unsigned image erosion.*

## Convolution (2D Masks)

General purpose 2D convolution filters.

- `NppStatus nppiFilter_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*8-bit unsigned convolution filter.*

- `NppStatus nppiFilter_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*4 channel 8-bit unsigned convolution filter.*

## 2D Linear Fixed Filters

2D linear fixed filters for 8 bit images.

- `NppStatus nppiFilterBox_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*8-bit unsigned box filter.*

- `NppStatus nppiFilterBox_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*4 channel 8-bit unsigned box filter.*

## Image Rank Filters

Min, Median, and Max image filters.

- `NppStatus nppiFilterMax_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*8-bit unsigned maximum filter.*

- `NppStatus nppiFilterMax_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*4 channel 8-bit unsigned maximum filter.*

- `NppStatus nppiFilterMin_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*8-bit unsigned minimum filter.*

- `NppStatus nppiFilterMin_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*4 channel 8-bit unsigned minimum filter.*

## Image Linear Transforms

Linear image transforms, like Fourier and DCT transformations.

- `NppStatus nppiMagnitude_32fc32f_C1R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit floating point complex to 32-bit floating point magnitude.*

- `NppStatus nppiMagnitudeSqr_32fc32f_C1R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit floating point complex to 32-bit floating point squared magnitude.*

## Histogram

- `NppStatus nppiEvenLevelsHost_32s` (`Npp32s` \*hpLevels, int nLevels, `Npp32s` nLowerLevel, `Npp32s` nUpperLevel)

*Compute levels with even distribution.*

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, int nLevels, int \*hpBufferSize)

*Scratch-buffer size for `nppiHistogramEven_8u_C1R`.*

- **NppStatus** **nppiHistogramEven\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, int nLevels, **Npp32s** nLowerLevel, **Npp32s** nUpperLevel, **Npp8u** \*pBuffer)  
*8-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_8u\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_C4R.*
- **NppStatus** **nppiHistogramEven\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], int nLevels[4], **Npp32s** nLowerLevel[4], **Npp32s** nUpperLevel[4], **Npp8u** \*pBuffer)  
*4 channel 8-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_8u\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_AC4R.*
- **NppStatus** **nppiHistogramEven\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], int nLevels[3], **Npp32s** nLowerLevel[3], **Npp32s** nUpperLevel[3], **Npp8u** \*pBuffer)  
*4 channel (alpha as the last channel) 8-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_16u\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_C1R.*
- **NppStatus** **nppiHistogramEven\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, int nLevels, **Npp32s** nLowerLevel, **Npp32s** nUpperLevel, **Npp8u** \*pBuffer)  
*16-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_16u\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_C4R.*
- **NppStatus** **nppiHistogramEven\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], int nLevels[4], **Npp32s** nLowerLevel[4], **Npp32s** nUpperLevel[4], **Npp8u** \*pBuffer)  
*4 channel 16-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_16u\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_AC4R.*
- **NppStatus** **nppiHistogramEven\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], int nLevels[3], **Npp32s** nLowerLevel[3], **Npp32s** nUpperLevel[3], **Npp8u** \*pBuffer)  
*4 channel (alpha as the last channel) 16-bit unsigned histogram with evenly distributed bins.*
- **NppStatus** **nppiHistogramEvenGetBufferSize\_16s\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_C1R.*

- `NppStatus nppiHistogramEven_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist, int nLevels, `Npp32s` nLowerLevel, `Npp32s` nUpperLevel, `Npp8u` \*pBuffer)  
*16-bit signed histogram with evenly distributed bins.*
- `NppStatus nppiHistogramEvenGetBufferSize_16s_C4R` (`NppiSize` oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_C4R.*
- `NppStatus nppiHistogramEven_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[4], int nLevels[4], `Npp32s` nLowerLevel[4], `Npp32s` nUpperLevel[4], `Npp8u` \*pBuffer)  
*4 channel 16-bit signed histogram with evenly distributed bins.*
- `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R` (`NppiSize` oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_AC4R.*
- `NppStatus nppiHistogramEven_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` \*pBuffer)  
*4 channel (alpha as the last channel) 16-bit signed histogram with evenly distributed bins.*
- `NppStatus nppiHistogramRangeGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C1R.*
- `NppStatus nppiHistogramRange_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist, const `Npp32s` \*pLevels, int nLevels, `Npp8u` \*pBuffer)  
*8-bit unsigned histogram with bins determined by pLevels array.*
- `NppStatus nppiHistogramRangeGetBufferSize_8u_C4R` (`NppiSize` oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C4R.*
- `NppStatus nppiHistogramRange_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[4], const `Npp32s` \*pLevels[4], int nLevels[4], `Npp8u` \*pBuffer)  
*4 channel 8-bit unsigned histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_8u_AC4R` (`NppiSize` oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.*
- `NppStatus nppiHistogramRange_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], const `Npp32s` \*pLevels[3], int nLevels[3], `Npp8u` \*pBuffer)  
*4 channel (alpha as a last channel) 8-bit unsigned histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_16u_C1R` (`NppiSize` oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C1R.*

- **NppStatus nppiHistogramRange\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*16-bit unsigned histogram with bins determined by pLevels array.*
- **NppStatus nppiHistogramRangeGetBufferSize\_16u\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C4R.*
- **NppStatus nppiHistogramRange\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*4 channel 16-bit unsigned histogram with bins determined by pLevels.*
- **NppStatus nppiHistogramRangeGetBufferSize\_16u\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.*
- **NppStatus nppiHistogramRange\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*4 channel (alpha as a last channel) 16-bit unsigned histogram with bins determined by pLevels.*
- **NppStatus nppiHistogramRangeGetBufferSize\_16s\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C1R.*
- **NppStatus nppiHistogramRange\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*16-bit signed histogram with bins determined by pLevels array.*
- **NppStatus nppiHistogramRangeGetBufferSize\_16s\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C4R.*
- **NppStatus nppiHistogramRange\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*4 channel 16-bit signed histogram with bins determined by pLevels.*
- **NppStatus nppiHistogramRangeGetBufferSize\_16s\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_AC4R.*
- **NppStatus nppiHistogramRange\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*4 channel (alpha as a last channel) 16-bit signed histogram with bins determined by pLevels.*
- **NppStatus nppiHistogramRangeGetBufferSize\_32f\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C1R.*
- **NppStatus nppiHistogramRange\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32f** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*32-bit float histogram with bins determined by pLevels array.*



- `NppStatus nppiHistogramRangeGetBufferSize_32f_C4R` (`NppiSize` oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C4R.*
- `NppStatus nppiHistogramRange_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[4], const `Npp32f` \*pLevels[4], int nLevels[4], `Npp8u` \*pBuffer)  
*4 channel 32-bit float histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_32f_AC4R` (`NppiSize` oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.*
- `NppStatus nppiHistogramRange_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], const `Npp32f` \*pLevels[3], int nLevels[3], `Npp8u` \*pBuffer)  
*4 channel (alpha as a last channel) 32-bit float histogram with bins determined by pLevels.*

## Sum

Sum of 8 bit images.

- `NppStatus nppiReductionGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Scratch-buffer size for nppiSum\_8u\_C1R.*
- `NppStatus nppiReductionGetBufferHostSize_8u_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Scratch-buffer size for nppiSum\_8u\_C4R.*
- `NppStatus nppiSum_8u64s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64s` \*pSum)  
*8-bit unsigned image sum with 64-bit long long result.*
- `NppStatus nppiSum_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pSum)  
*8-bit unsigned image sum with 64-bit double precision result.*
- `NppStatus nppiSum_8u64s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64s` aSum[4])  
*4 channel 8-bit unsigned image sum with 64-bit long long result.*
- `NppStatus nppiSum_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aSum[4])  
*4 channel 8-bit unsigned image sum with 64-bit double precision result.*

## MinMax

Minimum and maximum of 8-bit images.

- `NppStatus nppiMinMaxGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Scratch-buffer size for nppiMinMax\_8u\_C1R.*

- `NppStatus nppiMinMax_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pMin, `Npp8u` \*pMax, `Npp8u` \*pDeviceBuffer)

*8-bit unsigned pixel minimum and maximum.*

- `NppStatus nppiMinMaxGetBufferSize_8u_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Scratch-buffer size for nppiMinMax\_8u\_C4R.*

- `NppStatus nppiMinMax_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[4], `Npp8u` aMax[4], `Npp8u` \*pDeviceBuffer)

*4 channel 8-bit unsigned pixel minimum and maximum.*

## RGBToYCbCr

RGB to YCbCr color conversion.

- `NppStatus nppiRGBToYCbCr_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed RGB to packed YCbCr color conversion.*

- `NppStatus nppiRGBToYCbCr422_8u_C3C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned RGB to 2 channel chroma packed YCbCr422 color conversion.*

- `NppStatus nppiRGBToYCbCr420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*\*pDst, int nDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed RGB to planar YCbCr420 color conversion.*

- `NppStatus nppiRGBToYCbCr_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep, `Npp8u` \*\*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel planar 8-bit unsigned RGB to YCbCr color conversion.*

- `NppStatus nppiRGBToYCbCr_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 8-bit unsigned RGB to YCbCr color conversion, ignoring Alpha.*

## YCbCrToRGB

YCbCr to RGB color conversion.

- `NppStatus nppiYCbCrToRGB_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed YCbCr to RGB color conversion.*

- `NppStatus nppiYCbCrToRGB_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep, `Npp8u` \*\*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YCbCr to RGB color conversion.*

- `NppStatus nppiYCbCrToRGB_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed YCbCr to RGB color conversion, not affecting Alpha.*
- `NppStatus nppiYCbCr422ToRGB_8u_C2C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned YCbCr422 to 3 channel packed RGB color conversion.*
- `NppStatus nppiYCbCr420ToRGB_8u_P3C3R` (const `Npp8u` \*const \*pSrc, int nSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to packed RGB color conversion.*

### Sample Pattern Conversion.

- `NppStatus nppiYCbCr422ToYCbCr420_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep[3], `Npp8u` \*\*pDst, int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr:422 to YCbCr:420 resampling.*
- `NppStatus nppiYCbCr422ToYCbCr411_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep[3], `Npp8u` \*\*pDst, int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr:422 to YCbCr:411 resampling.*
- `NppStatus nppiYCbCr420ToYCbCr422_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep[3], `Npp8u` \*\*pDst, int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr:420 to YCbCr:422 resampling.*
- `NppStatus nppiYCbCr420ToYCbCr411_8u_P3P2R` (const `Npp8u` \*const \*pSrc, int aSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr:420 to YCbCr:411 resampling.*

### Color Processing

Color manipulation functions.

- `NppStatus nppiColorTwist32f_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*3 channel 8-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep, `Npp8u` \*\*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*3 channel planar 8-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*4 channel 8-bit unsigned color twist, not affecting Alpha.*

- `NppStatus npplUT_Linear_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*8-bit unsigned look-up-table color conversion.*
- `NppStatus npplUT_Linear_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 8-bit unsigned look-up-table color conversion.*
- `NppStatus npplUT_Linear_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.*

## 7.4.1 Typedef Documentation

### 7.4.1.1 typedef struct NppiGraphcutState NppiGraphcutState

## 7.4.2 Enumeration Type Documentation

### 7.4.2.1 enum NppiAlphaOp

Enumerator:

`NPPI_OP_ALPHA_OVER`  
`NPPI_OP_ALPHA_IN`  
`NPPI_OP_ALPHA_OUT`  
`NPPI_OP_ALPHA_ATOP`  
`NPPI_OP_ALPHA_XOR`  
`NPPI_OP_ALPHA_PLUS`  
`NPPI_OP_ALPHA_OVER_PREMUL`  
`NPPI_OP_ALPHA_IN_PREMUL`  
`NPPI_OP_ALPHA_OUT_PREMUL`  
`NPPI_OP_ALPHA_ATOP_PREMUL`  
`NPPI_OP_ALPHA_XOR_PREMUL`  
`NPPI_OP_ALPHA_PLUS_PREMUL`  
`NPPI_OP_ALPHA_PREMUL`

## 7.4.3 Function Documentation

### 7.4.3.1 `NppStatus npplColorTwist32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])

4 channel 8-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point pixel values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*twist* The color twist matrix with floating-point pixel values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.2 NppStatus npplColorTwist32f\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI, const Npp32f twist[3][4])

3 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point pixel values is applied within ROI.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*twist* The color twist matrix with floating-point pixel values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.3 NppStatus npplColorTwist32f\_8u\_P3R (const Npp8u \*const \*pSrc, int nSrcStep, Npp8u \*\*pDst, int nDstStep, NppiSize oSizeROI, const Npp32f twist[3][4])

3 channel planar 8-bit unsigned color twist.

An input color twist matrix with floating-point pixel values is applied within ROI.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*twist* The color twist matrix with floating-point pixel values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.4 NppStatus nppiDilate\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

8-bit unsigned image dilation.

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.5 NppStatus nppiDilate\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

4 channel 8-bit unsigned image dilation.

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.6 NppStatus nppeErode\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

8-bit unsigned image erosion.

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.7 NppStatus nppeErode\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

4 channel 8-bit unsigned image erosion.

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.8 NppStatus nppiEvenLevelsHost\_32s (Npp32s \* *hpLevels*, int *nLevels*, Npp32s *nLowerLevel*, Npp32s *nUpperLevel*)

Compute levels with even distribution.

##### Parameters:

*hpLevels* A host pointer to array which receives the levels being computed. The array needs to be of size *nLevels*.

*nLevels* The number of levels being computed. *nLevels* must be at least 2, otherwise an NPP\_HISTO\_NUMBER\_OF\_LEVELS\_ERROR error is returned.

*nLowerLevel* Lower boundary value of the lowest level.

*nUpperLevel* Upper boundary value of the greatest level.

##### Returns:

Error code.

#### 7.4.3.9 NppStatus nppiFilter\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*, Npp32s *nDivisor*)

8-bit unsigned convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by *nDivisor*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.10 NppStatus nppiFilter\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*, Npp32s *nDivisor*)

4 channel 8-bit unsigned convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by *nDivisor*.



**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.
- nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.11 NppStatus nppiFilterBox\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

8-bit unsigned box filter.

Computes the average pixel values of the pixels under a rectangular mask.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- oMaskSize* Width and Height of the neighborhood region for the local Avg operation.
- oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.12 NppStatus nppiFilterBox\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

4 channel 8-bit unsigned box filter.

Computes the average pixel values of the pixels under a rectangular mask.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.13** `NppStatus nppiFilterColumn_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned 1D (column) image convolution.

Apply convolution filter with user specified 1D column of weights. Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.14** `NppStatus nppiFilterColumn_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

4 channel 8-bit unsigned 1D (column) image convolution.

Apply convolution filter with user specified 1D column of weights. Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.15 **NppStatus nppiFilterMax\_8u\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

8-bit unsigned maximum filter.

Result pixel value is the maximum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.16 **NppStatus nppiFilterMax\_8u\_C4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

4 channel 8-bit unsigned maximum filter.

Result pixel value is the maximum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.17 NppStatus nppiFilterMin\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)**

8-bit unsigned minimum filter.

Result pixel value is the minimum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.18 NppStatus nppiFilterMin\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)**

4 channel 8-bit unsigned minimum filter.

Result pixel value is the minimum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.4.3.19 NppStatus nppiFilterRow\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32s \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*)

8-bit unsigned 1D (row) image convolution.

Apply general linear Row convolution filter, with rescaling, in a 1D mask region around each source pixel for 1-channel 8 bit/pixel images. Result pixel is equal to the sum of the products between the kernel coefficients (*pKernel* array) and corresponding neighboring row pixel values in the source image defined by *iKernelDim* and *iAnchorX*, divided by *iDivisor*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.20 NppStatus nppiFilterRow\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32s \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*)

4 channel 8-bit unsigned 1D (row) image convolution.

Apply general linear Row convolution filter, with rescaling, in a 1D mask region around each source pixel for 1-channel 8 bit/pixel images. Result pixel is equal to the sum of the products between the kernel coefficients (*pKernel* array) and corresponding neighboring row pixel values in the source image defined by *iKernelDim* and *iAnchorX*, divided by *iDivisor*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

***nAnchor*** X offset of the kernel origin frame of reference w.r.t the source pixel.

***nDivisor*** The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.21 NppStatus nppiGraphcut8\_32s8u (Npp32s \* *pTerminals*, Npp32s \* *pLeftTransposed*, Npp32s \* *pRightTransposed*, Npp32s \* *pTop*, Npp32s \* *pTopLeft*, Npp32s \* *pTopRight*, Npp32s \* *pBottom*, Npp32s \* *pBottomLeft*, Npp32s \* *pBottomRight*, int *nStep*, int *nTransposedStep*, NppiSize *size*, Npp8u \* *pLabel*, int *nLabelStep*, NppiGraphcutState \* *pState*)**

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( *terminals(x)* = *source(x)* - *sink(x)* ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example *left(0,\*)* == 0). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 / 255).

#### See also:

[nppiGraphcut8GetSize](#)  
[nppiGraphcut8InitAlloc](#)

#### Parameters:

***pTerminals*** Pointer to differences of terminal edge capacities (*terminal(x)* = *source(x)* - *sink(x)*)

***pLeftTransposed*** Pointer to transposed left edge capacities (*left(0,\*)* must be 0)

***pRightTransposed*** Pointer to transposed right edge capacities (*right(width-1,\*)* must be 0)

***pTop*** Pointer to top edge capacities (*top(\*,0)* must be 0)

***pTopLeft*** Pointer to top left edge capacities (*opleft(\*,0)* & *opleft(0,\*)* must be 0)

***pTopRight*** Pointer to top right edge capacities (*topright(\*,0)* & *topright(width-1,\*)* must be 0)

***pBottom*** Pointer to bottom edge capacities (*bottom(\*,height-1)* must be 0)

***pBottomLeft*** Pointer to bottom left edge capacities (*bottomleft(\*,height-1)* && *bottomleft(0,\*)* must be 0)

***pBottomRight*** Pointer to bottom right edge capacities (*bottomright(\*,height-1)* && *bottomright(width-1,\*)* must be 0)

***nStep*** Step in bytes between any pair of sequential rows of edge capacities

***nTransposedStep*** Step in bytes between any pair of sequential rows of transposed edge capacities

***size*** Graph size

***pLabel*** Pointer to destination label image

***nLabelStep*** Step in bytes between any pair of sequential rows of label image

***pState*** Pointer to graph-cut state structure

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.22 NppStatus nppiGraphcut8GetSize (NppiSize oSize, int \* pBufSize)**

Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8\\_32s8u](#)

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.4.3.23 NppStatus nppiGraphcut8InitAlloc (NppiSize oSize, NppiGraphcutState \*\* ppState, Npp8u \* pDeviceMem)**

Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8\\_32s8u](#)

[nppiGraphcut8GetSize](#)

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* to the device memory used

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.4.3.24 NppStatus nppiGraphcut\_32s8u (Npp32s \* pTerminals, Npp32s \* pLeftTransposed, Npp32s \* pRightTransposed, Npp32s \* pTop, Npp32s \* pBottom, int nStep, int nTransposedStep, NppiSize size, Npp8u \* pLabel, int nLabelStep, NppiGraphcutState \* pState)**

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( terminals(x) = source(x) - sink(x) ). The implementation assumes that the edge capacities for boundary edges that would

connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 / 255).

**See also:**

[nppiGraphcutGetSize](#)  
[nppiGraphcutInitAlloc](#)

**Parameters:**

*pTerminals* Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)  
*pLeftTransposed* Pointer to transposed left edge capacities (`left(0,*)` must be 0)  
*pRightTransposed* Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)  
*pTop* Pointer to top edge capacities (`top(*,0)` must be 0)  
*pBottom* Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)  
*nStep* Step in bytes between any pair of sequential rows of edge capacities  
*nTransposedStep* Step in bytes between any pair of sequential rows of transposed edge capacities  
*size* Graph size  
*pLabel* Pointer to destination label image  
*nLabelStep* Step in bytes between any pair of sequential rows of label image  
*pState* Pointer to graph-cut state structure

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.25 NppStatus nppiGraphcutFree (NppiGraphcutState \* pState)

Frees the additional resources of the graph-cut state structure.

**See also:**

[nppiGraphcutInitAlloc](#)  
[nppiGraphcut8InitAlloc](#)

**Parameters:**

*pState* Pointer to graph-cut state structure.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pState pointer is NULL



**7.4.3.26 NppStatus nppiGraphcutGetSize (NppiSize *oSize*, int \* *pBufSize*)**

Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.

See also:

[nppiGraphcut\\_32s8u](#)

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.4.3.27 NppStatus nppiGraphcutInitAlloc (NppiSize *oSize*, NppiGraphcutState \*\* *ppState*, Npp8u \* *pDeviceMem*)**

Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut\\_32s8u](#)

[nppiGraphcutGetSize](#)

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* to the device memory used

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.4.3.28 NppStatus nppiHistogramEven\_16s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u \* *pBuffer*)**

4 channel (alpha as the last channel) 16-bit signed histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16s\_AC4R) scratch buffer.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.29** `NppStatus nppiHistogramEven_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

16-bit signed histogram with evenly distributed bins.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16s\_C1R) scratch buffer.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.30** `NppStatus nppiHistogramEven_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

4 channel 16-bit signed histogram with evenly distributed bins.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.  
*nLowerLevel* Array containing lower-level of lowest bin per color channel.  
*nUpperLevel* Array containing upper-level of highest bin per color channel.  
*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16s\_C4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.31** `NppStatus nppiHistogramEven_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

4 channel (alpha as the last channel) 16-bit unsigned histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.  
*nLevels* Array containing number of levels per color channel.  
*nLowerLevel* Array containing lower-level of lowest bin per color channel.  
*nUpperLevel* Array containing upper-level of highest bin per color channel.  
*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16u\_AC4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.32** `NppStatus nppiHistogramEven_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

16-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.  
*nLevels* Number of levels.  
*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16u\_C1R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.33** `NppStatus nppiHistogramEven_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

4 channel 16-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16u\_C4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.34** `NppStatus nppiHistogramEven_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

4 channel (alpha as the last channel) 8-bit unsigned histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_8u\_AC4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.35** `NppStatus nppiHistogramEven_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

8-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.36** `NppStatus nppiHistogramEven_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

4 channel 8-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by `pHist[i]` be of size `nLevels[i]-1`.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.37** `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)`

Scratch-buffer size for `nppiHistogramEven_16s_AC4R`.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.38 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramEven\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.39 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramEven\_16s\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.40 NppStatus nppiHistogramEvenGetBufferSize\_16u\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramEven\_16u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.41 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramEven\_16u\_C1R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.42 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramEven\_16u\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.43 NppStatus nppiHistogramEvenGetBufferSize\_8u\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramEven\_8u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.44 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramEven\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nLevels* Number of levels in the histogram.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.45 **NppStatus nppiHistogramEvenGetBufferSize\_8u\_C4R** (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramEven\_8u\_C4R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.46 **NppStatus nppiHistogramRange\_16s\_AC4R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

4 channel (alpha as a last channel) 16-bit signed histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.  
*nLevels* Array containing number of levels per color channel.  
*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].  
*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



#### 7.4.3.47 **NppStatus nppiHistogramRange\_16s\_C1R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*, const Npp32s \* *pLevels*, int *nLevels*, Npp8u \* *pBuffer*)

16-bit signed histogram with bins determined by *pLevels* array.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16_C1R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.48 **NppStatus nppiHistogramRange\_16s\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

4 channel 16-bit signed histogram with bins determined by *pLevels*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16s_C4R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.49 **NppStatus nppiHistogramRange\_16u\_AC4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

4 channel (alpha as a last channel) 16-bit unsigned histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.
- nLevels* Array containing number of levels per color channel.
- pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].
- pBuffer* Pointer to appropriately sized (nppiHistogramRangeGetBufferSize\_16u\_AC4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.50 **NppStatus nppiHistogramRange\_16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*, const Npp32s \* *pLevels*, int *nLevels*, Npp8u \* *pBuffer*)

16-bit unsigned histogram with bins determined by pLevels array.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.
- pLevels* Pointer to array containing the level sizes of the bins. The array must be of size nLevels.
- nLevels* Number of levels in histogram.
- pBuffer* Pointer to appropriately sized (nppiHistogramRangeGetBufferSize\_16u\_C1R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.51 **NppStatus nppiHistogramRange\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

4 channel 16-bit unsigned histogram with bins determined by pLevels.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`npplHistogramRangeGetBufferSize_16u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.52 NppStatus npplHistogramRange\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist[3], const Npp32f \* pLevels[3], int nLevels[3], Npp8u \* pBuffer)**

4 channel (alpha as a last channel) 32-bit float histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`npplHistogramRangeGetBufferSize_32f_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.53 NppStatus npplHistogramRange\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist, const Npp32f \* pLevels, int nLevels, Npp8u \* pBuffer)**

32-bit float histogram with bins determined by *pLevels* array.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels-1*.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`npplHistogramRangeGetBufferSize_32f_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.54 **NppStatus nppiHistogramRange\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

4 channel 32-bit float histogram with bins determined by *pLevels*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_C4R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.55 **NppStatus nppiHistogramRange\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

4 channel (alpha as a last channel) 8-bit unsigned histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_AC4R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.56 **NppStatus nppiHistogramRange\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*, const Npp32s \* *pLevels*, int *nLevels*, Npp8u \* *pBuffer*)

8-bit unsigned histogram with bins determined by *pLevels* array.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_C1R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.57 **NppStatus nppiHistogramRange\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

4 channel 8-bit unsigned histogram with bins determined by *pLevels*.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_C4R`) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.58 **NppStatus nppiHistogramRangeGetBufferSize\_16s\_AC4R** (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)

Scratch-buffer size for `nppiHistogramRange_16s_AC4R`.

##### Parameters:

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.59 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.60 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16s\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.61 NppStatus nppiHistogramRangeGetBufferSize\_16u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.62 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_C1R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.63 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.64 NppStatus nppiHistogramRangeGetBufferSize\_32f\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.65 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nLevels* Number of levels in the histogram.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.66 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_C4R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.67 NppStatus nppiHistogramRangeGetBufferSize\_8u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.68 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nLevels* Number of levels in the histogram.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.



#### 7.4.3.69 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)

Scratch-buffer size for nppiHistogramRange\_8u\_C4R.

##### Parameters:

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

##### Returns:

Error Code.

#### 7.4.3.70 NppStatus nppiLUT\_Linear\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues[4], const Npp32s \* pLevels[4], int nLevels[4])

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Double pointer to an [4] of arrays of user defined OUTPUT values per CHANNEL

*pLevels* Double pointer to an [4] of arrays of user defined INPUT values per CHANNEL

*nLevels* A [4] array of user defined input/output mapping points (levels) per CHANNEL

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR** if the number of levels is less than 2.

#### 7.4.3.71 NppStatus nppiLUT\_Linear\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues, const Npp32s \* pLevels, int nLevels)

8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pValues* Pointer to an array of user defined OUTPUT values  
*pLevels* Pointer to an array of user defined INPUT values  
*nLevels* Number of user defined input/output mapping points (levels)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2.

**7.4.3.72** `NppStatus nppiLUT_Linear_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pValues* Double pointer to an [3] of arrays of user defined OUTPUT values per CHANNEL  
*pLevels* Double pointer to an [3] of arrays of user defined INPUT values per CHANNEL  
*nLevels* A [3] array of user defined input/output mapping points (levels) per CHANNEL

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2.

**7.4.3.73** `NppStatus nppiMagnitude_32fc32f_C1R (const Npp32fc * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit floating point complex to 32-bit floating point magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the magnitude of the complex values.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.74 NppStatus nppiMagnitudeSqr\_32fc32f\_C1R (const Npp32fc \*pSrc, int nSrcStep, Npp32f \*pDst, int nDstStep, NppiSize oSizeROI)**

32-bit floating point complex to 32-bit floating point squared magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the squared magnitude of the complex values.

The squared magnitude is an intermediate result of magnitude computation and can thus be computed faster than actual magnitude. If magnitudes are required for sorting/comparing only, using this function instead of nppiMagnitude\_32fc32f\_C1R can be a worthwhile performance optimization.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.75 NppStatus nppiMean\_StdDev\_8u\_C1R (const Npp8u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \*pDeviceBuffer, Npp64f \*pMean, Npp64f \*pStdDev)**

8-bit unsigned mean standard deviation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.4.3.76 **NppStatus nppiMeanStdDev8uC1RGetBufferHostSize** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for mean and standard deviation of image.

This primitive provides the correct buffer size for nppiMean\_StdDev\_8u\_C1R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_SUCCESS

#### 7.4.3.77 **NppStatus nppiMinMax\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pMin*, Npp8u \* *pMax*, Npp8u \* *pDeviceBuffer*)

8-bit unsigned pixel minimum and maximum.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMin* Device-memory pointer receiving the minimum result.

*pMax* Device-memory pointer receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.78 **NppStatus nppiMinMax\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u *aMin*[4], Npp8u *aMax*[4], Npp8u \* *pDeviceBuffer*)

4 channel 8-bit unsigned pixel minimum and maximum.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Device-pointer (array) receiving the minimum result.

*aMax* Device-pointer (array) receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferSize\\_8u\\_C4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**Note:**

Unlike `nppiMinMax_8u_C1R`, this primitive returns its results as device pointers.

**7.4.3.79 NppStatus nppiMinMaxGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Scratch-buffer size for `nppiMinMax_8u_C1R`.

**Parameters:**

*oSizeROI* ROI size.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.80 NppStatus nppiMinMaxGetBufferSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Scratch-buffer size for `nppiMinMax_8u_C4R`.

**Parameters:**

*oSizeROI* ROI size.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.4.3.81 NppStatus nppiNormDiff\_Inf\_8u\_C1R (const Npp8u \* pSrc1, int nSrcStep1, const Npp8u \* pSrc2, int nSrcStep2, NppiSize oSizeROI, Npp64f \* pRetVal)**

8-bit unsigned Infinity Norm of pixel differences.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrcStep1* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrcStep2* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*\*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.82** `NppStatus nppiNormDiff_L1_8u_C1R (const Npp8u * pSrc1, int nSrcStep1, const Npp8u * pSrc2, int nSrcStep2, NppiSize oSizeROI, Npp64f * pRetVal)`

8-bit unsigned L1 norm of pixel differences.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrcStep1* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrcStep2* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.83** `NppStatus nppiNormDiff_L2_8u_C1R (const Npp8u * pSrc1, int nSrcStep1, const Npp8u * pSrc2, int nSrcStep2, NppiSize oSizeROI, Npp64f * pRetVal)`

8-bit unsigned L2 norm of pixel differences.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrcStep1* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrcStep2* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.84** `NppStatus nppiRectStdDev_32s32f_C1R (const Npp32s * pSrc, int nSrcStep, const Npp64f * pSqr, int nSqrStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiRect rect)`

RectStdDev Computes the standard deviation of integral images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSqr* Destination-Image Pointer.

*nSqrStep* Destination-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rect* rectangular window

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.4.3.85 NppStatus nppiReductionGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Scratch-buffer size for nppiSum\_8u\_C1R.

**Parameters:**

*oSizeROI* ROI size.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.86 NppStatus nppiReductionGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Scratch-buffer size for nppiSum\_8u\_C4R.

**Parameters:**

*oSizeROI* ROI size.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.4.3.87 NppStatus nppiRGBToYCbCr420\_8u\_C3P3R (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \*\**pDst*, int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to planar YCbCr420 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.88 NppStatus nppiRGBToYCbCr422\_8u\_C3C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned RGB to 2 channel chroma packed YCbCr422 color conversion.  
images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.89 NppStatus nppiRGBToYCbCr\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned RGB to YCbCr color conversion, ignoring Alpha.  
Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.90 NppStatus nppiRGBToYCbCr\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to packed YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



#### 7.4.3.91 **NppStatus nppiRGBToYCbCr\_8u\_P3R** (const Npp8u \*const *pSrc*, int *nSrcStep*, Npp8u \*\**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel planar 8-bit unsigned RGB to YCbCr color conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.4.3.92 **NppStatus nppiSqrIntegral\_8u32s32f\_C1R** (Npp8u \**pSrc*, int *nSrcStep*, Npp32s \**pDst*, int *nDstStep*, Npp32f \**pSqr*, int *nSqrStep*, NppiSize *oSrcROI*, Npp32s *val*, Npp32f *valSqr*, Npp32s *integralImageNewHeight*)

SqrIntegral Transforms an image to integral and integral of pixel squares representation.

This function assumes that the integral and integral of squares images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pSqr* Destination-Image Pointer.  
*nSqrStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).  
*val* The value to add to pDst image pixels  
*valSqr* The value to add to pSqr image pixels  
*integralImageNewHeight* Extended height of output surfaces (needed by transpose in primitive)

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.4.3.93 **NppStatus nppiSum\_8u64s\_C1R** (const Npp8u \**pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \**pDeviceBuffer*, Npp64s \**pSum*)

8-bit unsigned image sum with 64-bit long long result.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*\*pSum* Contains computed sum.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.94** `NppStatus nppiSum_8u64s_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64s aSum[4])`

4 channel 8-bit unsigned image sum with 64-bit long long result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*aSum* Array contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.95** `NppStatus nppiSum_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)`

8-bit unsigned image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*\*pSum* Contains computed sum.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.96 **NppStatus nppiSum\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[4])

4 channel 8-bit unsigned image sum with 64-bit double precision result.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*aSum* Array contains computed sum for each channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.97 **NppStatus nppiSumWindowColumn\_8u32f\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by *nMaskSize* and *nAnchor*.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.98 **NppStatus nppiSumWindowRow\_8u32f\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oROI* Region-of-Interest (ROI).  
*nMaskSize* Length of the linear kernel array.  
*nAnchor* X offset of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.99 NppStatus nppiYCbCr420ToRGB\_8u\_P3C3R (const Npp8u \*const \*pSrc, int nSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr420 to packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.4.3.100 NppStatus nppiYCbCr420ToYCbCr411\_8u\_P3P2R (const Npp8u \*const \*pSrc, int aSrcStep[3], Npp8u \*pDstY, int nDstYStep, Npp8u \*pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr:420 to YCbCr:411 resampling.

**Parameters:**

*pSrc* Array of pointers to the source image planes.  
*aSrcStep* Array with distances in bytes between starts of consecutive lines of the source image planes.  
*pDstY* Destination-Image Pointer. Y-channel.  
*nDstYStep* Destination-Image Line Step. Y-channel.  
*pDstCbCr* Destination-Image Pointer. CbCr image.  
*nDstCbCrStep* Destination-Image Line Step. CbCr image.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.101 NppStatus nppiYCbCr420ToYCbCr422\_8u\_P3R (const Npp8u \*const \*pSrc, int nSrcStep[3], Npp8u \*\*pDst, int nDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr:420 to YCbCr:422 resampling.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.102 NppStatus nppiYCbCr422ToRGB\_8u\_C2C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned YCbCr422 to 3 channel packed RGB color conversion.  
images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.103 NppStatus nppiYCbCr422ToYCbCr411\_8u\_P3R (const Npp8u \*const \*pSrc, int nSrcStep[3], Npp8u \*\*pDst, int nDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr:422 to YCbCr:411 resampling.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.4.3.104 NppStatus nppiYCbCr422ToYCbCr420\_8u\_P3R (const Npp8u \*const \*pSrc, int nSrcStep[3], Npp8u \*\*pDst, int nDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr:422 to YCbCr:420 resampling.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.105 NppStatus nppiYCbCrToRGB\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed YCbCr to RGB color conversion, not affecting Alpha.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.106 NppStatus nppiYCbCrToRGB\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed YCbCr to RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.4.3.107 NppStatus nppiYCbCrToRGB\_8u\_P3R (const Npp8u \*const \*pSrc, int nSrcStep, Npp8u \*\*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr to RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.5 Memory Management

Routines for allocating and deallocating pitched image storage.

### Functions

- void [nppiFree](#) (void \*pData)  
*Free method for any 2D allocated memory.*

### Image-Memory Allocation

ImageAllocator methods for 2D arrays of data.

The allocators have width and height parameters to specify the size of the image data being allocated. They return a pointer to the newly created memory and return the numbers of bytes between successive lines.

If the memory allocation failed due to lack of free device memory or device memory fragmentation the routine returns 0.

All allocators return memory with line strides that are beneficial for performance. It is not mandatory to use these allocators. Any valid CUDA device-memory pointers can be used by the NPP primitives and there are no restrictions on line strides.

- [Npp8u \\* nppiMalloc\\_8u\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C2](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C3](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C4](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 8-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C2](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C3](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C4](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit unsigned image memory allocator.*
- [Npp16s \\* nppiMalloc\\_16s\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit signed image memory allocator.*



- `Npp16s * nppiMalloc_16s_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed image memory allocator.*
- `Npp16s * nppiMalloc_16s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*1 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed complex image memory allocator.*
- `Npp32s * nppiMalloc_32s_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit signed image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit integer complex image memory allocator.*
- `Npp32f * nppiMalloc_32f_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit floating point image memory allocator.*

- `Npp32fc * nppiMalloc_32fc_C1` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C2` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*2 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C3` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*3 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C4` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*4 channel 32-bit float complex image memory allocator.*

### 7.5.1 Detailed Description

Routines for allocating and deallocating pitched image storage.

These methods are provided for convenience. They allocate memory that may contain additional padding bytes at the end of each line of pixels. Though padding is not necessary for any of the NPP image-processing primitives to work correctly, its absence may cause severe performance degradation compared to properly padded images.

### 7.5.2 Function Documentation

#### 7.5.2.1 void nppiFree (void \* *pData*)

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the `nppiMalloc_<modifier>` methods.

##### Parameters:

*pData* A pointer to memory allocated using `nppiMalloc_<modifier>`.

#### 7.5.2.2 Npp16s\* nppiMalloc\_16s\_C1 (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)

16-bit signed image memory allocator.

##### Parameters:

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

##### Returns:

Pointer to new image data.

**7.5.2.3 Npp16s\* nppiMalloc\_16s\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.4 Npp16s\* nppiMalloc\_16s\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.5 Npp16sc\* nppiMalloc\_16sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

1 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.6 Npp16sc\* nppiMalloc\_16sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.7 Npp16sc\* nppiMalloc\_16sc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.8 Npp16sc\* nppiMalloc\_16sc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.9 Npp16u\* nppiMalloc\_16u\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.10 Npp16u\* nppiMalloc\_16u\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.11 Npp16u\* nppiMalloc\_16u\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.12 Npp16u\* nppiMalloc\_16u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.13 Npp32f\* nppiMalloc\_32f\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.14 Npp32f\* nppiMalloc\_32f\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.15 Npp32f\* nppiMalloc\_32f\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.16 Npp32f\* nppiMalloc\_32f\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.17 Npp32fc\* nppiMalloc\_32fc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.18 Npp32fc\* nppiMalloc\_32fc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.19 Npp32fc\* nppiMalloc\_32fc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.20 Npp32fc\* nppiMalloc\_32fc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.21 Npp32s\* nppiMalloc\_32s\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.22 Npp32s\* nppiMalloc\_32s\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.23 Npp32s\* nppiMalloc\_32s\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.



**7.5.2.24 Npp32sc\* nppiMalloc\_32sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.25 Npp32sc\* nppiMalloc\_32sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.26 Npp32sc\* nppiMalloc\_32sc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.27 Npp32sc\* nppiMalloc\_32sc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.28 Npp8u\* nppiMalloc\_8u\_C1 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.29 Npp8u\* nppiMalloc\_8u\_C2 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

2 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.30 Npp8u\* nppiMalloc\_8u\_C3 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

3 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.5.2.31 Npp8u\* nppiMalloc\_8u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

## 7.6 Data-Exchange and Initialization

Primitives for initialization, copying and converting image data.

### Image-Memory Set

Set methods for images of various types.

Images are passed to these primitives via a pointer to the image data (first pixel in the ROI) and a step-width, i.e. the number of bytes between successive lines. The size of the area to be set (region-of-interest, ROI) is specified via a [NppiSize](#) struct. In addition to the image data and ROI, all methods have a parameter to specify the value being set. In case of single channel images this is a single value, in case of multi-channel, an array of values is passed.

- [NppStatus nppiSet\\_8s\\_C1R](#) ([Npp8s](#) nValue, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit image set.*
- [NppStatus nppiSet\\_8s\\_C2R](#) ([Npp8s](#) aValue[2], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit two-channel image set.*
- [NppStatus nppiSet\\_8s\\_C3R](#) ([Npp8s](#) aValue[3], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit three-channel image set.*
- [NppStatus nppiSet\\_8s\\_C4R](#) ([Npp8s](#) aValue[4], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit four-channel image set.*
- [NppStatus nppiSet\\_8s\\_AC4R](#) ([Npp8s](#) aValue[3], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit four-channel image set ignoring alpha channel.*
- [NppStatus nppiSet\\_8u\\_C1R](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C1MR](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C4R](#) (const [Npp8u](#) aValues[4], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C4MR](#) (const [Npp8u](#) aValues[4], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_AC4R](#) (const [Npp8u](#) aValues[3], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 8-bit unsigned image set method, not affecting Alpha channel.*
- [NppStatus nppiSet\\_8u\\_AC4MR](#) (const [Npp8u](#) aValues[3], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)

*Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.*

- **NppStatus nppiSet\_8u\_C4CR** (**Npp8u** nValue, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_16u\_C1R** (**Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C1MR** (**Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C2R** (const **Npp16u** aValues[2], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C4R** (const **Npp16u** aValues[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C4MR** (const **Npp16u** aValues[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_AC4R** (const **Npp16u** aValues[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_16u\_AC4MR** (const **Npp16u** aValues[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_16u\_C4CR** (**Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_16s\_C1R** (**Npp16s** nValue, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit image set.*
- **NppStatus nppiSet\_16s\_C1MR** (**Npp16s** nValue, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 16-bit image set.*
- **NppStatus nppiSet\_16s\_C2R** (const **Npp16s** aValues[2], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 16-bit image set.*
- **NppStatus nppiSet\_16s\_C4R** (const **Npp16s** aValues[4], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image set.*

- `NppStatus nppiSet_16s_C4MR` (const `Npp16s` aValues[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set.*
- `NppStatus nppiSet_16s_AC4R` (const `Npp16s` aValues[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_16s_AC4MR` (const `Npp16s` aValues[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_16s_C4CR` (`Npp16s` nValue, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image set affecting only single channel.*
- `NppStatus nppiSet_16sc_C1R` (`Npp16sc` oValue, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer image set.*
- `NppStatus nppiSet_16sc_C2R` (`Npp16sc` aValue[2], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer two-channel image set.*
- `NppStatus nppiSet_16sc_C3R` (`Npp16sc` aValue[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer three-channel image set.*
- `NppStatus nppiSet_16sc_AC4R` (`Npp16sc` aValue[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer four-channel image set ignoring alpha.*
- `NppStatus nppiSet_16sc_C4R` (`Npp16sc` aValue[4], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer four-channel image set.*
- `NppStatus nppiSet_32s_C1R` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit image set.*
- `NppStatus nppiSet_32s_C1MR` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 32-bit image set.*
- `NppStatus nppiSet_32s_C4R` (const `Npp32s` aValues[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image set.*
- `NppStatus nppiSet_32s_C4MR` (const `Npp32s` aValues[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit image set.*

- **NppStatus nppiSet\_32s\_AC4R** (const **Npp32s** aValues[3], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_32s\_AC4MR** (const **Npp32s** aValues[3], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_32s\_C4CR** (**Npp32s** nValue, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_32sc\_C1R** (**Npp32sc** oValue, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_C2R** (**Npp32sc** aValue[2], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_C3R** (**Npp32sc** aValue[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_C4R** (**Npp32sc** aValue[4], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_AC4R** (**Npp32sc** aValue[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit complex integer four-channel image set ignoring alpha.*
- **NppStatus nppiSet\_32f\_C1R** (**Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C1MR** (**Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C4R** (const **Npp32f** aValues[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C4MR** (const **Npp32f** aValues[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_AC4R** (const **Npp32f** aValues[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set method, not affecting Alpha channel.*

- **NppStatus nppiSet\_32f\_AC4MR** (const **Npp32f** aValues[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_32f\_C4CR** (**Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set affecting only single channel.*
- **NppStatus nppiSet\_32fc\_C1R** (**Npp32fc** oValue, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C2R** (**Npp32fc** aValue[2], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C3R** (**Npp32fc** aValue[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C4R** (**Npp32fc** aValue[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_AC4R** (**Npp32fc** aValue[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit complex four-channel image set ignoring alpha.*

## Image-Memory Copy

Copy methods for images of various types.

In addition to routines for copying pixels of identical layout from one image to another, there are copy routines for select channels as well as packed-planar conversions:

- Select channel to multi-channel copy. E.g. given a three-channel source and destination image one may copy the second channel of the source to the third channel of the destination.
- Single channel to multi-channel copy. E.g. given a single-channel source and a four-channel destination, one may copy the contents of the single-channel source to the second channel of the destination.
- Select channel to single-channel copy. E.g. given a three-channel source and a single-channel destination one may copy the third channel of the source to the destination.
- Multi-channel to planar copy. These copy operations split a multi-channel image into a set of single-channel images.
- Planar image to multi-channel copy. These copy routines combine separate color-planes (single channel images) into a single multi-channel image.
- **NppStatus nppiCopy\_8s\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit image copy.*



- `NppStatus nppiCopy_8s_C2R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 8-bit image copy.*
- `NppStatus nppiCopy_8s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 8-bit image copy.*
- `NppStatus nppiCopy_8s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 8-bit image copy.*
- `NppStatus nppiCopy_8s_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 8-bit image copy, ignoring alpha channel.*
- `NppStatus nppiCopy_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned image copy, not affecting Alpha channel.*
- `NppStatus nppiCopy_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit unsigned image copy.*
- `NppStatus nppiCopy_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image copy.*
- `NppStatus nppiCopy_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image copy, not affecting Alpha channel.*
- `NppStatus nppiCopy_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit image copy.*
- `NppStatus nppiCopy_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image copy.*
- `NppStatus nppiCopy_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 16-bit image copy, not affecting Alpha.*

- `NppStatus nppiCopy_16sc_C1R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*16-bit complex image copy.*

- `NppStatus nppiCopy_16sc_C2R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Two-channel 16-bit complex image copy.*

- `NppStatus nppiCopy_16sc_C3R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 16-bit complex image copy.*

- `NppStatus nppiCopy_16sc_C4R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit complex image copy.*

- `NppStatus nppiCopy_16sc_AC4R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit complex image copy, ignoring alpha.*

- `NppStatus nppiCopy_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit image copy.*

- `NppStatus nppiCopy_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit image copy.*

- `NppStatus nppiCopy_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit image copy, not affecting Alpha.*

- `NppStatus nppiCopy_32sc_C1R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit complex image copy.*

- `NppStatus nppiCopy_32sc_C2R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Two-channel 32-bit complex image copy.*

- `NppStatus nppiCopy_32sc_C3R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit complex image copy.*

- `NppStatus nppiCopy_32sc_C4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit complex image copy.*

- `NppStatus nppiCopy_32sc_AC4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit complex image copy, ignoring alpha.*
- `NppStatus nppiCopy_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy, not affecting Alpha.*
- `NppStatus nppiCopy_32fc_C1R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C2R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C3R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_AC4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit floating-point complex image copy, ignoring alpha.*
- `NppStatus nppiCopy_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_C3MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation three channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_C4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation four channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_AC4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.*

- `NppStatus nppiCopy_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* 16-bit unsigned image copy.
- `NppStatus nppiCopy_16u_C3MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* three channel 16-bit unsigned image copy.
- `NppStatus nppiCopy_16u_C4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 16-bit unsigned image copy.
- `NppStatus nppiCopy_16u_AC4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 16-bit unsigned image copy, ignoring alpha.
- `NppStatus nppiCopy_16s_C1MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* 16-bit signed image copy.
- `NppStatus nppiCopy_16s_C3MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* three channel 16-bit signed image copy.
- `NppStatus nppiCopy_16s_C4MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 16-bit signed image copy.
- `NppStatus nppiCopy_16s_AC4MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 16-bit signed image copy, ignoring alpha.
- `NppStatus nppiCopy_32s_C1MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* 32-bit signed image copy.
- `NppStatus nppiCopy_32s_C3MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* three channel 32-bit signed image copy.
- `NppStatus nppiCopy_32s_C4MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 32-bit signed image copy.
- `NppStatus nppiCopy_32s_AC4MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked Operation* four channel 32-bit signed image copy, ignoring alpha.
- `NppStatus nppiCopy_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* 32-bit float image copy.

- `NppStatus nppiCopy_32f_C3MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* three channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_C4MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_AC4MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit float image copy, ignoring alpha.

- `NppStatus nppiCopy_8u_C3CR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 8-bit unsigned image copy for three-channel images.

- `NppStatus nppiCopy_8u_C4CR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 8-bit unsigned image copy for four-channel images.

- `NppStatus nppiCopy_16s_C3CR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 16-bit signed image copy for three-channel images.

- `NppStatus nppiCopy_16s_C4CR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 16-bit signed image copy for four-channel images.

- `NppStatus nppiCopy_16u_C3CR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 16-bit unsigned image copy for three-channel images.

- `NppStatus nppiCopy_16u_C4CR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 16-bit unsigned image copy for four-channel images.

- `NppStatus nppiCopy_32s_C3CR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 32-bit signed image copy for three-channel images.

- `NppStatus nppiCopy_32s_C4CR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 32-bit signed image copy for four-channel images.

- `NppStatus nppiCopy_32f_C3CR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel* 32-bit float image copy for three-channel images.

- `NppStatus nppiCopy_32f_C4CR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Select-channel 32-bit float image copy for four-channel images.*
- `NppStatus nppiCopy_8u_C3C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel to single-channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_C4C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel to single-channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_16s_C3C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel to single-channel 16-bit signed image copy.*
- `NppStatus nppiCopy_16s_C4C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel to single-channel 16-bit signed image copy.*
- `NppStatus nppiCopy_16u_C3C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel to single-channel 16-bit unsigned image copy.*
- `NppStatus nppiCopy_16u_C4C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel to single-channel 16-bit unsigned image copy.*
- `NppStatus nppiCopy_32s_C3C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel to single-channel 32-bit signed image copy.*
- `NppStatus nppiCopy_32s_C4C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel to single-channel 32-bit signed image copy.*
- `NppStatus nppiCopy_32f_C3C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel to single-channel 32-bit float image copy.*
- `NppStatus nppiCopy_32f_C4C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel to single-channel 32-bit float image copy.*
- `NppStatus nppiCopy_8u_C1C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single-channel to three-channel 8-bit unsigned image copy.*
- `NppStatus nppiCopy_8u_C1C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single-channel to four-channel 8-bit unsigned image copy.*

- **NppStatus nppiCopy\_16s\_C1C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to three-channel 16-bit signed image copy.*
- **NppStatus nppiCopy\_16s\_C1C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to four-channel 16-bit signed image copy.*
- **NppStatus nppiCopy\_16u\_C1C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to three-channel 16-bit unsigned image copy.*
- **NppStatus nppiCopy\_16u\_C1C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to four-channel 16-bit unsigned image copy.*
- **NppStatus nppiCopy\_32s\_C1C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to three-channel 32-bit signed image copy.*
- **NppStatus nppiCopy\_32s\_C1C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to four-channel 32-bit signed image copy.*
- **NppStatus nppiCopy\_32f\_C1C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to three-channel 32-bit float image copy.*
- **NppStatus nppiCopy\_32f\_C1C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to four-channel 32-bit float image copy.*
- **NppStatus nppiCopy\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_8u\_C4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_16s\_C3P3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_16s\_C4P4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_16u\_C3P3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)

*Three-channel 16-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_16u_C4P4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_32s_C3P3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_32s_C4P4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_32f_C3P3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit float packed to planar image copy.*

- `NppStatus nppiCopy_32f_C4P4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit float packed to planar image copy.*

- `NppStatus nppiCopy_8u_P3C3R` (const `Npp8u` \*const aSrc[3], int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 8-bit unsigned planar to packed image copy.*

- `NppStatus nppiCopy_8u_P4C4R` (const `Npp8u` \*const aSrc[4], int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 8-bit unsigned planar to packed image copy.*

- `NppStatus nppiCopy_16u_P3C3R` (const `Npp16u` \*const aSrc[3], int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 16-bit unsigned planar to packed image copy.*

- `NppStatus nppiCopy_16u_P4C4R` (const `Npp16u` \*const aSrc[4], int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit unsigned planar to packed image copy.*

- `NppStatus nppiCopy_16s_P3C3R` (const `Npp16s` \*const aSrc[3], int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 16-bit signed planar to packed image copy.*

- `NppStatus nppiCopy_16s_P4C4R` (const `Npp16s` \*const aSrc[4], int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit signed planar to packed image copy.*

- `NppStatus nppiCopy_32s_P3C3R` (const `Npp32s` \*const aSrc[3], int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit signed planar to packed image copy.*



- `NppStatus nppiCopy_32s_P4C4R` (const `Npp32s` \*const aSrc[4], int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit signed planar to packed image copy.*
- `NppStatus nppiCopy_32f_P3C3R` (const `Npp32f` \*const aSrc[3], int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 32-bit float planar to packed image copy.*
- `NppStatus nppiCopy_32f_P4C4R` (const `Npp32f` \*const aSrc[4], int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit float planar to packed image copy.*

## Bit-Depth Conversion

Convert bit-depth up and down.

The integer conversion methods do not involve any scaling. Conversions that reduce bit-depth saturate values exceeding the reduced range to the range's maximum/minimum value. When converting from floating-point values to integer values, a rounding mode can be specified. After rounding to integer values the values get saturated to the destination data type's range.

- `NppStatus nppiConvert_8u16u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiConvert_16u8u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit unsigned to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_8u16u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiConvert_16u8u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_8u16u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16u8u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_8u16s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*8-bit unsigned to 16-bit signed conversion.*

- **NppStatus nppiConvert\_16s8u\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit signed to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_8u16s\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned to 16-bit signed conversion.*
- **NppStatus nppiConvert\_16s8u\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8u16s\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_16s8u\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_16s32f\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit signed to 32-bit floating point conversion.*
- **NppStatus nppiConvert\_32f16s\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*32-bit floating point to 16-bit conversion.*
- **NppStatus nppiConvert\_8u32f\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit unsigned to 32-bit floating point conversion.*
- **NppStatus nppiConvert\_16u32f\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit unsigned to 32-bit floating point conversion.*
- **NppStatus nppiConvert\_32f16u\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f8u\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*32-bit floating point to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_16u32s\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit unsigned to 32-bit signed conversion.*
- **NppStatus nppiConvert\_16s32s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit to 32-bit conversion.*

## Copy Const Border

Methods for copying images and padding borders with a constant, user-specifiable color.

- **NppStatus nppiCopyConstBorder\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp8u** nValue)

*8-bit unsigned image copy width constant border color.*

- **NppStatus nppiCopyConstBorder\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp8u** aValue[4])

*4channel 8-bit unsigned image copy with constant border color.*

- **NppStatus nppiCopyConstBorder\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp8u** aValue[3])

*4 channel 8-bit unsigned image copy with constant border color.*

- **NppStatus nppiCopyConstBorder\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp32s** nValue)

*32-bit image copy with constant border color.*

## Image Transpose

Methods for transposing images of various types.

Like matrix transpose, image transpose is a mirror along the image's diagonal (upper-left to lower-right corner).

- **NppStatus nppiTranspose\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI)

*8-bit image transpose.*

## Image Color Channel Swap

Methods for exchanging the color channels of an image.

The methods support arbitrary permutations of the original channels, including replication.

- **NppStatus nppiSwapChannels\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])

*4 channel 8-bit unsigned swap channels, in-place.*

### 7.6.1 Detailed Description

Primitives for initialization, copying and converting image data.

## 7.6.2 Function Documentation

### 7.6.2.1 `NppStatus nppiConvert_16s32f_C1R (const Npp16s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

16-bit singedto 32-bit floating point conversion.

For detailed documentation see `nppiConverte_8u16u_C1R()`.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.2 `NppStatus nppiConvert_16s32s_C1R (const Npp16s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

16-bit to 32-bit conversion.

For detailed documentation see `nppiConvert_8u16u_C1R()`.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.3 `NppStatus nppiConvert_16s8u_AC4R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For detailed documentation see `nppiConverte_8u16u_C1R()`.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.4 NppStatus nppiConvert\_16s8u\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit signed to 8-bit unsigned conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.5 NppStatus nppiConvert\_16s8u\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.6 NppStatus nppiConvert\_16u32f\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit unsigned to 32-bit floating point conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.7 NppStatus nppiConvert\_16u32s\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit unsigned to 32-bit signed conversion.

For detailed documentation see [nppiConverte\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.8 NppStatus nppiConvert\_16u8u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.9 NppStatus nppiConvert\_16u8u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit unsigned to 8-bit unsigned conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.10 NppStatus nppiConvert\_16u8u\_C4R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned to 8-bit unsigned conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.11 NppStatus nppiConvert\_32f16s\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

32-bit floating point to 16-bit conversion.

For detailed documentation see [nppiConverte\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.12 NppStatus nppiConvert\_32f16u\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

32-bit floating point to 16-bit unsigned conversion.

For detailed documentation see `nppiConverte_8u16u_C1R()`.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.13 NppStatus nppiConvert\_32f8u\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

32-bit floating point to 8-bit unsigned conversion.

For detailed documentation see `nppiConverte_8u16u_C1R()`.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.6.2.14 NppStatus nppiConvert\_8u16s\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.15 NppStatus nppiConvert\_8u16s\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit unsigned to 16-bit signed conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.16 NppStatus nppiConvert\_8u16s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned to 16-bit signed conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.17 NppStatus nppiConvert\_8u16u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.18 NppStatus nppiConvert\_8u16u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.19 NppStatus nppiConvert\_8u16u\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned to 16-bit unsigned conversion.

For detailed documentation see [nppiConvert\\_8u16u\\_C1R\(\)](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.20 NppStatus nppiConvert\_8u32f\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

8-bit unsigned to 32-bit floating point conversion.

For detailed documentation see `nppiConverte_8u16u_C1R()`.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.21 NppStatus nppiCopy\_16s\_AC4MR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation four channel 16-bit signed image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.22 NppStatus nppiCopy\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.23 NppStatus nppiCopy\_16s\_C1C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to three-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.24 NppStatus nppiCopy\_16s\_C1C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.25** `NppStatus nppiCopy_16s_C1MR (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

**Masked Operation** 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.26** `NppStatus nppiCopy_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

16-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.27** `NppStatus nppiCopy_16s_C3C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.28 NppStatus nppiCopy\_16s\_C3CR (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 16-bit signed image copy for three-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.29 NppStatus nppiCopy\_16s\_C3MR (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 16-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.30 NppStatus nppiCopy\_16s\_C3P3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed packed to planar image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.31 NppStatus nppiCopy\_16s\_C4C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel to single-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.32 NppStatus nppiCopy\_16s\_C4CR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 16-bit signed image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.33 NppStatus nppiCopy\_16s\_C4MR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation four channel 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.34 **NppStatus nppiCopy\_16s\_C4P4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \*const *aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit signed packed to planar image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.35 **NppStatus nppiCopy\_16s\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.36 **NppStatus nppiCopy\_16s\_P3C3R** (const Npp16s \*const *aSrc*[3], int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed planar to packed image copy.

##### Parameters:

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes



**7.6.2.37 NppStatus nppiCopy\_16s\_P4C4R (const Npp16s \*const aSrc[4], int nSrcStep, Npp16s \*pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 16-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.38 NppStatus nppiCopy\_16sc\_AC4R (const Npp16sc \*pSrc, int nSrcStep, Npp16sc \*pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 16-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.39 NppStatus nppiCopy\_16sc\_C1R (const Npp16sc \*pSrc, int nSrcStep, Npp16sc \*pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.40 NppStatus nppiCopy\_16sc\_C2R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.41 NppStatus nppiCopy\_16sc\_C3R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.42 NppStatus nppiCopy\_16sc\_C4R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.43** `NppStatus nppiCopy_16u_AC4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 16-bit unsigned image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.44** `NppStatus nppiCopy_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.45** `NppStatus nppiCopy_16u_C1C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to three-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.46 NppStatus nppiCopy\_16u\_C1C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to four-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.47 NppStatus nppiCopy\_16u\_C1MR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked Operation 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.48 NppStatus nppiCopy\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.49 NppStatus nppiCopy\_16u\_C3C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.50 NppStatus nppiCopy\_16u\_C3CR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 16-bit unsigned image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.51 NppStatus nppiCopy\_16u\_C3MR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked Operation three channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.52 NppStatus nppiCopy\_16u\_C3P3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \*const *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.53 NppStatus nppiCopy\_16u\_C4C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.54 NppStatus nppiCopy\_16u\_C4CR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 16-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.55** `NppStatus nppiCopy_16u_C4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.56** `NppStatus nppiCopy_16u_C4P4R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.57** `NppStatus nppiCopy_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.58 **NppStatus nppiCopy\_16u\_P3C3R** (const Npp16u \*const *aSrc*[3], int *nSrcStep*, Npp16u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.59 **NppStatus nppiCopy\_16u\_P4C4R** (const Npp16u \*const *aSrc*[4], int *nSrcStep*, Npp16u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.60 **NppStatus nppiCopy\_32f\_AC4MR** (const Npp32f \**pSrc*, int *nSrcStep*, Npp32f \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \**pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 32-bit float image copy, ignoring alpha.

#### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.6.2.61 NppStatus nppiCopy\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit floating point image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.62 NppStatus nppiCopy\_32f\_C1C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to three-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.63 NppStatus nppiCopy\_32f\_C1C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to four-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.64 NppStatus nppiCopy\_32f\_C1MR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.65 NppStatus nppiCopy\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.66 NppStatus nppiCopy\_32f\_C3C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel to single-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.67 **NppStatus nppiCopy\_32f\_C3CR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit float image copy for three-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.68 **NppStatus nppiCopy\_32f\_C3MR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 32-bit float image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.69 **NppStatus nppiCopy\_32f\_C3P3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit float packed to planar image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.70 **NppStatus nppiCopy\_32f\_C4C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel to single-channel 32-bit float image copy.

##### Parameters:

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.71 **NppStatus nppiCopy\_32f\_C4CR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit float image copy for four-channel images.

##### Parameters:

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Select-Channel Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.72 **NppStatus nppiCopy\_32f\_C4MR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 32-bit float image copy.

##### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.73 NppStatus nppiCopy\_32f\_C4P4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \*const *aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit float packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.74 NppStatus nppiCopy\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.75 NppStatus nppiCopy\_32f\_P3C3R (const Npp32f \*const *aSrc*[3], int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit float planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.76 **NppStatus nppiCopy\_32f\_P4C4R** (const Npp32f \*const *aSrc*[4], int *nSrcStep*, Npp32f \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit float planar to packed image copy.

##### Parameters:

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.77 **NppStatus nppiCopy\_32fc\_AC4R** (const Npp32fc \**pSrc*, int *nSrcStep*, Npp32fc \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

##### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.78 **NppStatus nppiCopy\_32fc\_C1R** (const Npp32fc \**pSrc*, int *nSrcStep*, Npp32fc \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit floating-point complex image copy.

##### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.79 NppStatus nppiCopy\_32fc\_C2R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.80 NppStatus nppiCopy\_32fc\_C3R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.81 NppStatus nppiCopy\_32fc\_C4R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.82** `NppStatus nppiCopy_32s_AC4MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 32-bit signed image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.83** `NppStatus nppiCopy_32s_AC4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.84** `NppStatus nppiCopy_32s_C1C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to three-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.6.2.85 NppStatus nppiCopy\_32s\_C1C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.86 NppStatus nppiCopy\_32s\_C1MR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.87 NppStatus nppiCopy\_32s\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.88 **NppStatus nppiCopy\_32s\_C3C1R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel to single-channel 32-bit signed image copy.

##### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.89 **NppStatus nppiCopy\_32s\_C3CR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit signed image copy for three-channel images.

##### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.90 **NppStatus nppiCopy\_32s\_C3MR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 32-bit signed image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.91 NppStatus nppiCopy\_32s\_C3P3R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* const aDst[3], int nDstStep, NppiSize oSizeROI)**

Three-channel 32-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.92 NppStatus nppiCopy\_32s\_C4C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel to single-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.93 NppStatus nppiCopy\_32s\_C4CR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 32-bit signed image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.6.2.94 **NppStatus nppiCopy\_32s\_C4MR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 32-bit signed image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.95 **NppStatus nppiCopy\_32s\_C4P4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *const aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit signed packed to planar image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.6.2.96 **NppStatus nppiCopy\_32s\_C4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.97 NppStatus nppiCopy\_32s\_P3C3R (const Npp32s \*const aSrc[3], int nSrcStep, Npp32s \*pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.98 NppStatus nppiCopy\_32s\_P4C4R (const Npp32s \*const aSrc[4], int nSrcStep, Npp32s \*pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.99 NppStatus nppiCopy\_32sc\_AC4R (const Npp32sc \*pSrc, int nSrcStep, Npp32sc \*pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.100 NppStatus nppiCopy\_32sc\_C1R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.101 NppStatus nppiCopy\_32sc\_C2R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.102 NppStatus nppiCopy\_32sc\_C3R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.103 NppStatus nppiCopy\_32sc\_C4R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.104 NppStatus nppiCopy\_8s\_AC4R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit image copy, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.105 NppStatus nppiCopy\_8s\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.106 NppStatus nppiCopy\_8s\_C2R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.107 NppStatus nppiCopy\_8s\_C3R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.108 NppStatus nppiCopy\_8s\_C4R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.6.2.109** `NppStatus nppiCopy_8u_AC4MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 8-bit unsigned image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.110** `NppStatus nppiCopy_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.111** `NppStatus nppiCopy_8u_C1C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to three-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.112 NppStatus nppiCopy\_8u\_C1C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to four-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.113 NppStatus nppiCopy\_8u\_C1MR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked Operation 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.114 NppStatus nppiCopy\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.115** `NppStatus nppiCopy_8u_C3C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.116** `NppStatus nppiCopy_8u_C3CR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 8-bit unsigned image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.117** `NppStatus nppiCopy_8u_C3MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.118** `NppStatus nppiCopy_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.119** `NppStatus nppiCopy_8u_C4C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.120** `NppStatus nppiCopy_8u_C4CR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 8-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.121** `NppStatus nppiCopy_8u_C4MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.122** `NppStatus nppiCopy_8u_C4P4R (const Npp8u * pSrc, int nSrcStep, Npp8u * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.123** `NppStatus nppiCopy_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.124 **NppStatus nppiCopy\_8u\_P3C3R** (const Npp8u \*const *aSrc*[3], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.125 **NppStatus nppiCopy\_8u\_P4C4R** (const Npp8u \*const *aSrc*[4], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.126 **NppStatus nppiCopyConstBorder\_32s\_C1R** (const Npp32s \**pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s \**pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*, Npp32s *nValue*)

32-bit image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*nValue* Border luminance value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.127** `NppStatus nppiCopyConstBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])`

4 channel 8-bit unsigned image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.128** `NppStatus nppiCopyConstBorder_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp8u nValue)`

8-bit unsigned image copy width constant border color.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

***nTopBorderHeight*** Height (in pixels) of the top border. The height of the border at the bottom of the destination ROI is implicitly defined by the size of the source ROI:  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

***nLeftBorderWidth*** Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

***nValue*** The pixel value to be set for border pixels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.129** `NppStatus nppiCopyConstBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[4])`

4channel 8-bit unsigned image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Size of the destination region-of-interest.

***nTopBorderHeight*** Height of top border.

***nLeftBorderWidth*** Width of left border.

***aValue*** Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.130** `NppStatus nppiSet_16s_AC4MR (const Npp16s aValues[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

***aValues*** Three-channel array containing the pixel-value to be set.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oSizeROI*** [Region-of-Interest \(ROI\)](#).



*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.131 NppStatus nppiSet\_16s\_AC4R (const Npp16s aValues[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.132 NppStatus nppiSet\_16s\_C1MR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 16-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.133 NppStatus nppiSet\_16s\_C1R (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit image set.

**Parameters:**

*nValue* New pixel value.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.134 NppStatus nppiSet\_16s\_C2R (const Npp16s aValues[2], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.135 NppStatus nppiSet\_16s\_C4CR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the pDst pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass pDst unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass pDst + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.136 NppStatus nppiSet\_16s\_C4MR (const Npp16s *aValues*[4], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked 4 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.137 NppStatus nppiSet\_16s\_C4R (const Npp16s *aValues*[4], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.138 NppStatus nppiSet\_16sc\_AC4R (Npp16sc *aValue*[3], Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

16-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.139 NppStatus nppiSet\_16sc\_C1R (Npp16sc oValue, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer image set.

**Parameters:**

*oValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.140 NppStatus nppiSet\_16sc\_C2R (Npp16sc aValue[2], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer two-channel image set.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.141 NppStatus nppiSet\_16sc\_C3R (Npp16sc aValue[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer three-channel image set.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.142 **NppStatus nppiSet\_16sc\_C4R** (Npp16sc *aValue*[4], Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

16-bit complex integer four-channel image set.

#### Parameters:

*aValue* New pixel value.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.143 **NppStatus nppiSet\_16u\_AC4MR** (const Npp16u *aValues*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

#### Parameters:

*aValues* Three-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.  
*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.144 **NppStatus nppiSet\_16u\_AC4R** (const Npp16u *aValues*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

#### Parameters:

*aValues* Three-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.145 NppStatus nppiSet\_16u\_C1MR (Npp16u *nValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked 16-bit unsigned image set.

**Parameters:**

*nValue* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.146 NppStatus nppiSet\_16u\_C1R (Npp16u *nValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

16-bit unsigned image set.

**Parameters:**

*nValue* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.147 NppStatus nppiSet\_16u\_C2R (const Npp16u *aValues*[2], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 16-bit unsigned image set.

**Parameters:**

*aValues* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.148 NppStatus nppiSet\_16u\_C4CR (Npp16u *nValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 16-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.149 NppStatus nppiSet\_16u\_C4MR (const Npp16u *aValues*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked 4 channel 16-bit unsigned image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.150 NppStatus nppiSet\_16u\_C4R (const Npp16u *aValues*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 16-bit unsigned image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.151** `NppStatus nppiSet_32f_AC4MR (const Npp32f aValues[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.152** `NppStatus nppiSet_32f_AC4R (const Npp32f aValues[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.6.2.153 `NppStatus nppiSet_32f_C1MR (Npp32f nValue, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit floating point image set.

#### Parameters:

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.154 `NppStatus nppiSet_32f_C1R (Npp32f nValue, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit floating point image set.

#### Parameters:

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.155 `NppStatus nppiSet_32f_C4CR (Npp32f nValue, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

#### Parameters:

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.156** `NppStatus nppiSet_32f_C4MR (const Npp32f aValues[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set.

**Parameters:**

*aValues* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.157** `NppStatus nppiSet_32f_C4R (const Npp32f aValues[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set.

**Parameters:**

*aValues* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.158** `NppStatus nppiSet_32fc_AC4R (Npp32fc aValue[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit complex four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.159 NppStatus nppiSet\_32fc\_C1R (Npp32fc oValue, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 32-bit complex image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.160 NppStatus nppiSet\_32fc\_C2R (Npp32fc aValue[2], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.161 NppStatus nppiSet\_32fc\_C3R (Npp32fc aValue[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.162** `NppStatus nppiSet_32fc_C4R (Npp32fc aValue[4], Npp32fc *pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.163** `NppStatus nppiSet_32s_AC4MR (const Npp32s aValues[3], Npp32s *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, int nMaskStep)`

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.164** `NppStatus nppiSet_32s_AC4R (const Npp32s aValues[3], Npp32s *pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.165** `NppStatus nppiSet_32s_C1MR (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.166** `NppStatus nppiSet_32s_C1R (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.167 **NppStatus nppiSet\_32s\_C4CR** (Npp32s *nValue*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

#### Parameters:

*nValue* The pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.168 **NppStatus nppiSet\_32s\_C4MR** (const Npp32s *aValues*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 4 channel 32-bit image set.

#### Parameters:

*aValues* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.169 **NppStatus nppiSet\_32s\_C4R** (const Npp32s *aValues*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image set.

#### Parameters:

*aValues* New pixel value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.170** `NppStatus nppiSet_32sc_AC4R (Npp32sc aValue[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.171** `NppStatus nppiSet_32sc_C1R (Npp32sc oValue, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit complex integer image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.172** `NppStatus nppiSet_32sc_C2R (Npp32sc aValue[2], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Two channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.173 NppStatus nppiSet\_32sc\_C3R (Npp32sc aValue[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.174 NppStatus nppiSet\_32sc\_C4R (Npp32sc aValue[4], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.6.2.175 NppStatus nppiSet\_8s\_AC4R (Npp8s aValue[3], Npp8s \* pDst, int nDstStep, NppiSize oSizeROI)**

8-bit four-channel image set ignoring alpha channel.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.6.2.176 NppStatus nppiSet\_8s\_C1R (Npp8s *nValue*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit image set.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.177 NppStatus nppiSet\_8s\_C2R (Npp8s *aValue*[2], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit two-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.178 NppStatus nppiSet\_8s\_C3R (Npp8s *aValue*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit three-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.6.2.179 `NppStatus nppiSet_8s_C4R (Npp8s aValue[4], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit four-channel image set.

#### Parameters:

*aValue* The pixel value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.180 `NppStatus nppiSet_8u_AC4MR (const Npp8u aValues[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

#### Parameters:

*aValues* Three-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.  
*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.181 `NppStatus nppiSet_8u_AC4R (const Npp8u aValues[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

#### Parameters:

*aValues* Three-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.182 `NppStatus nppiSet_8u_C1MR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 8-bit unsigned image set.

The 8-bit mask image affects setting of the respective pixels in the destination image. If the mask value is zero (0) the pixel is not set, if the mask is non-zero, the corresponding destination pixel is set to specified value.

#### Parameters:

*nValue* The pixel value to be set.

*pDst* Pointer [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.183 `NppStatus nppiSet_8u_C1R (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit unsigned image set.

#### Parameters:

*nValue* The pixel value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.6.2.184 `NppStatus nppiSet_8u_C4CR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

#### Parameters:

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.185** `NppStatus nppiSet_8u_C4MR (const Npp8u aValues[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set.

**Parameters:**

*aValues* Four-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.  
*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.186** `NppStatus nppiSet_8u_C4R (const Npp8u aValues[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set.

**Parameters:**

*aValues* Four-channel array containing the pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.187** `NppStatus nppiSwapChannels_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 8-bit unsigned swap channels, in-place.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [3,2,1,0] converts this to ABGR channel order.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.6.2.188** `NppStatus npptTranspose_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oROI)`

8-bit image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.7 Arithmetic and Logical Operations

### AddC

Adds a constant value to each pixel of an image.

- `NppStatus nppiAddC_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C1IRSfs` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C3IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_AC4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C3IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)



*One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3IRSfs` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_AC4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_AC4IRSfs` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C4IRSfs` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pConstants, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_C3IRSfs` (const `Npp32sc` \*pConstants, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pConstants, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- **NppStatus nppiAddC\_32sc\_AC4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_C4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image add constant.*
- **NppStatus nppiAddC\_32f\_C1IR** (const **Npp32f** nConstant, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image add constant.*
- **NppStatus nppiAddC\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image add constant.*
- **NppStatus nppiAddC\_32f\_C3IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image add constant.*
- **NppStatus nppiAddC\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image add constant.*
- **NppStatus nppiAddC\_32f\_AC4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image add constant.*
- **NppStatus nppiAddC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image add constant.*
- **NppStatus nppiAddC\_32f\_C4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image add constant.*
- **NppStatus nppiAddC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- **NppStatus nppiAddC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*
- **NppStatus nppiAddC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*
- **NppStatus nppiAddC\_32fc\_C3IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*
- **NppStatus nppiAddC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.*
- **NppStatus nppiAddC\_32fc\_AC4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.*
- **NppStatus nppiAddC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*
- **NppStatus nppiAddC\_32fc\_C4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

## MulC

Multiplies each pixel of an image by a constant value.

- **NppStatus nppiMulC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_8u_C3IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_8u_AC4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_8u_C4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C3IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_AC4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16u\_C4IRSfs** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** nConstant, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C1IRSfs** (const **Npp16s** nConstant, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pConstants, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C3IRSfs** (const **Npp16s** \*pConstants, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pConstants, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_AC4IRSfs** (const **Npp16s** \*pConstants, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pConstants, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16s\_C4IRSfs** (const **Npp16s** \*pConstants, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** nConstant, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_C1IRSfs** (const **Npp16sc** nConstant, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pConstants, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_C3IRSfs** (const **Npp16sc** \*pConstants, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pConstants, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_AC4IRSfs** (const **Npp16sc** \*pConstants, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_C4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pConstants, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_16sc\_C4IRSfs** (const **Npp16sc** \*pConstants, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** nConstant, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_32s\_C1IRSfs** (const **Npp32s** nConstant, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_32s\_C3IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiMulC\_32s\_AC4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus nppiMulC\_32s\_AC4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32s\_C4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C3IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_AC4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_32sc\_C4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel image multiply by constant.*

- `NppStatus nppiMulC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel in place image multiply by constant.*

- `NppStatus nppiMulC_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pConstants, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point channel image multiply by constant.*

- `NppStatus nppiMulC_32f_C3IR` (const `Npp32f` \*pConstants, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point channel in place image multiply by constant.*

- `NppStatus nppiMulC_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pConstants, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image multiply by constant.*

- `NppStatus nppiMulC_32f_AC4IR` (const `Npp32f` \*pConstants, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.*

- `NppStatus nppiMulC_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pConstants, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel image multiply by constant.*

- `NppStatus nppiMulC_32f_C4IR` (const `Npp32f` \*pConstants, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel in place image multiply by constant.*

- `NppStatus nppiMulC_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` nConstant, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*

- `NppStatus nppiMulC_32fc_C1IR` (const `Npp32fc` nConstant, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*

- `NppStatus nppiMulC_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pConstants, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*

- `NppStatus nppiMulC_32fc_C3IR` (const `Npp32fc` \*pConstants, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)



*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*

- `NppStatus nppiMulC_32fc_AC4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pConstants, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.*

- `NppStatus nppiMulC_32fc_AC4IR` (const `Npp32fc` \*pConstants, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.*

- `NppStatus nppiMulC_32fc_C4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pConstants, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*

- `NppStatus nppiMulC_32fc_C4IR` (const `Npp32fc` \*pConstants, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*

## MulCScale

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

- `NppStatus nppiMulCScale_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_8u_C1IR` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_8u_C3IR` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_8u\_AC4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.*

- **NppStatus nppiMulCScale\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_8u\_C4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C3IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_AC4IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- **NppStatus nppiMulCScale\_16u\_C4IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

## SubC

Subtracts a constant value from each pixel of an image.

- **NppStatus** **nppiSubC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C3IRSfs** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_AC4IRSfs** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C4IRSfs** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_C3IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_AC4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_C4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_C3IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_AC4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C3IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_AC4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16sc_C4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus nppiSubC\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32s\_C3IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32s\_AC4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32s\_AC4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32s\_C4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C3IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- **NppStatus nppiSubC\_32sc\_AC4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32sc\_C4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppiSubC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image subtract constant.*
- **NppStatus nppiSubC\_32f\_C1IR** (const **Npp32f** nConstant, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image subtract constant.*
- **NppStatus nppiSubC\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image subtract constant.*
- **NppStatus nppiSubC\_32f\_C3IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image subtract constant.*
- **NppStatus nppiSubC\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image subtract constant.*
- **NppStatus nppiSubC\_32f\_AC4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image subtract constant.*
- **NppStatus nppiSubC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image subtract constant.*
- **NppStatus nppiSubC\_32f\_C4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image subtract constant.*
- **NppStatus nppiSubC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*



- **NppStatus nppiSubC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*
- **NppStatus nppiSubC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*
- **NppStatus nppiSubC\_32fc\_C3IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*
- **NppStatus nppiSubC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.*
- **NppStatus nppiSubC\_32fc\_AC4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.*
- **NppStatus nppiSubC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*
- **NppStatus nppiSubC\_32fc\_C4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

## DivC

Divides each pixel of an image by a constant value.

- **NppStatus nppiDivC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_8u_C3IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_8u_AC4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_8u_C4IRSfs` (const `Npp8u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C3IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_AC4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16u_C4IRSfs` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C3IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_AC4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C4IRSfs` (const `Npp16s` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C3IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pConstants, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C4IRSfs` (const `Npp16sc` \*pConstants, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3IRSfs` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_AC4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- **NppStatus nppiDivC\_32s\_AC4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32s\_C4IRSfs** (const **Npp32s** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C3IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_AC4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pConstants, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppiDivC\_32sc\_C4IRSfs** (const **Npp32sc** \*pConstants, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus nppiDivC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image divided by constant.*
- **NppStatus nppiDivC\_32f\_C1IR** (const **Npp32f** nConstant, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image divided by constant.*
- **NppStatus nppiDivC\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image divided by constant.*
- **NppStatus nppiDivC\_32f\_C3IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image divided by constant.*
- **NppStatus nppiDivC\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image divided by constant.*
- **NppStatus nppiDivC\_32f\_AC4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pConstants, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4IR** (const **Npp32f** \*pConstants, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*

- **NppStatus** **nppiDivC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.*
- **NppStatus** **nppiDivC\_32fc\_AC4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.*
- **NppStatus** **nppiDivC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pConstants, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus** **nppiDivC\_32fc\_C4IR** (const **Npp32fc** \*pConstants, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*

## AbsDiffC

Determines absolute difference between each pixel of an image and a constant value.

- **NppStatus** **nppiAbsDiffC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nConstant)  
*One 8-bit unsigned char channel image absolute difference with constant.*
- **NppStatus** **nppiAbsDiffC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32u** nConstant)  
*One 16-bit unsigned short channel image absolute difference with constant.*
- **NppStatus** **nppiAbsDiffC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nConstant)  
*One 32-bit floating point channel image absolute difference with constant.*

## Add Image

Pixel by pixel addition of two images.

- **NppStatus** **nppiAdd\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16u_C1IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16u_C3IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiAdd\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_16sc\_C4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- `NppStatus nppiAdd_32s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_AC4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_AC4IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C4IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32sc\_C4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image addition.*

- **NppStatus nppiAdd\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image addition.*

- **NppStatus nppiAdd\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel image addition.*

- **NppStatus nppiAdd\_32f\_C3IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image addition.*

- **NppStatus nppiAdd\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image addition.*

- **NppStatus nppiAdd\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image addition.*

- **NppStatus nppiAdd\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image addition.*
- **NppStatus nppiAdd\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image addition.*
- **NppStatus nppiAdd\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- **NppStatus nppiAdd\_32fc\_C1IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*
- **NppStatus nppiAdd\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- **NppStatus nppiAdd\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*
- **NppStatus nppiAdd\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.*
- **NppStatus nppiAdd\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.*
- **NppStatus nppiAdd\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- **NppStatus nppiAdd\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*

## Add Square Image

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

- **NppStatus nppiAddSquare\_8u32f\_C1IMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddSquare\_8u32f\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image squared then added to in place floating point destination image.*

- **NppStatus** **nppiAddSquare\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddSquare\_16u32f\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image squared then added to in place floating point destination image.*

- **NppStatus** **nppiAddSquare\_32f\_C1IMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddSquare\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image squared then added to in place floating point destination image.*

## Add Product Image

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

- **NppStatus** **nppiAddProduct\_8u32f\_C1IMR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddProduct\_8u32f\_C1IR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image product added to in place floating point destination image.*

- **NppStatus** **nppiAddProduct\_16u32f\_C1IMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddProduct\_16u32f\_C1IR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image product added to in place floating point destination image.*

- **NppStatus** **nppiAddProduct\_32f\_C1IMR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddProduct\_32f\_C1IR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image product added to in place floating point destination image.*

## Add Weighted Image

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

- **NppStatus** **nppiAddWeighted\_8u32f\_C1IMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddWeighted\_8u32f\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.*

- **NppStatus** **nppiAddWeighted\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddWeighted\_16u32f\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.*

- **NppStatus** **nppiAddWeighted\_32f\_C1IMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*

- **NppStatus** **nppiAddWeighted\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

*One 32-bit floating point channel alpha weighted image added to in place floating point destination image.*

## Mul Image

Pixel by pixel multiply of two images.

- **NppStatus** **nppiMul\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiMul\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiMul\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16sc\_C4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- `NppStatus nppiMul_32s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C3RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_AC4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_AC4RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C4RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32sc\_C4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image multiplication.*
- **NppStatus nppiMul\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*
- **NppStatus nppiMul\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image multiplication.*
- **NppStatus nppiMul\_32f\_C3IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*

- **NppStatus nppiMul\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image multiplication.*
- **NppStatus nppiMul\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image multiplication.*
- **NppStatus nppiMul\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image multiplication.*
- **NppStatus nppiMul\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image multiplication.*
- **NppStatus nppiMul\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- **NppStatus nppiMul\_32fc\_C1IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- **NppStatus nppiMul\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- **NppStatus nppiMul\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.*
- **NppStatus nppiMul\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- **NppStatus nppiMul\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*

## MulScale Image

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

- **NppStatus nppiMulScale\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C3IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_16u\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C3IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

## Sub Image

Pixel by pixel subtraction of two images.

- **NppStatus nppiSub\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_C1IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_C3IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSub_16u_AC4IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiSub\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiSub\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_C4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16sc\_C4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*
- **NppStatus nppiSub\_32s\_C1IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_C3IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_AC4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_AC4IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32s\_C4IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiSub\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_C4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image subtraction.*
- **NppStatus nppiSub\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C3IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image subtraction.*
- **NppStatus nppiSub\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image subtraction.*
- **NppStatus nppiSub\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image subtraction.*
- **NppStatus nppiSub\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel in place image subtraction.*

- `NppStatus nppiSub_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*

- `NppStatus nppiSub_32fc_C1IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*

- `NppStatus nppiSub_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*

- `NppStatus nppiSub_32fc_C3IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*

- `NppStatus nppiSub_32fc_AC4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.*

- `NppStatus nppiSub_32fc_AC4IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.*

- `NppStatus nppiSub_32fc_C4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*

- `NppStatus nppiSub_32fc_C4IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*

## Div Image

Pixel by pixel division of two images.

- `NppStatus nppiDiv_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_C1IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_16u_C1IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_16u_C3IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiDiv\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- `NppStatus nppiDiv_32s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_AC4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_AC4IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C4RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C4IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)



*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32sc\_C4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32sc\_C4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image division.*

- **NppStatus** **nppiDiv\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image division.*

- **NppStatus** **nppiDiv\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel image division.*

- **NppStatus** **nppiDiv\_32f\_C3IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image division.*

- **NppStatus** **nppiDiv\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image division.*

- **NppStatus** **nppiDiv\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image division.*

- **NppStatus** **nppiDiv\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image division.*
- **NppStatus** **nppiDiv\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image division.*
- **NppStatus** **nppiDiv\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- **NppStatus** **nppiDiv\_32fc\_C1IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- **NppStatus** **nppiDiv\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- **NppStatus** **nppiDiv\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- **NppStatus** **nppiDiv\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.*
- **NppStatus** **nppiDiv\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.*
- **NppStatus** **nppiDiv\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- **NppStatus** **nppiDiv\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*

## Div\_Round Image

Pixel by pixel division of two images using result rounding modes.

- **NppStatus** **nppiDiv\_Round\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiDiv\_Round\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

## Abs Image

Absolute value of each pixel value in an image.

- **NppStatus** **nppiAbs\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 16-bit signed short channel image absolute value.*

- **NppStatus** **nppiAbs\_16s\_C1IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit signed short channel in place image absolute value.*

- **NppStatus** **nppiAbs\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 16-bit signed short channel image absolute value.*

- **NppStatus** **nppiAbs\_16s\_C3IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 16-bit signed short channel in place image absolute value.*

- **NppStatus** **nppiAbs\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit signed short channel image absolute value with unmodified alpha.*

- **NppStatus** **nppiAbs\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit signed short channel in place image absolute value with unmodified alpha.*

- **NppStatus** **nppiAbs\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit signed short channel image absolute value.*

- **NppStatus nppiAbs\_16s\_C4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit signed short channel in place image absolute value.*
- **NppStatus nppiAbs\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image absolute value.*
- **NppStatus nppiAbs\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image absolute value.*
- **NppStatus nppiAbs\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image absolute value.*
- **NppStatus nppiAbs\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image absolute value.*
- **NppStatus nppiAbs\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image absolute value with unmodified alpha.*
- **NppStatus nppiAbs\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value with unmodified alpha.*
- **NppStatus nppiAbs\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image absolute value.*
- **NppStatus nppiAbs\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value.*

## AbsDiff Image

Pixel by pixel absolute difference between two images.

- **NppStatus nppiAbsDiff\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel absolute difference of image1 minus image2.*

## Sqr Image

Square each pixel in an image.

- **NppStatus nppiSqr\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiSqr\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16s\_AC4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_AC4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image squared.*

- **NppStatus nppiSqr\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image squared.*

- **NppStatus nppiSqr\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel image squared.*

- **NppStatus nppiSqr\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel in place image squared.*

- **NppStatus nppiSqr\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel image squared with unmodified alpha.*

- **NppStatus nppiSqr\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel in place image squared with unmodified alpha.*

- **NppStatus nppiSqr\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel image squared.*

- **NppStatus nppiSqr\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel in place image squared.*

## Sqrt Image

Pixel by pixel square root of each pixel in an image.

- **NppStatus** **nppiSqrt\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_C4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_AC4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16s\_AC4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image square root.*
- **NppStatus nppiSqrt\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image square root.*
- **NppStatus nppiSqrt\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image square root.*
- **NppStatus nppiSqrt\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image square root.*
- **NppStatus nppiSqrt\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image square root with unmodified alpha.*
- **NppStatus nppiSqrt\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image square root with unmodified alpha.*
- **NppStatus nppiSqrt\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image square root.*
- **NppStatus nppiSqrt\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image square root.*

## Ln Image

Pixel by pixel natural logarithm of each pixel in an image.

- **NppStatus nppiLn\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_8u\_C1RSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiLn\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiLn\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image natural logarithm.*
- **NppStatus** **nppiLn\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image natural logarithm.*
- **NppStatus** **nppiLn\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image natural logarithm.*
- **NppStatus** **nppiLn\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image natural logarithm.*

## Exp Image

Exponential value of each pixel in an image.

- **NppStatus** **nppiExp\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiExp\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image exponential.*
- **NppStatus nppiExp\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image exponential.*

## AndC Image

Pixel by pixel logical and of an image with a constant.

- **NppStatus nppiAndC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C3IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.*
- **NppStatus nppiAndC\_8u\_AC4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.*
- **NppStatus nppiAndC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C3IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical and with constant.*



- `NppStatus nppiAndC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_AC4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and with constant.*
- `NppStatus nppiAndC_16u_C4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C3IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_AC4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with constant.*

## OrC Image

Pixel by pixel logical or of an image with a constant.

- **NppStatus nppiOrC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_AC4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C3IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical or with constant.*

- `NppStatus nppiOrC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_AC4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant.*
- `NppStatus nppiOrC_16u_C4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C3IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_AC4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant.*

## XorC Image

Pixel by pixel logical exclusive or of an image with a constant.

- **NppStatus nppiXorC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_AC4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C4IR** (const **Npp8u** \*pConstants, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pConstants, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C3IR** (const **Npp16u** \*pConstants, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical exclusive or with constant.*

- `NppStatus nppiXorC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_16u_AC4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_16u_C4IR` (const `Npp16u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C3IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_32s_AC4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C4IR` (const `Npp32s` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant.*

## RShiftC Image

Pixel by pixel right shift of an image by a constant value.

- `NppStatus nppiRShiftC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C1IR` (const `Npp32u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C3IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8u_AC4IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C4IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C1R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit signed char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C1IR` (const `Npp32u` nConstant, `Npp8s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit signed char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C3R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit signed char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C3IR` (const `Npp32u` \*pConstants, `Npp8s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit signed char channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_8s_AC4R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit signed char channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8s_AC4IR` (const `Npp32u` \*pConstants, `Npp8s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8s_C4R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit signed char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C4IR` (const `Npp32u` \*pConstants, `Npp8s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16u_C1IR` (const `Npp32u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16u_C3IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16u_AC4IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16u_C4IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)



*One 16-bit signed short channel image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C1IR` (const `Npp32u` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit signed short channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C3R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit signed short channel image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C3IR` (const `Npp32u` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit signed short channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_16s_AC4R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit signed short channel image right shift by constant with unmodified alpha.*

- `NppStatus nppiRShiftC_16s_AC4IR` (const `Npp32u` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.*

- `NppStatus nppiRShiftC_16s_C4R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit signed short channel image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C4IR` (const `Npp32u` \*pConstants, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit signed short channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel image right shift by constant.*

- `NppStatus nppiRShiftC_32s_C1IR` (const `Npp32u` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel image right shift by constant.*

- `NppStatus nppiRShiftC_32s_C3IR` (const `Npp32u` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image right shift by constant with unmodified alpha.*



- `NppStatus nppiRShiftC_32s_AC4IR` (const `Npp32u` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C4IR` (const `Npp32u` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image right shift by constant.*

## LShiftC Image

Pixel by pixel left shift of an image by a constant value.

- `NppStatus nppiLShiftC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image left shift by constant.*
- `NppStatus nppiLShiftC_8u_C1IR` (const `Npp32u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image left shift by constant.*
- `NppStatus nppiLShiftC_8u_C3IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_8u_AC4IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant.*
- `NppStatus nppiLShiftC_8u_C4IR` (const `Npp32u` \*pConstants, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C1IR` (const `Npp32u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C3IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image left shift by constant.*

- `NppStatus nppiLShiftC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.*

- `NppStatus nppiLShiftC_16u_AC4IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.*

- `NppStatus nppiLShiftC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C4IR` (const `Npp32u` \*pConstants, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image left shift by constant.*

- `NppStatus nppiLShiftC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel image left shift by constant.*

- `NppStatus nppiLShiftC_32s_C1IR` (const `Npp32u` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel in place image left shift by constant.*

- `NppStatus nppiLShiftC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel image left shift by constant.*

- `NppStatus nppiLShiftC_32s_C3IR` (const `Npp32u` \*pConstants, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel in place image left shift by constant.*

- `NppStatus nppiLShiftC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` \*pConstants, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image left shift by constant with unmodified alpha.*

- **NppStatus nppiLShiftC\_32s\_AC4IR** (const **Npp32u** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.*
- **NppStatus nppiLShiftC\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32u** \*pConstants, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image left shift by constant.*
- **NppStatus nppiLShiftC\_32s\_C4IR** (const **Npp32u** \*pConstants, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant.*

## And Image

Pixel by pixel logical and of images.

- **NppStatus nppiAnd\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical and.*
- **NppStatus nppiAnd\_8u\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical and.*
- **NppStatus nppiAnd\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical and.*
- **NppStatus nppiAnd\_8u\_C3IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and.*
- **NppStatus nppiAnd\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and.*
- **NppStatus nppiAnd\_8u\_C4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and.*
- **NppStatus nppiAnd\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image logical and.*

- `NppStatus nppiAnd_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image logical and.*

- `NppStatus nppiAnd_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image logical and.*

- `NppStatus nppiAnd_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image logical and.*

- `NppStatus nppiAnd_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical and with unmodified alpha.*

- `NppStatus nppiAnd_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical and with unmodified alpha.*

- `NppStatus nppiAnd_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical and.*

- `NppStatus nppiAnd_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical and.*

- `NppStatus nppiAnd_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel image logical and.*

- `NppStatus nppiAnd_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel in place image logical and.*

- `NppStatus nppiAnd_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel image logical and.*

- `NppStatus nppiAnd_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel in place image logical and.*

- `NppStatus nppiAnd_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image logical and with unmodified alpha.*

- `NppStatus nppiAnd_32s_AC4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel in place image logical and with unmodified alpha.*

- `NppStatus nppiAnd_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image logical and.*

- `NppStatus nppiAnd_32s_C4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel in place image logical and.*

## Or Image

Pixel by pixel logical or of images.

- `NppStatus nppiOr_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel image logical or.*

- `NppStatus nppiOr_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel in place image logical or.*

- `NppStatus nppiOr_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel image logical or.*

- `NppStatus nppiOr_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel in place image logical or.*

- `NppStatus nppiOr_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image logical or with unmodified alpha.*

- `NppStatus nppiOr_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image logical or with unmodified alpha.*

- `NppStatus nppiOr_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image logical or.*

- `NppStatus nppiOr_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image logical or.*

- `NppStatus nppiOr_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel image logical or.*

- `NppStatus nppiOr_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image logical or.*

- `NppStatus nppiOr_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image logical or.*

- `NppStatus nppiOr_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image logical or.*

- `NppStatus nppiOr_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical or with unmodified alpha.*

- `NppStatus nppiOr_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical or with unmodified alpha.*

- `NppStatus nppiOr_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical or.*

- `NppStatus nppiOr_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical or.*

- `NppStatus nppiOr_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel image logical or.*

- `NppStatus nppiOr_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel in place image logical or.*

- `NppStatus nppiOr_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel image logical or.*

- `NppStatus nppiOr_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel in place image logical or.*

- `NppStatus nppiOr_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image logical or with unmodified alpha.*

- `NppStatus nppiOr_32s_AC4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel in place image logical or with unmodified alpha.*

- `NppStatus nppiOr_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image logical or.*

- `NppStatus nppiOr_32s_C4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel in place image logical or.*

## Xor Image

Pixel by pixel logical exclusive or of images.

- `NppStatus nppiXor_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel image logical exclusive or.*

- `NppStatus nppiXor_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel in place image logical exclusive or.*

- `NppStatus nppiXor_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel image logical exclusive or.*

- `NppStatus nppiXor_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel in place image logical exclusive or.*

- `NppStatus nppiXor_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.*

- `NppStatus nppiXor_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.*

- `NppStatus nppiXor_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image logical exclusive or.*

- `NppStatus nppiXor_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image logical exclusive or.*

- `NppStatus nppiXor_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)



*One 16-bit unsigned short channel image logical exclusive or.*

- `NppStatus nppiXor_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image logical exclusive or.*

- `NppStatus nppiXor_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image logical exclusive or.*

- `NppStatus nppiXor_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image logical exclusive or.*

- `NppStatus nppiXor_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.*

- `NppStatus nppiXor_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.*

- `NppStatus nppiXor_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image logical exclusive or.*

- `NppStatus nppiXor_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image logical exclusive or.*

- `NppStatus nppiXor_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel image logical exclusive or.*

- `NppStatus nppiXor_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit signed integer channel in place image logical exclusive or.*

- `NppStatus nppiXor_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel image logical exclusive or.*

- `NppStatus nppiXor_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit signed integer channel in place image logical exclusive or.*

- `NppStatus nppiXor_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.*



- **NppStatus nppiXor\_32s\_AC4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.*

- **NppStatus nppiXor\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel image logical exclusive or.*

- **NppStatus nppiXor\_32s\_C4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel in place image logical exclusive or.*

## Not Image

Pixel by pixel logical not of image.

- **NppStatus nppiNot\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image logical not.*

- **NppStatus nppiNot\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel in place image logical not.*

- **NppStatus nppiNot\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 8-bit unsigned char channel image logical not.*

- **NppStatus nppiNot\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 8-bit unsigned char channel in place image logical not.*

- **NppStatus nppiNot\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel image logical not with unmodified alpha.*

- **NppStatus nppiNot\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel in place image logical not with unmodified alpha.*

- **NppStatus nppiNot\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel image logical not.*

- **NppStatus nppiNot\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel in place image logical not.*

## AlphaCompC Image

Composite two images using constant alpha values.

- `NppStatus nppiAlphaCompC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 8-bit unsigned char channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Three 8-bit unsigned char channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 8-bit unsigned char channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 8-bit unsigned char channel image composition with alpha using constant source alpha.*

- `NppStatus nppiAlphaCompC_8s_C1R` (const `Npp8s` \*pSrc1, int nSrc1Step, `Npp8s` nAlpha1, const `Npp8s` \*pSrc2, int nSrc2Step, `Npp8s` nAlpha2, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 8-bit signed char channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 16-bit unsigned short channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Three 16-bit unsigned short channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 16-bit unsigned short channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 16-bit unsigned short channel image composition with alpha using constant source alpha.*

- `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, `Npp16s` nAlpha1, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` nAlpha2, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 16-bit signed short channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32u_C1R` (const `Npp32u` \*pSrc1, int nSrc1Step, `Npp32u` nAlpha1, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` nAlpha2, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit unsigned integer channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, `Npp32s` nAlpha1, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` nAlpha2, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit signed integer channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, `Npp32f` nAlpha1, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` nAlpha2, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit floating point channel image composition using constant alpha.*

## AlphaPremulC Image

Premultiplies pixels of an image using a constant alpha value.

- `NppStatus nppiAlphaPremulC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_C1IR` (`Npp8u` nAlpha1, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 8-bit unsigned char channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_C3IR` (`Npp8u` nAlpha1, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 8-bit unsigned char channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_C4IR` (`Npp8u` nAlpha1, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.*

- `NppStatus nppiAlphaPremulC_8u_AC4IR` (`Npp8u` nAlpha1, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C1IR` (`Npp16u` nAlpha1, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C3IR` (`Npp16u` nAlpha1, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C4IR` (`Npp16u` nAlpha1, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_AC4IR` (`Npp16u` nAlpha1, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.*

## AlphaComp Image

Composite two images using alpha opacity values contained in each image.

- `NppStatus nppiAlphaComp_8u_AC1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_8s_AC1R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_16u_AC1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_16s_AC1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_32u_AC1R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_32u_AC4R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_32s_AC1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*

- `NppStatus nppiAlphaComp_32f_AC1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*

- `NppStatus nppiAlphaComp_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` nppAlphaOp)

*Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*

## AlphaPremul Image

Premultiplies image pixels by image alpha opacity values.

- **NppStatus nppiAlphaPremul\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).*
- **NppStatus nppiAlphaPremul\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*
- **NppStatus nppiAlphaPremul\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).*
- **NppStatus nppiAlphaPremul\_16u\_AC4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*

## 7.7.1 Function Documentation

### 7.7.1.1 NppStatus nppiAbs\_16s\_AC4IR (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel in place image absolute value with unmodified alpha.

#### Parameters:

**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.2 NppStatus nppiAbs\_16s\_AC4R (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel image absolute value with unmodified alpha.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.3 NppStatus nppiAbs\_16s\_C1IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.4 NppStatus nppiAbs\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.5 NppStatus nppiAbs\_16s\_C3IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.6 NppStatus nppiAbs\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.7 NppStatus nppiAbs\_16s\_C4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.8 NppStatus nppiAbs\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.9 NppStatus nppiAbs\_32f\_AC4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image absolute value with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.10 NppStatus nppiAbs\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image absolute value with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.11 NppStatus nppiAbs\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.12 NppStatus nppiAbs\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.13 NppStatus nppiAbs\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.14 NppStatus nppiAbs\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.15 NppStatus nppiAbs\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.16 NppStatus nppiAbs\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.17 NppStatus nppiAbsDiff\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.18 **NppStatus nppiAbsDiff\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel absolute difference of image1 minus image2.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.19 **NppStatus nppiAbsDiff\_8u\_C1R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel absolute difference of image1 minus image2.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.20 **NppStatus nppiAbsDiffC\_16u\_C1R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32u *nConstant*)

One 16-bit unsigned short channel image absolute difference with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.21 NppStatus nppiAbsDiffC\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, Npp32f nConstant)**

One 32-bit floating point channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.22 NppStatus nppiAbsDiffC\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nConstant)**

One 8-bit unsigned char channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.23 NppStatus nppiAdd\_16s\_AC4IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.24 NppStatus nppiAdd\_16s\_AC4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.25 NppStatus nppiAdd\_16s\_C1IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.26 **NppStatus nppiAdd\_16s\_C1RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.27 **NppStatus nppiAdd\_16s\_C3IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.28 NppStatus nppiAdd\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.29 NppStatus nppiAdd\_16s\_C4IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.30 NppStatus nppiAdd\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.



*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.31 **NppStatus nppiAdd\_16sc\_AC4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.32 **NppStatus nppiAdd\_16sc\_AC4RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.33 **NppStatus nppiAdd\_16sc\_C1RSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.34 **NppStatus nppiAdd\_16sc\_C1RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.35 **NppStatus nppiAdd\_16sc\_C3IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.36 **NppStatus nppiAdd\_16sc\_C3RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.37 **NppStatus nppiAdd\_16sc\_C4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.38 NppStatus nppiAdd\_16sc\_C4RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.39 NppStatus nppiAdd\_16u\_AC4IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.40 NppStatus nppiAdd\_16u\_AC4RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.41 NppStatus nppiAdd\_16u\_C1IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.42 NppStatus nppiAdd\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.43 NppStatus npAdd\_16u\_C3IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.44 NppStatus npAdd\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.45 NppStatus nppiAdd\_16u\_C4IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.46 NppStatus nppiAdd\_16u\_C4RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.47 NppStatus nppiAdd\_32f\_AC4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha in place image addition.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.48 NppStatus nppiAdd\_32f\_AC4R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel with unmodified alpha image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.49 NppStatus nppiAdd\_32f\_C11R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.50 NppStatus nppiAdd\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.51 NppStatus nppiAdd\_32f\_C3IR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.52 NppStatus nppiAdd\_32f\_C3R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.53 NppStatus nppiAdd\_32f\_C4IR (const Npp32f \*pSrc, int nSrcStep, Npp32f \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.54 NppStatus nppiAdd\_32f\_C4R (const Npp32f \*pSrc1, int nSrc1Step, const Npp32f \*pSrc2, int nSrc2Step, Npp32f \*pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.55 NppStatus nppiAdd\_32fc\_AC4IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.56 NppStatus nppiAdd\_32fc\_AC4R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.57 NppStatus nppiAdd\_32fc\_C1IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.58** `NppStatus nppiAdd_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.59** `NppStatus nppiAdd_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.60** `NppStatus nppiAdd_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.61 **NppStatus nppiAdd\_32fc\_C4IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.62 **NppStatus nppiAdd\_32fc\_C4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pSrc2, int nSrc2Step, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.63 NppStatus npAdd\_32s\_AC4IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 32-bit signed integer channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.64 NppStatus npAdd\_32s\_AC4RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 32-bit signed integer channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.65 NppStatus npAdd\_32s\_C1IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.66 NppStatus npAdd\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image add. Add the pixel values of corresponding pixels in the ROI and write them to the output image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.67 NppStatus npAdd\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.68 NppStatus nppiAdd\_32s\_C3IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.69 NppStatus nppiAdd\_32s\_C3RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.70 NppStatus nppiAdd\_32s\_C4IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.71** `NppStatus nppiAdd_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.72** `NppStatus nppiAdd_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.73 NppStatus nppiAdd\_32sc\_AC4RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.74 NppStatus nppiAdd\_32sc\_C1IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.75 NppStatus nppiAdd\_32sc\_C1RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.76 **NppStatus nppiAdd\_32sc\_C3IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.77 **NppStatus nppiAdd\_32sc\_C3RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.78 NppStatus nppiAdd\_32sc\_C4IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.79 NppStatus nppiAdd\_32sc\_C4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.80 NppStatus nppiAdd\_8u\_AC4IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.81 NppStatus nppiAdd\_8u\_AC4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.82 NppStatus nppiAdd\_8u\_C1IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.83** `NppStatus nppiAdd_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.84** `NppStatus nppiAdd_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.85** `NppStatus nppiAdd_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.86** `NppStatus nppiAdd_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.87** `NppStatus nppiAdd_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.88 NppStatus nppiAddC\_16s\_AC4IRSfs (const Npp16s \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.89 NppStatus nppiAddC\_16s\_AC4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



#### 7.7.1.90 **NppStatus nppiAddC\_16s\_C1IRSfs** (const Npp16s *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.91 **NppStatus nppiAddC\_16s\_C1RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.92 **NppStatus nppiAddC\_16s\_C3IRSfs** (const Npp16s \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.93 **NppStatus nppiAddC\_16s\_C3RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.94 **NppStatus nppiAddC\_16s\_C4IRSfs** (const Npp16s \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.95 **NppStatus nppiAddC\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.96 NppStatus nppiAddC\_16sc\_AC4IRSfs (const Npp16sc \* pConstants, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.97 NppStatus nppiAddC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pConstants, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.98 NppStatus nppiAddC\_16sc\_C1IRSfs (const Npp16sc *nConstant*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.99 NppStatus nppiAddC\_16sc\_C1RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc *nConstant*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.100 NppStatus nppiAddC\_16sc\_C3IRSfs (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.101** `NppStatus npAddC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.102** `NppStatus npAddC_16sc_C4IRSfs (const Npp16sc * pConstants, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.103** `NppStatus nppiAddC_16sc_C4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.104** `NppStatus nppiAddC_16u_AC4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.105** `NppStatus nppiAddC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.106 NppStatus npAddC\_16u\_C1IRSfs (const Npp16u nConstant, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.107 NppStatus npAddC\_16u\_C1RSfs (const Npp16u \* pSrcI, int nSrcIStep, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.108** `NppStatus nppiAddC_16u_C3IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.109** `NppStatus nppiAddC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.110** `NppStatus nppiAddC_16u_C4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.111** `NppStatus nppiAddC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.112** `NppStatus nppiAddC_32f_AC4IR (const Npp32f * pConstants, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image add constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.113** `NppStatus nppiAddC_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.114 **NppStatus nppiAddC\_32f\_C1IR** (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image add constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.115 **NppStatus nppiAddC\_32f\_C1R** (const Npp32f \* *pSrcI*, int *nSrcIStep*, const Npp32f *nConstant*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image add constant.

##### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.116 **NppStatus nppiAddC\_32f\_C3IR** (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image add constant.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.117 NppStatus nppiAddC\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pConstants, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.118 NppStatus nppiAddC\_32f\_C4IR (const Npp32f \* pConstants, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image add constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.119 NppStatus nppiAddC\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pConstants, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.120 `NppStatus nppiAddC_32fc_AC4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.121 `NppStatus nppiAddC_32fc_AC4R (const Npp32fc * pSrcI, int nSrcIStep, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.122 `NppStatus nppiAddC_32fc_C1IR (const Npp32fc nConstant, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*nConstant* Constant.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.123 NppStatus nppiAddC\_32fc\_C1R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *nConstant*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.124 NppStatus nppiAddC\_32fc\_C3IR (const Npp32fc \* *pConstants*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.125 NppStatus nppiAddC\_32fc\_C3R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pConstants*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.126** `NppStatus nppiAddC_32fc_C4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.127** `NppStatus nppiAddC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.128** `NppStatus nppiAddC_32s_AC4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.129** `NppStatus npaiAddC_32s_AC4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.130** `NppStatus npaiAddC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.131 NppStatus nppiAddC\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.132 NppStatus nppiAddC\_32s\_C3IRSfs (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* [pointer to a list of constant values, one per channel](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.133 NppStatus nppiAddC\_32s\_C3RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* [pointer to a list of constant values, one per channel](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).



*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.134 NppStatus npAddC\_32s\_C4IRSfs (const Npp32s \* pConstants, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.135 NppStatus npAddC\_32s\_C4RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pConstants, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.136 **NppStatus nppiAddC\_32sc\_AC4IRSfs** (const Npp32sc \* *pConstants*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.137 **NppStatus nppiAddC\_32sc\_AC4RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pConstants*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.138 **NppStatus nppiAddC\_32sc\_C1IRSfs** (const Npp32sc *nConstant*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.139** `NppStatus nppiAddC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.140** `NppStatus nppiAddC_32sc_C3IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.141** `NppStatus nppiAddC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.142** `NppStatus nppiAddC_32sc_C4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.143** `NppStatus nppiAddC_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.144 NppStatus nppiAddC\_8u\_AC4IRSfs (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel..

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.145 NppStatus nppiAddC\_8u\_AC4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel..

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.146 **NppStatus nppiAddC\_8u\_C1RSfs** (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.147 **NppStatus nppiAddC\_8u\_C1RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.148 **NppStatus nppiAddC\_8u\_C3RSfs** (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel..  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.149** `NppStatus nppiAddC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel..  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.150** `NppStatus nppiAddC_8u_C4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.151** `NppStatus nppiAddC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel..  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.152** `NppStatus nppiAddProduct_16u32f_C1IMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.153** `NppStatus nppiAddProduct_16u32f_C1IR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.154** `NppStatus nppiAddProduct_32f_C1IMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.155** `NppStatus nppiAddProduct_32f_C1IR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.156** `NppStatus nppiAddProduct_8u32f_C1IMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.157** `NppStatus nppiAddProduct_8u32f_C1IR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.158** `NppStatus nppiAddSquare_16u32f_C1IMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.159** `NppStatus nppiAddSquare_16u32f_C1IR (const Npp16u * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.160** `NppStatus nppiAddSquare_32f_C1IMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.161** `NppStatus nppiAddSquare_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.162** `NppStatus nppiAddSquare_8u32f_C1IMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.163** `NppStatus nppiAddSquare_8u32f_C1IR (const Npp8u * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.164** `NppStatus nppiAddWeighted_16u32f_C1IMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.165** `NppStatus nppiAddWeighted_16u32f_C1IR (const Npp16u * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.166** `NppStatus nppiAddWeighted_32f_C1IMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.167** `NppStatus nppiAddWeighted_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.168** `NppStatus nppiAddWeighted_8u32f_C1IMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.169** `NppStatus nppiAddWeighted_8u32f_C1IR (const Npp8u * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)`

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.170** `NppStatus nppiAlphaComp_16s_AC1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.171** `NppStatus nppiAlphaComp_16u_AC1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.172** `NppStatus nppiAlphaComp_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.173** `NppStatus nppiAlphaComp_32f_AC1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.174** `NppStatus nppiAlphaComp_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.175** `NppStatus nppiAlphaComp_32s_AC1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.176** `NppStatus nppiAlphaComp_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.177** `NppStatus nppiAlphaComp_32u_AC1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.178** `NppStatus nppiAlphaComp_32u_AC4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.179** `NppStatus nppiAlphaComp_8s_AC1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.180** `NppStatus nppiAlphaComp_8u_AC1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.181** `NppStatus nppiAlphaComp_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.182** `NppStatus nppiAlphaCompC_16s_C1R (const Npp16s * pSrc1, int nSrc1Step, Npp16s nAlpha1, const Npp16s * pSrc2, int nSrc2Step, Npp16s nAlpha2, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 16-bit signed short channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.183** `NppStatus nppiAlphaCompC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u * pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.184** `NppStatus nppiAlphaCompC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u * pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.

*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nppAlphaOp* [alpha\\_compositing\\_operation](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.185** `NppStatus nppiAlphaCompC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u * pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Three 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nppAlphaOp* [alpha\\_compositing\\_operation](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.186** `NppStatus nppiAlphaCompC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u * pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*nAlpha2* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nppAlphaOp* alpha\_compositing\_operation.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.187** `NppStatus nppiAlphaCompC_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, Npp32f nAlpha1, const Npp32f * pSrc2, int nSrc2Step, Npp32f nAlpha2, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit floating point channel image composition using constant alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0.0 - 1.0).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*nAlpha2* Image alpha opacity (0.0 - 1.0).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nppAlphaOp* alpha\_compositing\_operation.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.188** `NppStatus nppiAlphaCompC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, Npp32s nAlpha1, const Npp32s * pSrc2, int nSrc2Step, Npp32s nAlpha2, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit signed integer channel image composition using constant alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*nAlpha2* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.189** `NppStatus nppiAlphaCompC_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, Npp32u nAlpha1, const Npp32u * pSrc2, int nSrc2Step, Npp32u nAlpha2, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 32-bit unsigned integer channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.190** `NppStatus nppiAlphaCompC_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, Npp8s nAlpha1, const Npp8s * pSrc2, int nSrc2Step, Npp8s nAlpha2, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 8-bit signed char channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.191** `NppStatus nppiAlphaCompC_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u *pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.192** `NppStatus nppiAlphaCompC_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u *pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

One 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nppAlphaOp* [alpha\\_compositing\\_operation](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.193** `NppStatus nppiAlphaCompC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Three 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*nAlpha2* Image alpha opacity (0 - max channel pixel value).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nppAlphaOp* [alpha\\_compositing\\_operation](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.194** `NppStatus nppiAlphaCompC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp nppAlphaOp)`

Four 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*nAlpha2* Image alpha opacity (0 - max channel pixel value).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nppAlphaOp* alpha\_compositing\_operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.195 NppStatus nppiAlphaPremul\_16u\_AC4IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.196 NppStatus nppiAlphaPremul\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.197 NppStatus nppiAlphaPremul\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.198 NppStatus nppiAlphaPremul\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.199 NppStatus nppiAlphaPremulC\_16u\_AC4IR (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

#### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.200 NppStatus nppiAlphaPremulC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.201 NppStatus nppiAlphaPremulC\_16u\_C1IR (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.202 NppStatus nppiAlphaPremulC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel image premultiplication using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.203 NppStatus nppiAlphaPremulC\_16u\_C3IR (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.204 **NppStatus nppiAlphaPremulC\_16u\_C3R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.205 **NppStatus nppiAlphaPremulC\_16u\_C4IR** (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.206 **NppStatus nppiAlphaPremulC\_16u\_C4R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.207 NppStatus nppiAlphaPremulC\_8u\_AC4IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.208 NppStatus nppiAlphaPremulC\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.209 NppStatus nppiAlphaPremulC\_8u\_C1IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.210 **NppStatus nppiAlphaPremulC\_8u\_C1R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.211 **NppStatus nppiAlphaPremulC\_8u\_C3IR** (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.212 **NppStatus nppiAlphaPremulC\_8u\_C3R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



#### 7.7.1.213 `NppStatus nppiAlphaPremulC_8u_C4IR (Npp8u nAlpha1, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.214 `NppStatus nppiAlphaPremulC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.215 `NppStatus nppiAnd_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with unmodified alpha.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.216** `NppStatus nppiAnd_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.217** `NppStatus nppiAnd_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.218** `NppStatus nppiAnd_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.219** `NppStatus nppiAnd_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.220** `NppStatus nppiAnd_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.221** `NppStatus nppiAnd_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.222** `NppStatus nppiAnd_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.223** `NppStatus nppiAnd_32s_AC4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.224** `NppStatus nppiAnd_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.225** `NppStatus nppiAnd_32s_C1IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.226** `NppStatus nppiAnd_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.227** `NppStatus nppiAnd_32s_C3IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.228** `NppStatus nppiAnd_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.229 NppStatus nppiAnd\_32s\_C4IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.230 NppStatus nppiAnd\_32s\_C4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.231 NppStatus nppiAnd\_8u\_AC4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.232** `NppStatus nppiAnd_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.233** `NppStatus nppiAnd_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.234** `NppStatus nppiAnd_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.235** `NppStatus nppiAnd_8u_C3IR (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.236** `NppStatus nppiAnd_8u_C3R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.237 **NppStatus nppiAnd\_8u\_C4IR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical and.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.238 **NppStatus nppiAnd\_8u\_C4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical and.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.239 **NppStatus nppiAndC\_16u\_AC4IR** (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.240 NppStatus nppiAndC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.241 NppStatus nppiAndC\_16u\_C1IR (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.242 NppStatus nppiAndC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.243 NppStatus nppiAndC\_16u\_C3IR (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.244 NppStatus nppiAndC\_16u\_C3R (const Npp16u \* *pSrcI*, int *nSrcIStep*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.245 NppStatus nppiAndC\_16u\_C4IR (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.246** `NppStatus nppiAndC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.247** `NppStatus nppiAndC_32s_AC4IR (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.248** `NppStatus nppiAndC_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical and with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.249 NppStatus nppiAndC\_32s\_C1IR (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical and with constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.250 NppStatus nppiAndC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical and with constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.251 NppStatus nppiAndC\_32s\_C3IR (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical and with constant.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.252 NppStatus nppiAndC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.253 NppStatus nppiAndC\_32s\_C4IR (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical and with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.254 NppStatus nppiAndC\_32s\_C4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.255 `NppStatus nppiAndC_8u_AC4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.256 `NppStatus nppiAndC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.257 `NppStatus nppiAndC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical and with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.258 NppStatus nppiAndC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.259 NppStatus nppiAndC\_8u\_C3IR (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel in place image logical and with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.260 NppStatus nppiAndC\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.261 `NppStatus nppiAndC_8u_C4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and with constant.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.262 `NppStatus nppiAndC_8u_C4R (const Npp8u * pSrcI, int nSrcIStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with constant.

##### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.263 `NppStatus nppiDiv_16s_AC4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.264** `NppStatus nppiDiv_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.265** `NppStatus nppiDiv_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.266** `NppStatus nppiDiv_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.267** `NppStatus nppiDiv_16s_C3IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.268** `NppStatus nppiDiv_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.269 NppStatus nppiDiv\_16s\_C4IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.270 NppStatus nppiDiv\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.271 NppStatus nppiDiv\_16sc\_AC4IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.272** `NppStatus nppiDiv_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.273** `NppStatus nppiDiv_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.274** `NppStatus nppiDiv_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.275** `NppStatus nppiDiv_16sc_C3RSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.276** `NppStatus nppiDiv_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.277** `NppStatus nppiDiv_16sc_C4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.278** `NppStatus nppiDiv_16sc_C4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.279 NppStatus nppiDiv\_16u\_AC4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.280 NppStatus nppiDiv\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.281** `NppStatus nppiDiv_16u_C1IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.282** `NppStatus nppiDiv_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.283** `NppStatus nppiDiv_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.284** `NppStatus npDiv_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.285** `NppStatus npDiv_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.286** `NppStatus nppiDiv_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.287** `NppStatus nppiDiv_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.288** `NppStatus nppiDiv_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.289 NppStatus nppiDiv\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.290 NppStatus nppiDiv\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.291 NppStatus nppiDiv\_32f\_C3IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.292 NppStatus nppiDiv\_32f\_C3R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.293 NppStatus nppiDiv\_32f\_C4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.294** `NppStatus nppiDiv_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.295** `NppStatus nppiDiv_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.296** `NppStatus nppiDiv_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.297 NppStatus nppiDiv\_32fc\_C1IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.298 NppStatus nppiDiv\_32fc\_C1R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pSrc2, int nSrc2Step, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.299 NppStatus nppiDiv\_32fc\_C3IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.300 NppStatus nppiDiv\_32fc\_C3R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.301 NppStatus nppiDiv\_32fc\_C4IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.302** `NppStatus nppiDiv_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.303** `NppStatus nppiDiv_32s_AC4IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.304** `NppStatus nppiDiv_32s_AC4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.305 NppStatus nppiDiv\_32s\_C1IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.306 NppStatus nppiDiv\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image division. Divide pixels in *pSrc2* by *pSrc1*'s pixels.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.307** `NppStatus nppiDiv_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.308** `NppStatus nppiDiv_32s_C3RSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.309** `NppStatus nppiDiv_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.310 NppStatus nppiDiv\_32s\_C4IRSfs (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.311 NppStatus nppiDiv\_32s\_C4RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.312 NppStatus nppiDiv\_32sc\_AC4IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.313 NppStatus nppiDiv\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.314 NppStatus nppiDiv\_32sc\_C1IRSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.315 NppStatus nppiDiv\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.316 NppStatus nppiDiv\_32sc\_C3IRSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.317** `NppStatus nppiDiv_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.318** `NppStatus nppiDiv_32sc_C4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.319** `NppStatus nppiDiv_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.320** `NppStatus nppiDiv_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.321** `NppStatus nppiDiv_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.322 `NppStatus nppiDiv_8u_C1IRSfs (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.323 `NppStatus nppiDiv_8u_C1RSfs (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.324 NppStatus nppiDiv\_8u\_C3IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.325 NppStatus nppiDiv\_8u\_C3RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.326 NppStatus nppiDiv\_8u\_C4IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.327** `NppStatus npDiv_8u_C4RSfs (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.328** `NppStatus npDiv_Round_16s_AC4IRSfs (const Npp16s *pSrc, int nSrcStep, Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.329** `NppStatus nppiDiv_Round_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.330** `NppStatus nppiDiv_Round_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.331** `NppStatus nppiDiv_Round_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.332** `NppStatus nppiDiv_Round_16s_C3IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.333** `NppStatus nppiDiv_Round_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.334** `NppStatus nppiDiv_Round_16s_C4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.335** `NppStatus nppiDiv_Round_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.336** `NppStatus nppiDiv_Round_16u_AC4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.337** `NppStatus nppiDiv_Round_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.338** `NppStatus nppiDiv_Round_16u_C1IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.339** `NppStatus nppiDiv_Round_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.340** `NppStatus nppiDiv_Round_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.341** `NppStatus nppiDiv_Round_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.342** `NppStatus nppiDiv_Round_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.343** `NppStatus nppiDiv_Round_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.344** `NppStatus nppiDiv_Round_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.345** `NppStatus nppiDiv_Round_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.346** `NppStatus nppiDiv_Round_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.347** `NppStatus nppiDiv_Round_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.348** `NppStatus nppiDiv_Round_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.349 NppStatus nppiDiv\_Round\_8u\_C3RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.350 NppStatus nppiDiv\_Round\_8u\_C4IRSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.351** `NppStatus nppiDiv_Round_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.352** `NppStatus nppiDivC_16s_AC4IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.353** `NppStatus nppiDivC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).



*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.354 NppStatus nppiDivC\_16s\_C1RSfs (const Npp16s nConstant, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.355 NppStatus nppiDivC\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s nConstant, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.356 **NppStatus nppiDivC\_16s\_C3IRSfs** (const Npp16s \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.357 **NppStatus nppiDivC\_16s\_C3RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.358 **NppStatus nppiDivC\_16s\_C4IRSfs** (const Npp16s \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.359** `NppStatus nppiDivC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.360** `NppStatus nppiDivC_16sc_AC4IRSfs (const Npp16sc * pConstants, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.361** `NppStatus nppiDivC_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.362 NppStatus nppiDivC\_16sc\_C1IRSfs (const Npp16sc nConstant, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.363 NppStatus nppiDivC\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.364 **NppStatus nppiDivC\_16sc\_C3IRSfs** (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.365 **NppStatus nppiDivC\_16sc\_C3RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pConstants*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.366 **NppStatus nppiDivC\_16sc\_C4IRSfs** (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.367** `NppStatus nppiDivC_16sc_C4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.368** `NppStatus nppiDivC_16u_AC4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.369** `NppStatus nppiDivC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.370** `NppStatus nppiDivC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.371** `NppStatus nppiDivC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.372 NppStatus nppiDivC\_16u\_C3IRSfs (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.373 NppStatus nppiDivC\_16u\_C3RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.7.1.374 `NppStatus nppiDivC_16u_C4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.375 `NppStatus nppiDivC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.376 `NppStatus nppiDivC_32f_AC4IR (const Npp32f * pConstants, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.377** `NppStatus nppiDivC_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.378** `NppStatus nppiDivC_32f_C1IR (const Npp32f nConstant, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image divided by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.379** `NppStatus nppiDivC_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.380 NppStatus nppiDivC\_32f\_C3IR (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image divided by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.381 NppStatus nppiDivC\_32f\_C3R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pConstants*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.382 NppStatus nppiDivC\_32f\_C4IR (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image divided by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.383** `NppStatus nppiDivC_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.384** `NppStatus nppiDivC_32fc_AC4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.385** `NppStatus nppiDivC_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.386 NppStatus nppiDivC\_32fc\_C1IR (const Npp32fc *nConstant*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.387 NppStatus nppiDivC\_32fc\_C1R (const Npp32fc \* *pSrcI*, int *nSrcIStep*, const Npp32fc *nConstant*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.388 NppStatus nppiDivC\_32fc\_C3IR (const Npp32fc \* *pConstants*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.389** `NppStatus nppiDivC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.390** `NppStatus nppiDivC_32fc_C4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.391** `NppStatus nppiDivC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.392** `NppStatus nppiDivC_32s_AC4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* [pointer to a list of constant values, one per channel](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.393** `NppStatus nppiDivC_32s_AC4RSfs (const Npp32s * pSrcI, int nSrcIStep, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*pConstants* [pointer to a list of constant values, one per channel](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.394 `NppStatus nppiDivC_32s_C1RSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.395 `NppStatus nppiDivC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.396 `NppStatus nppiDivC_32s_C3IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.397** `NppStatus nppiDivC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.398** `NppStatus nppiDivC_32s_C4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.399** `NppStatus nppiDivC_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.400** `NppStatus nppiDivC_32sc_AC4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.401** `NppStatus nppiDivC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.402 NppStatus nppiDivC\_32sc\_C1IRSfs (const Npp32sc *nConstant*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.403 NppStatus nppiDivC\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *nConstant*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.404 NppStatus nppiDivC\_32sc\_C3IRSfs (const Npp32sc \* *pConstants*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.405** `NppStatus nppiDivC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.406** `NppStatus nppiDivC_32sc_C4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.407** `NppStatus nppiDivC_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.408** `NppStatus nppiDivC_8u_AC4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.409** `NppStatus nppiDivC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.410 NppStatus nppiDivC\_8u\_C1IRSfs (const Npp8u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.411 NppStatus nppiDivC\_8u\_C1RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.412 `NppStatus nppiDivC_8u_C3IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.413 `NppStatus nppiDivC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.414 `NppStatus nppiDivC_8u_C4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.415** `NppStatus nppiDivC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.416** `NppStatus nppiExp_16s_C1RSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.417** `NppStatus nppiExp_16s_C1RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).



*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.418** `NppStatus nppiExp_16s_C3IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.419** `NppStatus nppiExp_16s_C3RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.420** `NppStatus nppiExp_16u_C1IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.421** `NppStatus nppiExp_16u_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.422** `NppStatus nppiExp_16u_C3IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.423 NppStatus nppiExp\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.424 NppStatus nppiExp\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image exponential.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.425 NppStatus nppiExp\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image exponential.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.426 NppStatus nppiExp\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image exponential.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.427 NppStatus nppiExp\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image exponential.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.428 NppStatus nppiExp\_8u\_C1IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.429 NppStatus nppiExp\_8u\_C1RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.430 NppStatus nppiExp\_8u\_C3IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.431 NppStatus nppiExp\_8u\_C3RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.432 **NppStatus nppiLn\_16s\_C1IRSfs** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.433 **NppStatus nppiLn\_16s\_C1RSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.434 **NppStatus nppiLn\_16s\_C3IRSfs** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.435 NppStatus nppiLn\_16s\_C3RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.436 NppStatus nppiLn\_16u\_C1IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.437 NppStatus nppiLn\_16u\_C1RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.438 NppStatus nppiLn\_16u\_C3IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.439 NppStatus nppiLn\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.440 NppStatus nppiLn\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image natural logarithm.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.441 NppStatus nppiLn\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image natural logarithm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.442 NppStatus nppiLn\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image natural logarithm.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.443 NppStatus nppiLn\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image natural logarithm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.444 `NppStatus nppiLn_8u_C1IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.445 `NppStatus nppiLn_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.446 `NppStatus nppiLn_8u_C3IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.447 NppStatus nppiLn\_8u\_C3RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.448 NppStatus nppiLShiftC\_16u\_AC4IR (const Npp32u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.449 NppStatus nppiLShiftC\_16u\_AC4R (const Npp16u \* *pSrcI*, int *nSrcIStep*, const Npp32u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.450 `NppStatus nppiLShiftC_16u_C1R (const Npp32u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image left shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.451 `NppStatus nppiLShiftC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.452 `NppStatus nppiLShiftC_16u_C3IR (const Npp32u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image left shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.453** `NppStatus nppiLShiftC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.454** `NppStatus nppiLShiftC_16u_C4IR (const Npp32u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image left shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.455** `NppStatus nppiLShiftC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.456 **NppStatus nppiLShiftC\_32s\_AC4IR** (const Npp32u \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.457 **NppStatus nppiLShiftC\_32s\_AC4R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image left shift by constant with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.458 **NppStatus nppiLShiftC\_32s\_C1IR** (const Npp32u *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image left shift by constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.459** `NppStatus nppiLShiftC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.460** `NppStatus nppiLShiftC_32s_C3IR (const Npp32u * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image left shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.461** `NppStatus nppiLShiftC_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.462 **NppStatus nppiLShiftC\_32s\_C4IR** (const Npp32u \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image left shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.463 **NppStatus nppiLShiftC\_32s\_C4R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image left shift by constant.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.464 **NppStatus nppiLShiftC\_8u\_AC4IR** (const Npp32u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.465** `NppStatus nppiLShiftC_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, const Npp32u *pConstants, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.466** `NppStatus nppiLShiftC_8u_C1IR (const Npp32u nConstant, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image left shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.467** `NppStatus nppiLShiftC_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.468 `NppStatus nppiLShiftC_8u_C3IR (const Npp32u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image left shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.469 `NppStatus nppiLShiftC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.470 `NppStatus nppiLShiftC_8u_C4IR (const Npp32u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image left shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.471** `NppStatus nppiLShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.472** `NppStatus nppiMul_16s_AC4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.473** `NppStatus nppiMul_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.474** `NppStatus nppiMul_16s_C1RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.475** `NppStatus nppiMul_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.476 **NppStatus nppiMul\_16s\_C3IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.477 **NppStatus nppiMul\_16s\_C3RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.478 **NppStatus nppiMul\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.479** `NppStatus nppiMul_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.480** `NppStatus nppiMul_16sc_AC4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.481** `NppStatus nppiMul_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.482** `NppStatus nppiMul_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.483** `NppStatus nppiMul_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.484** `NppStatus nppiMul_16sc_C3IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.485** `NppStatus nppiMul_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.486** `NppStatus nppiMul_16sc_C4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.487** `NppStatus nppiMul_16sc_C4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.488** `NppStatus nppiMul_16u_AC4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.489** `NppStatus nppiMul_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.490** `NppStatus nppiMul_16u_C1IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.491** `NppStatus nppiMul_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.492** `NppStatus nppiMul_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.493** `NppStatus nppiMul_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.494** `NppStatus nppiMul_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.495** `NppStatus nppiMul_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.496** `NppStatus nppiMul_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.497** `NppStatus nppiMul_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.498** `NppStatus nppiMul_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.499** `NppStatus nppiMul_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.500** `NppStatus nppiMul_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.501** `NppStatus nppiMul_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.502** `NppStatus nppiMul_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.503** `NppStatus nppiMul_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.504** `NppStatus nppiMul_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.505** `NppStatus nppiMul_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.506** `NppStatus nppiMul_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.507** `NppStatus nppiMul_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.508** `NppStatus nppiMul_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.509** `NppStatus nppiMul_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.510** `NppStatus nppiMul_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.511** `NppStatus nppiMul_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.512** `NppStatus nppiMul_32s_AC4IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.513** `NppStatus nppiMul_32s_AC4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.514** `NppStatus nppiMul_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.515** `NppStatus nppiMul_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

1 channel 32-bit image multiplication. Multiply corresponding pixels in ROI.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.516** `NppStatus nppiMul_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.517** `NppStatus nppiMul_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.518** `NppStatus nppiMul_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.519** `NppStatus nppiMul_32s_C4IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.520** `NppStatus nppiMul_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.521 NppStatus nppiMul\_32sc\_AC4IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.522 NppStatus nppiMul\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.523 **NppStatus nppiMul\_32sc\_C1IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.524 **NppStatus nppiMul\_32sc\_C1RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.525 **NppStatus nppiMul\_32sc\_C3IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.526** `NppStatus nppiMul_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.527** `NppStatus nppiMul_32sc_C4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.528** `NppStatus nppiMul_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.529** `NppStatus nppiMul_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.530** `NppStatus nppiMul_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.531** `NppStatus nppiMul_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.532** `NppStatus nppiMul_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.533** `NppStatus nppiMul_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.534** `NppStatus nppiMul_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.535** `NppStatus nppiMul_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.536** `NppStatus nppiMul_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.537** `NppStatus nppiMulC_16s_AC4IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.538** `NppStatus nppiMulC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.539** `NppStatus nppiMulC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.540** `NppStatus nppiMulC_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s nConstant, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.541 NppStatus nppiMulC\_16s\_C3IRSfs (const Npp16s \* pConstants, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.542 NppStatus nppiMulC\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pConstants, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.543 `NppStatus nppiMulC_16s_C4IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.544 `NppStatus nppiMulC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.545 `NppStatus nppiMulC_16sc_AC4IRSfs (const Npp16sc * pConstants, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.546** `NppStatus nppiMulC_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.547** `NppStatus nppiMulC_16sc_C1IRSfs (const Npp16sc nConstant, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.548** `NppStatus nppiMulC_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.549** `NppStatus nppiMulC_16sc_C3IRSfs (const Npp16sc * pConstants, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.550** `NppStatus nppiMulC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.551 **NppStatus nppiMulC\_16sc\_C4IRSfs** (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.552 **NppStatus nppiMulC\_16sc\_C4RSfs** (const Npp16sc \* *pSrcI*, int *nSrcIStep*, const Npp16sc \* *pConstants*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.553 **NppStatus nppiMulC\_16u\_AC4IRSfs** (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.554** `NppStatus nppiMulC_16u_AC4RSfs (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pConstants, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.555** `NppStatus nppiMulC_16u_C1IRSfs (const Npp16u nConstant, Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.556** `NppStatus nppiMulC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.557** `NppStatus nppiMulC_16u_C3IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.558** `NppStatus nppiMulC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.559 NppStatus nppiMulC\_16u\_C4IRSfs (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.560 NppStatus nppiMulC\_16u\_C4RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.561 NppStatus nppiMulC\_32f\_AC4IR (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.562 NppStatus nppiMulC\_32f\_AC4R (const Npp32f \* *pSrcI*, int *nSrcIStep*, const Npp32f \* *pConstants*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.563 NppStatus nppiMulC\_32f\_C1IR (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image multiply by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.564** `NppStatus nppiMulC_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.565** `NppStatus nppiMulC_32f_C3IR (const Npp32f * pConstants, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image multiply by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.566** `NppStatus nppiMulC_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.7.1.567 `NppStatus nppiMulC_32f_C4IR (const Npp32f * pConstants, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image multiply by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.568 `NppStatus nppiMulC_32f_C4R (const Npp32f * pSrcI, int nSrcIStep, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image multiply by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.569 `NppStatus nppiMulC_32fc_AC4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.570 NppStatus nppiMulC\_32fc\_AC4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pConstants, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.571 NppStatus nppiMulC\_32fc\_C1IR (const Npp32fc nConstant, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.572 NppStatus nppiMulC\_32fc\_C1R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc nConstant, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.573** `NppStatus nppiMulC_32fc_C3IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.574** `NppStatus nppiMulC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.575** `NppStatus nppiMulC_32fc_C4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.576** `NppStatus nppiMulC_32fc_C4R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pConstants, Npp32fc *pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.577** `NppStatus nppiMulC_32s_AC4IRSfs (const Npp32s *pConstants, Npp32s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.578** `NppStatus nppiMulC_32s_AC4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.579** `NppStatus nppiMulC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.580** `NppStatus nppiMulC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.581** `NppStatus nppiMulC_32s_C3IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.582** `NppStatus nppiMulC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.583** `NppStatus nppiMulC_32s_C4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.584** `NppStatus nppiMulC_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.585** `NppStatus nppiMulC_32sc_AC4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.586** `NppStatus nppiMulC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.587** `NppStatus nppiMulC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.588** `NppStatus nppiMulC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.589** `NppStatus nppiMulC_32sc_C3IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.590** `NppStatus nppiMulC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.591** `NppStatus nppiMulC_32sc_C4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.592** `NppStatus nppiMulC_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.593 `NppStatus nppiMulC_8u_AC4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.594 `NppStatus nppiMulC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.595 `NppStatus nppiMulC_8u_C1IRSfs (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.596** `NppStatus nppiMulC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.597** `NppStatus nppiMulC_8u_C3IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.598** `NppStatus nppiMulC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.599** `NppStatus nppiMulC_8u_C4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.600** `NppStatus nppiMulC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.601 **NppStatus nppiMulCScale\_16u\_AC4IR** (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.602 **NppStatus nppiMulCScale\_16u\_AC4R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.603 **NppStatus nppiMulCScale\_16u\_C1IR** (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.604 `NppStatus nppiMulCScale_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.605 `NppStatus nppiMulCScale_16u_C3IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.606 `NppStatus nppiMulCScale_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.607 `NppStatus nppiMulCScale_16u_C4IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.608 `NppStatus nppiMulCScale_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.609 `NppStatus nppiMulCScale_8u_AC4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.7.1.610 `NppStatus nppiMulCScale_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.611 `NppStatus nppiMulCScale_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.612 `NppStatus nppiMulCScale_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.613 `NppStatus nppiMulCScale_8u_C3IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.614 `NppStatus nppiMulCScale_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.615 `NppStatus nppiMulCScale_8u_C4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.616** `NppStatus nppiMulCScale_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.617** `NppStatus nppiMulScale_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.618** `NppStatus nppiMulScale_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.619** `NppStatus nppiMulScale_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.620** `NppStatus nppiMulScale_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.621** `NppStatus nppiMulScale_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.622** `NppStatus nppiMulScale_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.623** `NppStatus nppiMulScale_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.624** `NppStatus nppiMulScale_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.625** `NppStatus nppiMulScale_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.626** `NppStatus nppiMulScale_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.627** `NppStatus nppiMulScale_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.628** `NppStatus nppiMulScale_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.629** `NppStatus nppiMulScale_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.630** `NppStatus nppiMulScale_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.631** `NppStatus nppiMulScale_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.



*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.632** `NppStatus nppiMulScale_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.633** `NppStatus nppiNot_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical not with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.634** `NppStatus nppiNot_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical not with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.635 NppStatus nppiNot\_8u\_C1IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.636 NppStatus nppiNot\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.637 NppStatus nppiNot\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.638 NppStatus nppiNot\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.639 NppStatus nppiNot\_8u\_C4IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.640 NppStatus nppiNot\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.641** `NppStatus nppiOr_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.642** `NppStatus nppiOr_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.643** `NppStatus nppiOr_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.644** `NppStatus nppiOr_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.645** `NppStatus nppiOr_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.646** `NppStatus nppiOr_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.647** `NppStatus nppiOr_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.648** `NppStatus nppiOr_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.649 NppStatus nppiOr\_32s\_AC4IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.650 NppStatus nppiOr\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.651 NppStatus nppiOr\_32s\_C1IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.652** `NppStatus nppiOr_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.653** `NppStatus nppiOr_32s_C3IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.654** `NppStatus nppiOr_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.655** `NppStatus nppiOr_32s_C4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.656** `NppStatus nppiOr_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.657** `NppStatus nppiOr_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.658** `NppStatus nppiOr_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.659** `NppStatus nppiOr_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.660** `NppStatus nppiOr_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.661** `NppStatus nppiOr_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.662** `NppStatus nppiOr_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.663** `NppStatus nppiOr_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.664** `NppStatus nppiOr_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.665 NppStatus nppiOrC\_16u\_AC4IR (const Npp16u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.666 NppStatus nppiOrC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.667 NppStatus nppiOrC\_16u\_C1IR (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.668** `NppStatus nppiOrC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.669** `NppStatus nppiOrC_16u_C3IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.670** `NppStatus nppiOrC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.671** `NppStatus nppiOrC_16u_C4IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.672** `NppStatus nppiOrC_16u_C4R (const Npp16u * pSrcI, int nSrcIStep, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.673** `NppStatus nppiOrC_32s_AC4IR (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.674** `NppStatus nppiOrC_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.675** `NppStatus nppiOrC_32s_C1IR (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.676** `NppStatus nppiOrC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.677 NppStatus nppiOrC\_32s\_C3IR (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.678 NppStatus nppiOrC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.679 NppStatus nppiOrC\_32s\_C4IR (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.680** `NppStatus nppiOrC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.681** `NppStatus nppiOrC_8u_AC4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.682** `NppStatus nppiOrC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.683 NppStatus nppiOrC\_8u\_C1IR (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.684 NppStatus nppiOrC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.685 NppStatus nppiOrC\_8u\_C3IR (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.686** `NppStatus nppiOrC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.687** `NppStatus nppiOrC_8u_C4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.688** `NppStatus nppiOrC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.689 NppStatus nppiRShiftC\_16s\_AC4IR (const Npp32u \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.690 NppStatus nppiRShiftC\_16s\_AC4R (const Npp16s \* *pSrcI*, int *nSrcIStep*, const Npp32u \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel image right shift by constant with unmodified alpha.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.691 NppStatus nppiRShiftC\_16s\_C1IR (const Npp32u *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel in place image right shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.692** `NppStatus nppiRShiftC_16s_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.693** `NppStatus nppiRShiftC_16s_C3IR (const Npp32u * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.694** `NppStatus nppiRShiftC_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.695 NppStatus nppiRShiftC\_16s\_C4IR (const Npp32u \* *pConstants*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel in place image right shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.696 NppStatus nppiRShiftC\_16s\_C4R (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pConstants*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.697 NppStatus nppiRShiftC\_16u\_AC4IR (const Npp32u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.698** `NppStatus nppiRShiftC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.699** `NppStatus nppiRShiftC_16u_C11R (const Npp32u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.700** `NppStatus nppiRShiftC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.701 NppStatus nppiRShiftC\_16u\_C3IR (const Npp32u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.702 NppStatus nppiRShiftC\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pConstants*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.703 NppStatus nppiRShiftC\_16u\_C4IR (const Npp32u \* *pConstants*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.704** `NppStatus nppiRShiftC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.705** `NppStatus nppiRShiftC_32s_AC4IR (const Npp32u * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.706** `NppStatus nppiRShiftC_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.707 NppStatus nppiRShiftC\_32s\_C1IR (const Npp32u *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.708 NppStatus nppiRShiftC\_32s\_C1R (const Npp32s \* *pSrcI*, int *nSrcIStep*, const Npp32u *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.709 NppStatus nppiRShiftC\_32s\_C3IR (const Npp32u \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.710** `NppStatus nppiRShiftC_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.711** `NppStatus nppiRShiftC_32s_C4IR (const Npp32u * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.712** `NppStatus nppiRShiftC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.1.713 **NppStatus nppiRShiftC\_8s\_AC4IR** (const Npp32u \* *pConstants*, Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.714 **NppStatus nppiRShiftC\_8s\_AC4R** (const Npp8s \* *pSrcI*, int *nSrcIStep*, const Npp32u \* *pConstants*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel image right shift by constant with unmodified alpha.

##### Parameters:

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.715 **NppStatus nppiRShiftC\_8s\_C1IR** (const Npp32u *nConstant*, Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit signed char channel in place image right shift by constant.

##### Parameters:

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.716** `NppStatus nppiRShiftC_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit signed char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.717** `NppStatus nppiRShiftC_8s_C3IR (const Npp32u * pConstants, Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit signed char channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.718** `NppStatus nppiRShiftC_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit signed char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.719 `NppStatus nppiRShiftC_8s_C4IR (const Npp32u * pConstants, Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel in place image right shift by constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.720 `NppStatus nppiRShiftC_8s_C4R (const Npp8s * pSrcI, int nSrcIStep, const Npp32u * pConstants, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel image right shift by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.721 `NppStatus nppiRShiftC_8u_AC4IR (const Npp32u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.722** `NppStatus nppiRShiftC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.723** `NppStatus nppiRShiftC_8u_C1IR (const Npp32u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.724** `NppStatus nppiRShiftC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.725 NppStatus nppiRShiftC\_8u\_C3IR (const Npp32u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.726 NppStatus nppiRShiftC\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.727 NppStatus nppiRShiftC\_8u\_C4IR (const Npp32u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.728** `NppStatus nppiRShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.729** `NppStatus nppiSqr_16s_AC4IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.730** `NppStatus nppiSqr_16s_AC4RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.731 NppStatus nppiSqr\_16s\_C1IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.732 NppStatus nppiSqr\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.733 NppStatus nppiSqr\_16s\_C3IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.734 **NppStatus nppiSqr\_16s\_C3RSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.735 **NppStatus nppiSqr\_16s\_C4IRSfs** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.736 **NppStatus nppiSqr\_16s\_C4RSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.737 NppStatus nppiSqr\_16u\_AC4IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.738 NppStatus nppiSqr\_16u\_AC4RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.739 NppStatus nppiSqr\_16u\_C1IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.740 NppStatus nppiSqr\_16u\_C1RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.741 NppStatus nppiSqr\_16u\_C3IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.742 NppStatus nppiSqr\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.743 NppStatus nppiSqr\_16u\_C4IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.744 NppStatus nppiSqr\_16u\_C4RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.745 NppStatus nppiSqr\_32f\_AC4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image squared with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.746 NppStatus nppiSqr\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image squared with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.747 NppStatus nppiSqr\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.748 NppStatus nppiSqr\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.749 NppStatus nppiSqr\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.750 NppStatus nppiSqr\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.751 NppStatus nppiSqr\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.752** `NppStatus nppiSqr_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.753** `NppStatus nppiSqr_8u_AC4IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.754** `NppStatus nppiSqr_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.755 NppStatus nppiSqr\_8u\_C1IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.756 NppStatus nppiSqr\_8u\_C1RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.757 NppStatus nppiSqr\_8u\_C3IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.758 **NppStatus nppiSqr\_8u\_C3RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.759 **NppStatus nppiSqr\_8u\_C4IRSfs** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.760 **NppStatus nppiSqr\_8u\_C4RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.761 NppStatus nppiSqrt\_16s\_AC4IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.762 NppStatus nppiSqrt\_16s\_AC4RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.763 NppStatus nppiSqrt\_16s\_C1IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.764 NppStatus nppiSqrt\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.765 NppStatus nppiSqrt\_16s\_C3IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.766 NppStatus nppiSqrt\_16s\_C3RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.767 NppStatus nppiSqrt\_16s\_C4IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.768 NppStatus nppiSqrt\_16s\_C4RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.769 NppStatus nppiSqrt\_16u\_AC4IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.770** `NppStatus nppiSqrt_16u_AC4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.771** `NppStatus nppiSqrt_16u_C1IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.772** `NppStatus nppiSqrt_16u_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).



*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.773** `NppStatus nppiSqrt_16u_C3IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.774** `NppStatus nppiSqrt_16u_C3RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.775** `NppStatus nppiSqrt_16u_C4IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.776** `NppStatus nppiSqrt_16u_C4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.777** `NppStatus nppiSqrt_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image square root with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.778** `NppStatus nppiSqrt_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image square root with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.779 NppStatus nppiSqrt\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.780 NppStatus nppiSqrt\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.781 NppStatus nppiSqrt\_32f\_C3IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.782 NppStatus nppiSqrt\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.783 NppStatus nppiSqrt\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.784 NppStatus nppiSqrt\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.785** `NppStatus nppiSqrt_8u_AC4IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.786** `NppStatus nppiSqrt_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.787** `NppStatus nppiSqrt_8u_C1IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.788 NppStatus nppiSqrt\_8u\_C1RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.789 NppStatus nppiSqrt\_8u\_C3IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.790 NppStatus nppiSqrt\_8u\_C3RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.791 `NppStatus nppiSqrt_8u_C4IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.792 `NppStatus nppiSqrt_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.793 `NppStatus nppiSub_16s_AC4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.794** `NppStatus nppiSub_16s_AC4RSfs (const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2, int nSrc2Step, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.795** `NppStatus nppiSub_16s_C1IRSfs (const Npp16s *pSrc, int nSrcStep, Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.796** `NppStatus nppiSub_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.797** `NppStatus nppiSub_16s_C3IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.798** `NppStatus nppiSub_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.799** `NppStatus nppiSub_16s_C4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.800** `NppStatus nppiSub_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.801 NppStatus nppiSub\_16sc\_AC4IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.802 NppStatus nppiSub\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.803 `NppStatus nppiSub_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.804 `NppStatus nppiSub_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.805 `NppStatus nppiSub_16sc_C3IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.806** `NppStatus nppiSub_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.807** `NppStatus nppiSub_16sc_C4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.808** `NppStatus nppiSub_16sc_C4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.809** `NppStatus nppiSub_16u_AC4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.810** `NppStatus nppiSub_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.811** `NppStatus nppiSub_16u_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.812** `NppStatus nppiSub_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.813 NppStatus nppiSub\_16u\_C3IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.814 NppStatus nppiSub\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.7.1.815 `NppStatus nppiSub_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.816 `NppStatus nppiSub_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.817 `NppStatus nppiSub_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.818** `NppStatus nppiSub_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.819** `NppStatus nppiSub_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.820** `NppStatus nppiSub_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.821** `NppStatus nppiSub_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.822** `NppStatus nppiSub_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.823** `NppStatus nppiSub_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.824** `NppStatus nppiSub_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.825** `NppStatus nppiSub_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.826** `NppStatus nppiSub_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.827** `NppStatus nppiSub_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.828** `NppStatus npipiSub_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.829** `NppStatus npipiSub_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.830** `NppStatus nppiSub_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.831** `NppStatus nppiSub_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.832** `NppStatus nppiSub_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.833 **NppStatus nppiSub\_32s\_AC4IRSfs** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.834 **NppStatus nppiSub\_32s\_AC4RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.835** `NppStatus nppiSub_32s_C1RSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.836** `NppStatus nppiSub_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image subtraction. Subtract pSrc1's pixels from corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.837** `NppStatus nppiSub_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.838** `NppStatus nppiSub_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.839** `NppStatus nppiSub_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.840** `NppStatus nppiSub_32s_C4IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.841** `NppStatus nppiSub_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.842** `NppStatus nppiSub_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.843** `NppStatus npipiSub_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.844** `NppStatus npipiSub_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.845** `NppStatus nppiSub_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.846** `NppStatus nppiSub_32sc_C3RSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.847** `NppStatus nppiSub_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.848** `NppStatus nppiSub_32sc_C4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.849** `NppStatus nppiSub_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.850 NppStatus nppiSub\_8u\_AC4IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.851 NppStatus nppiSub\_8u\_AC4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.852** `NppStatus nppiSub_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.853** `NppStatus nppiSub_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.854** `NppStatus nppiSub_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.855** `NppStatus nppiSub_8u_C3RSfs (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.856** `NppStatus nppiSub_8u_C4IRSfs (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.857** `NppStatus nppiSub_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.858** `NppStatus nppiSubC_16s_AC4IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.859** `NppStatus nppiSubC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.860 NppStatus npplSubC\_16s\_C1RSfs (const Npp16s nConstant, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.861 NppStatus npplSubC\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s nConstant, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.862 `NppStatus nppiSubC_16s_C3IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.863 `NppStatus nppiSubC_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.864 `NppStatus nppiSubC_16s_C4IRSfs (const Npp16s * pConstants, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.865** `NppStatus nppiSubC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pConstants, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.866** `NppStatus nppiSubC_16sc_AC4IRSfs (const Npp16sc * pConstants, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.867** `NppStatus nppiSubC_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.868 NppStatus nppiSubC\_16sc\_C1RSfs (const Npp16sc *nConstant*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.869 NppStatus nppiSubC\_16sc\_C1RSfs (const Npp16sc \* *pSrcI*, int *nSrcIStep*, const Npp16sc *nConstant*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.870 **NppStatus nppiSubC\_16sc\_C3IRSfs** (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.871 **NppStatus nppiSubC\_16sc\_C3RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pConstants*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.872 **NppStatus nppiSubC\_16sc\_C4IRSfs** (const Npp16sc \* *pConstants*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.873** `NppStatus nppiSubC_16sc_C4RSfs (const Npp16sc * pSrcI, int nSrcIStep, const Npp16sc * pConstants, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.874** `NppStatus nppiSubC_16u_AC4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.7.1.875** `NppStatus nppiSubC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.876** `NppStatus nppiSubC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.877** `NppStatus nppiSubC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.878** `NppStatus nppiSubC_16u_C3IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.879** `NppStatus nppiSubC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.880** `NppStatus nppiSubC_16u_C4IRSfs (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.881** `NppStatus nppiSubC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.882** `NppStatus nppiSubC_32f_AC4IR (const Npp32f * pConstants, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.883** `NppStatus nppiSubC_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.884** `NppStatus nppiSubC_32f_C1IR (const Npp32f nConstant, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtract constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.885** `NppStatus nppiSubC_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.886 NppStatus nppiSubC\_32f\_C3IR (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.887 NppStatus nppiSubC\_32f\_C3R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pConstants*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.888 NppStatus nppiSubC\_32f\_C4IR (const Npp32f \* *pConstants*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.889** `NppStatus nppiSubC_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pConstants, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.890** `NppStatus nppiSubC_32fc_AC4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.891** `NppStatus nppiSubC_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.892** `NppStatus nppiSubC_32fc_C1IR (const Npp32fc nConstant, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.893** `NppStatus nppiSubC_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc nConstant, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.894** `NppStatus nppiSubC_32fc_C3IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.895** `NppStatus nppiSubC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.896** `NppStatus nppiSubC_32fc_C4IR (const Npp32fc * pConstants, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.897** `NppStatus nppiSubC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pConstants, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.



*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.898** `NppStatus nppiSubC_32s_AC4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* [pointer to a list of constant values, one per channel](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.899** `NppStatus nppiSubC_32s_AC4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* [pointer to a list of constant values, one per channel](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.900 `NppStatus nppiSubC_32s_C1RSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.901 `NppStatus nppiSubC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.902 `NppStatus nppiSubC_32s_C3RSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.903** `NppStatus nppiSubC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.904** `NppStatus nppiSubC_32s_C4IRSfs (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.905** `NppStatus nppiSubC_32s_C4RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.906** `NppStatus nppiSubC_32sc_AC4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.907** `NppStatus nppiSubC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.908 NppStatus nppiSubC\_32sc\_C1IRSfs (const Npp32sc *nConstant*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.909 NppStatus nppiSubC\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *nConstant*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.910 NppStatus nppiSubC\_32sc\_C3IRSfs (const Npp32sc \* *pConstants*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.911** `NppStatus nppiSubC_32sc_C3RSfs (const Npp32sc * pSrcI, int nSrcIStep, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.912** `NppStatus nppiSubC_32sc_C4IRSfs (const Npp32sc * pConstants, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.913** `NppStatus nppiSubC_32sc_C4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pConstants, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.914** `NppStatus nppiSubC_8u_AC4IRSfs (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.915** `NppStatus nppiSubC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.916 NppStatus npplSubC\_8u\_C1RSfs (const Npp8u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.917 NppStatus npplSubC\_8u\_C1RSfs (const Npp8u \* pSrcI, int nSrcIStep, const Npp8u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.7.1.918 NppStatus nppiSubC\_8u\_C3IRSfs (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.919 NppStatus nppiSubC\_8u\_C3RSfs (const Npp8u \* *pSrcI*, int *nSrcIStep*, const Npp8u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.920 NppStatus nppiSubC\_8u\_C4IRSfs (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.921** `NppStatus nppiSubC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.922** `NppStatus nppiXor_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.923** `NppStatus nppiXor_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.924** `NppStatus nppiXor_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.925** `NppStatus nppiXor_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.926** `NppStatus nppiXor_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.927** `NppStatus nppiXor_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.928** `NppStatus nppiXor_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.929** `NppStatus nppiXor_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.930** `NppStatus nppiXor_32s_AC4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.931** `NppStatus nppiXor_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.932** `NppStatus nppiXor_32s_C1IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.933** `NppStatus nppiXor_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.934 NppStatus nppiXor\_32s\_C3IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.935 NppStatus nppiXor\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.936 NppStatus nppiXor\_32s\_C4IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.937** `NppStatus nppiXor_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.938** `NppStatus nppiXor_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.939** `NppStatus nppiXor_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.940** `NppStatus nppiXor_8u_C11R (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.941** `NppStatus nppiXor_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.942** `NppStatus nppiXor_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.943** `NppStatus nppiXor_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.944** `NppStatus nppiXor_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.945** `NppStatus nppiXor_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.946** `NppStatus nppiXorC_16u_AC4IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.947** `NppStatus nppiXorC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.948 `NppStatus nppiXorC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.949 `NppStatus nppiXorC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image logical exclusive or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.950 `NppStatus nppiXorC_16u_C3IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image logical exclusive or with constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.951** `NppStatus nppiXorC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.952** `NppStatus nppiXorC_16u_C4IR (const Npp16u * pConstants, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.953** `NppStatus nppiXorC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pConstants, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.954 `NppStatus nppiXorC_32s_AC4IR (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.

##### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.955 `NppStatus nppiXorC_32s_AC4R (const Npp32s * pSrcI, int nSrcIStep, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.

##### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.1.956 `NppStatus nppiXorC_32s_C1IR (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical exclusive or with constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.957** `NppStatus nppiXorC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.958** `NppStatus nppiXorC_32s_C3IR (const Npp32s * pConstants, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image logical exclusive or with constant.

**Parameters:**

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.959** `NppStatus nppiXorC_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pConstants, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.960 **NppStatus nppiXorC\_32s\_C4IR** (const Npp32s \* *pConstants*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical exclusive or with constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.961 **NppStatus nppiXorC\_32s\_C4R** (const Npp32s \* *pSrcI*, int *nSrcIStep*, const Npp32s \* *pConstants*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical exclusive or with constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.962 **NppStatus nppiXorC\_8u\_AC4IR** (const Npp8u \* *pConstants*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.7.1.963** `NppStatus nppiXorC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.964** `NppStatus nppiXorC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.1.965** `NppStatus nppiXorC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.1.966 `NppStatus nppiXorC_8u_C3IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical exclusive or with constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.967 `NppStatus nppiXorC_8u_C3R (const Npp8u * pSrcI, int nSrcIStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical exclusive or with constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.1.968 `NppStatus nppiXorC_8u_C4IR (const Npp8u * pConstants, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with constant.

#### Parameters:

*pConstants* pointer to a list of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.1.969** `NppStatus nppiXorC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.8 Threshold and Compare Operations

Methods for pixel-wise threshold and compare operations.

### Threshold

Threshold pixels.

- **NppStatus nppiThreshold\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nThreshold, **NppCmpOp** eComparisonOperation)  
*32-bit floating point threshold.*
- **NppStatus nppiThreshold\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** aThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned image threshold, not affecting Alpha.*

### Image Compare Methods

Compare the pixels of two images and create a binary result image.

In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u\_C1. False is represented by 0, true by NPP\_MAX\_8U.

- **NppStatus nppiCompare\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned image compare.*
- **NppStatus nppiCompare\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*32-bit floating point image compare.*

#### 7.8.1 Detailed Description

Methods for pixel-wise threshold and compare operations.

#### 7.8.2 Function Documentation

- 7.8.2.1 NppStatus nppiCompare\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)

32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.2 NppStatus nppiCompare\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 8-bit unsigned image compare, not affecting Alpha.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.3 NppStatus nppiCompare\_8u\_C4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 8-bit unsigned image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.4 NppStatus nppiThreshold\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, Npp32f nThreshold, NppCmpOp eComparisonOperation)

32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.8.2.5 NppStatus nppiThreshold\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u aThresholds[3], NppCmpOp eComparisonOperation)

4 channel 8-bit unsigned image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

## 7.9 Compression

Image compression primitives.

### Functions

- **NppStatus** **nppiQuantFwdRawTableInit\_JPEG\_8u** (**Npp8u** \*hpQuantRawTable, int nQualityFactor)  
*Apply quality factor to raw 8-bit quantization table.*
- **NppStatus** **nppiQuantFwdTableInit\_JPEG\_8u16u** (const **Npp8u** \*hpQuantRawTable, **Npp16u** \*hpQuantFwdRawTable)  
*Initializes a quantization table for **nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R()**.*
- **NppStatus** **nppiQuantInvTableInit\_JPEG\_8u16u** (const **Npp8u** \*hpQuantRawTable, **Npp16u** \*hpQuantFwdRawTable)  
*Initializes a quantization table for **nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R()**.*
- **NppStatus** **nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, const **Npp16u** \*pQuantFwdTable, **NppiSize** oSizeROI)  
*Forward DCT, quantization and level shift part of the JPEG encoding.*
- **NppStatus** **nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, const **Npp16u** \*pQuantInvTable, **NppiSize** oSizeROI)  
*Inverse DCT, de-quantization and level shift part of the JPEG decoding.*

### 7.9.1 Detailed Description

Image compression primitives.

The JPEG standard defines a flow of level shift, DCT and quantization for forward JPEG transform and inverse level shift, IDCT and de-quantization for inverse JPEG transform. This group has the functions for both forward and inverse functions.

### 7.9.2 Function Documentation

#### 7.9.2.1 **NppStatus** **nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, const **Npp16u** \*pQuantFwdTable, **NppiSize** oSizeROI)

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pDst** Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*pQuantFwdTable* Forward quantization tables for JPEG encoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

#### 7.9.2.2 NppStatus nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, const Npp16u \* pQuantInvTable, NppiSize oSizeROI)

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pQuantInvTable* Inverse quantization tables for JPEG decoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

#### 7.9.2.3 NppStatus nppiQuantFwdRawTableInit\_JPEG\_8u (Npp8u \* hpQuantRawTable, int nQualityFactor)

Apply quality factor to raw 8-bit quantization table.

This is effectively and in-place method that modifies a given raw quantization table based on a quality factor. Note that this method is a host method and that the pointer to the raw quantization table is a host pointer.

#### Parameters:

*hpQuantRawTable* Raw quantization table.

*nQualityFactor* Quality factor for the table. Range is [1:100].

#### Returns:

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if hpQuantRawTable is 0.

#### 7.9.2.4 NppStatus nppiQuantFwdTableInit\_JPEG\_8u16u (const Npp8u \* hpQuantRawTable, Npp16u \* hpQuantFwdRawTable)

Initializes a quantization table for [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).

The method creates a 16-bit version of the raw table and converts the data order from zigzag layout to original row-order layout since raw quantization tables are typically stored in zigzag format.

This method is a host method. It consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) function.

#### Parameters:

*hpQuantRawTable* Host pointer to raw quantization table as returned by [nppiQuantFwdRawTableInit\\_JPEG\\_8u\(\)](#). The raw quantization table is assumed to be in zigzag order.

*hpQuantFwdRawTable* Forward quantization table for use with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).

#### Returns:

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) pQuantRawTable is 0.

#### 7.9.2.5 NppStatus nppiQuantInvTableInit\_JPEG\_8u16u (const Npp8u \* hpQuantRawTable, Npp16u \* hpQuantFwdRawTable)

Initializes a quantization table for [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\(\)](#).

The [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) method uses a quantization table in a 16-bit format allowing for faster processing. In addition it converts the data order from zigzag layout to original row-order layout. Typically raw quantization tables are stored in zigzag format.

This method is a host method and consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) function.

#### Parameters:

*hpQuantRawTable* Raw quantization table.

*hpQuantFwdRawTable* Inverse quantization table.

#### Returns:

[NPP\\_NULL\\_POINTER\\_ERROR](#) pQuantRawTable or pQuantFwdRawTable is 0.

## 7.10 Geometric Transforms

Routines manipulating an image's geometry.

### Resize

Resizes 8 bit images.

Handles C1 and C4 images.

- **NppStatus nppiResize\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** dstROISize, double xFactor, double yFactor, int eInterpolation)  
*8-bit unsigned image resize.*
- **NppStatus nppiResize\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** dstROISize, double xFactor, double yFactor, int eInterpolation)  
*4 channel 8-bit unsigned image resize.*

### Rotate

Rotates an image around the origin (0,0) and then shifts it.

- **NppStatus nppiGetRotateQuad** (**NppiRect** oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)  
*Compute shape of rotated image.*
- **NppStatus nppiGetRotateBound** (**NppiRect** oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)  
*Compute bounding-box of rotated image.*
- **NppStatus nppiRotate\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*8-bit unsigned image rotate.*
- **NppStatus nppiRotate\_8u\_C3R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*3 channel 8-bit unsigned image rotate.*
- **NppStatus nppiRotate\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*4 channel 8-bit unsigned image rotate.*
- **NppStatus nppiRotate\_8u\_AC4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 8-bit unsigned image rotate ignoring alpha channel.*

- **NppStatus nppiRotate\_16u\_C1R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*16-bit unsigned image rotate.*

- **NppStatus nppiRotate\_16u\_C3R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*3 channel 16-bit unsigned image rotate.*

- **NppStatus nppiRotate\_16u\_C4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 16-bit unsigned image rotate.*

- **NppStatus nppiRotate\_16u\_AC4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 16-bit unsigned image rotate ignoring alpha channel.*

- **NppStatus nppiRotate\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*3 channel 32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 32-bit float image rotate ignoring alpha channel.*

## Mirror

Mirrors images horizontally, vertically and diagonally.

- **NppStatus nppiMirror\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)

*8-bit unsigned image mirror.*

- `NppStatus nppiMirror_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 8-bit unsigned image mirror.*
- `NppStatus nppiMirror_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 8-bit unsigned image mirror.*
- `NppStatus nppiMirror_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 8-bit unsigned image mirror not affecting alpha.*
- `NppStatus nppiMirror_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 16-bit unsigned image mirror not affecting alpha.*
- `NppStatus nppiMirror_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*32-bit image mirror.*
- `NppStatus nppiMirror_32s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit image mirror not affecting alpha.*
- `NppStatus nppiMirror_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*32-bit float image mirror.*
- `NppStatus nppiMirror_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit float image mirror.*

- **NppStatus** **nppiMirror\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 32-bit float image mirror.*
- **NppStatus** **nppiMirror\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 32-bit float image mirror not affecting alpha.*

## Affine warping, affine transform calculation

Affine warping of an image is the transform of image pixel positions, defined by the following formulas:

$$X_{new} = C_{00} * x + C_{01} * y + C_{02} \quad Y_{new} = C_{10} * x + C_{11} * y + C_{12} \quad C = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \end{bmatrix}$$

That is, any pixel with coordinates  $(X_{new}, Y_{new})$  in the transformed image is sourced from coordinates  $(x, y)$  in the original image.

The mapping  $C$  is completely specified by 6 values  $C_{ij}, i = \overline{0,1}, j = \overline{0,2}$ . The transform maps parallel lines to parallel lines and preserves ratios of distances of points to lines. Implementation specific properties are discussed in each function's documentation.

- **NppStatus** **nppiGetAffineTransform** (**NppiRect** srcRoi, const double quad[4][2], double coeffs[2][3])  
*Calculates affine transform coefficients given source rectangular ROI and its destination quadrangle projection.*
- **NppStatus** **nppiGetAffineQuad** (**NppiRect** srcRoi, double quad[4][2], const double coeffs[2][3])  
*Calculates affine transform projection of given source rectangular ROI.*
- **NppStatus** **nppiGetAffineBound** (**NppiRect** srcRoi, double bound[2][2], const double coeffs[2][3])  
*Calculates bounding box of the affine transform projection of the given source rectangular ROI.*
- **NppStatus** **nppiWarpAffine\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp8u** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)  
*Affine transform of an image (8bit unsigned integer, single channel).*
- **NppStatus** **nppiWarpAffine\_8u\_C3R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp8u** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)  
*Affine transform of an image (8bit unsigned integer, three channels).*
- **NppStatus** **nppiWarpAffine\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp8u** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)  
*Affine transform of an image (8bit unsigned integer, four channels).*
- **NppStatus** **nppiWarpAffine\_8u\_AC4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp8u** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffine_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst[3]`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffine_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst[4]`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, four planes).*

- `NppStatus nppiWarpAffineBack_8u_C1R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, single channel).*

- `NppStatus nppiWarpAffineBack_8u_C3R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, three channels).*

- `NppStatus nppiWarpAffineBack_8u_C4R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, four channels).*

- `NppStatus nppiWarpAffineBack_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffineBack_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst[3]`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffineBack_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u *pDst[4]`, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (8bit unsigned integer, four planes).*

- `NppStatus nppiWarpAffineQuad_8u_C1R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, single channel).*

- `NppStatus nppiWarpAffineQuad_8u_C3R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, three channels).*

- `NppStatus nppiWarpAffineQuad_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, four channels).*

- `NppStatus nppiWarpAffineQuad_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffineQuad_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffineQuad_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (8bit unsigned integer, four planes).*

- `NppStatus nppiWarpAffine_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpAffine_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpAffine_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four channels).*

- `NppStatus nppiWarpAffine_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffine_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffine_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four planes).*



- `NppStatus nppiWarpAffineBack_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpAffineBack_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpAffineBack_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, four channels).*

- `NppStatus nppiWarpAffineBack_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffineBack_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffineBack_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (16bit unsigned integer, four planes).*

- `NppStatus nppiWarpAffineQuad_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpAffineQuad_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpAffineQuad_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four channels).*

- `NppStatus nppiWarpAffineQuad_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpAffineQuad_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpAffineQuad_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (16bit unsigned integer, four planes).*

- `NppStatus nppiWarpAffine_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, single channel).*

- `NppStatus nppiWarpAffine_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, three channels).*

- `NppStatus nppiWarpAffine_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, four channels).*

- `NppStatus nppiWarpAffine_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, four channels RGBA).*

- `NppStatus nppiWarpAffine_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, three planes).*

- `NppStatus nppiWarpAffine_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Affine transform of an image (32bit float, four planes).*

- `NppStatus nppiWarpAffineBack_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, single channel).*

- `NppStatus nppiWarpAffineBack_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, three channels).*

- **NppStatus** **nppiWarpAffineBack\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, four channels).*

- **NppStatus** **nppiWarpAffineBack\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, four channels RGBA).*

- **NppStatus** **nppiWarpAffineBack\_32f\_P3R** (const **Npp32f** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, three planes).*

- **NppStatus** **nppiWarpAffineBack\_32f\_P4R** (const **Npp32f** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst[4], int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit float, four planes).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, single channel).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, three channels).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, four channels).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, four channels RGBA).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_P3R** (const **Npp32f** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, three planes).*

- **NppStatus** **nppiWarpAffineQuad\_32f\_P4R** (const **Npp32f** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32f** \*pDst[4], int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit float, four planes).*

- `NppStatus nppiWarpAffine_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, single channel).*

- `NppStatus nppiWarpAffine_32s_C3R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, three channels).*

- `NppStatus nppiWarpAffine_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, four channels).*

- `NppStatus nppiWarpAffine_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, four channels RGBA).*

- `NppStatus nppiWarpAffine_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst[3]`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, three planes).*

- `NppStatus nppiWarpAffine_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst[4]`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Affine transform of an image (32bit signed integer, four planes).*

- `NppStatus nppiWarpAffineBack_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Inverse affine transform of an image (32bit signed integer, single channel).*

- `NppStatus nppiWarpAffineBack_32s_C3R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Inverse affine transform of an image (32bit signed integer, three channels).*

- `NppStatus nppiWarpAffineBack_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Inverse affine transform of an image (32bit signed integer, four channels).*

- `NppStatus nppiWarpAffineBack_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `srcRoi`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `dstRoi`, const double `coeffs[2][3]`, int `eInterpolation`)

*Inverse affine transform of an image (32bit signed integer, four channels RGBA).*

- **NppStatus** **nppiWarpAffineBack\_32s\_P3R** (const **Npp32s** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32s** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit signed integer, three planes).*

- **NppStatus** **nppiWarpAffineBack\_32s\_P4R** (const **Npp32s** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32s** \*pDst[4], int nDstStep, **NppiRect** dstRoi, const double coeffs[2][3], int eInterpolation)

*Inverse affine transform of an image (32bit signed integer, four planes).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_C1R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, single channel).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_C3R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, three channels).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_C4R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, four channels).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_AC4R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, four channels RGBA).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_P3R** (const **Npp32s** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, three planes).*

- **NppStatus** **nppiWarpAffineQuad\_32s\_P4R** (const **Npp32s** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, const double srcQuad[4][2], **Npp32s** \*pDst[4], int nDstStep, **NppiRect** dstRoi, const double dstQuad[4][2], int eInterpolation)

*Affine transform of an image (32bit signed integer, four planes).*

## Perspective warping, perspective transform calculation

Perspective warping of an image is the transform of image pixel positions, defined by the following formulas:

$$X_{new} = \frac{C_{00} * x + C_{01} * y + C_{02}}{C_{20} * x + C_{21} * y + C_{22}} \quad Y_{new} = \frac{C_{10} * x + C_{11} * y + C_{12}}{C_{20} * x + C_{21} * y + C_{22}} \quad C = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}$$

That is, any pixel of the transformed image with coordinates  $(X_{new}, Y_{new})$  has a preimage with coordinates  $(x, y)$ .

The mapping  $C$  is fully defined by 8 values  $C_{ij}, (i, j) = \overline{0, 2}$ , except of  $C_{22}$ , which is a normalizer. The transform has a property of mapping any convex quadrangle to a convex quadrangle, which is used in a group of functions `npplWarpPerspectiveQuad`. The NPPI implementation of perspective transform has some issues which are discussed in each function's documentation.

- `NppStatus npplGetPerspectiveTransform` (`NppiRect` srcRoi, const double quad[4][2], double coeffs[3][3])  
*Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.*
- `NppStatus npplGetPerspectiveQuad` (`NppiRect` srcRoi, double quad[4][2], const double coeffs[3][3])  
*Calculates perspective transform projection of given source rectangular ROI.*
- `NppStatus npplGetPerspectiveBound` (`NppiRect` srcRoi, double bound[2][2], const double coeffs[3][3])  
*Calculates bounding box of the perspective transform projection of the given source rectangular ROI.*
- `NppStatus npplWarpPerspective_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, single channel).*
- `NppStatus npplWarpPerspective_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, three channels).*
- `NppStatus npplWarpPerspective_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, four channels).*
- `NppStatus npplWarpPerspective_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, four channels RGBA).*
- `NppStatus npplWarpPerspective_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, three planes).*
- `NppStatus npplWarpPerspective_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)  
*Perspective transform of an image (8bit unsigned integer, four planes).*
- `NppStatus npplWarpPerspectiveBack_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, single channel).*

- `NppStatus nppiWarpPerspectiveBack_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, three channels).*

- `NppStatus nppiWarpPerspectiveBack_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, four channels).*

- `NppStatus nppiWarpPerspectiveBack_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveBack_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, three planes).*

- `NppStatus nppiWarpPerspectiveBack_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp8u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (8bit unsigned integer, four planes).*

- `NppStatus nppiWarpPerspectiveQuad_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, single channel).*

- `NppStatus nppiWarpPerspectiveQuad_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, three channels).*

- `NppStatus nppiWarpPerspectiveQuad_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, four channels).*

- `NppStatus nppiWarpPerspectiveQuad_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveQuad_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, three planes).*



- `NppStatus nppiWarpPerspectiveQuad_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp8u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (8bit unsigned integer, four planes).*

- `NppStatus nppiWarpPerspective_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpPerspective_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpPerspective_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four channels).*

- `NppStatus nppiWarpPerspective_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpPerspective_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpPerspective_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four planes).*

- `NppStatus nppiWarpPerspectiveBack_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpPerspectiveBack_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpPerspectiveBack_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, four channels).*



- `NppStatus nppiWarpPerspectiveBack_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveBack_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpPerspectiveBack_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (16bit unsigned integer, four planes).*

- `NppStatus nppiWarpPerspectiveQuad_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, single channel).*

- `NppStatus nppiWarpPerspectiveQuad_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, three channels).*

- `NppStatus nppiWarpPerspectiveQuad_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four channels).*

- `NppStatus nppiWarpPerspectiveQuad_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveQuad_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, three planes).*

- `NppStatus nppiWarpPerspectiveQuad_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp16u` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (16bit unsigned integer, four planes).*

- `NppStatus nppiWarpPerspective_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, single channel).*

- **NppStatus** **nppiWarpPerspective\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, three channels).*

- **NppStatus** **nppiWarpPerspective\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, four channels).*

- **NppStatus** **nppiWarpPerspective\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, four channels RGBA).*

- **NppStatus** **nppiWarpPerspective\_32f\_P3R** (const **Npp32f** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, three planes).*

- **NppStatus** **nppiWarpPerspective\_32f\_P4R** (const **Npp32f** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst[4], int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit float, four planes).*

- **NppStatus** **nppiWarpPerspectiveBack\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, single channel).*

- **NppStatus** **nppiWarpPerspectiveBack\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, three channels).*

- **NppStatus** **nppiWarpPerspectiveBack\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, four channels).*

- **NppStatus** **nppiWarpPerspectiveBack\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst, int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, four channels RGBA).*

- **NppStatus** **nppiWarpPerspectiveBack\_32f\_P3R** (const **Npp32f** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** srcRoi, **Npp32f** \*pDst[3], int nDstStep, **NppiRect** dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, three planes).*

- `NppStatus nppiWarpPerspectiveBack_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit float, four planes).*

- `NppStatus nppiWarpPerspectiveQuad_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, single channel).*

- `NppStatus nppiWarpPerspectiveQuad_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, three channels).*

- `NppStatus nppiWarpPerspectiveQuad_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, four channels).*

- `NppStatus nppiWarpPerspectiveQuad_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveQuad_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, three planes).*

- `NppStatus nppiWarpPerspectiveQuad_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32f` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit float, four planes).*

- `NppStatus nppiWarpPerspective_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, single channel).*

- `NppStatus nppiWarpPerspective_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, three channels).*

- `NppStatus nppiWarpPerspective_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four channels).*

- `NppStatus nppiWarpPerspective_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four channels RGBA).*

- `NppStatus nppiWarpPerspective_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, three planes).*

- `NppStatus nppiWarpPerspective_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four planes).*

- `NppStatus nppiWarpPerspectiveBack_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, single channel).*

- `NppStatus nppiWarpPerspectiveBack_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, three channels).*

- `NppStatus nppiWarpPerspectiveBack_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, four channels).*

- `NppStatus nppiWarpPerspectiveBack_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveBack_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, three planes).*

- `NppStatus nppiWarpPerspectiveBack_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double coeffs[3][3], int eInterpolation)

*Inverse perspective transform of an image (32bit signed integer, four planes).*

- `NppStatus nppiWarpPerspectiveQuad_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, single channel).*

- `NppStatus nppiWarpPerspectiveQuad_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, three channels).*

- `NppStatus nppiWarpPerspectiveQuad_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four channels).*

- `NppStatus nppiWarpPerspectiveQuad_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four channels RGBA).*

- `NppStatus nppiWarpPerspectiveQuad_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst[3], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, three planes).*

- `NppStatus nppiWarpPerspectiveQuad_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` srcRoi, const double srcQuad[4][2], `Npp32s` \*pDst[4], int nDstStep, `NppiRect` dstRoi, const double dstQuad[4][2], int eInterpolation)

*Perspective transform of an image (32bit signed integer, four planes).*

### 7.10.1 Detailed Description

Routines manipulating an image's geometry.

### 7.10.2 Function Documentation

#### 7.10.2.1 `NppStatus nppiGetAffineBound` (`NppiRect` srcRoi, double bound[2][2], const double coeffs[2][3])

Calculates bounding box of the affine transform projection of the given source rectangular ROI.

##### Parameters:

**srcRoi** Source ROI

**bound** Bounding box of the transformed source ROI

**coeffs** Affine transform coefficients

##### Returns:

Error codes:

- `NPP_SIZE_ERROR` Indicates an error condition if any image dimension has zero or negative value
- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

### 7.10.2.2 NppStatus nppiGetAffineQuad (NppiRect *srcRoi*, double *quad*[4][2], const double *coeffs*[2][3])

Calculates affine transform projection of given source rectangular ROI.

#### Parameters:

*srcRoi* Source ROI  
*quad* Destination quadrangle  
*coeffs* Affine transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

### 7.10.2.3 NppStatus nppiGetAffineTransform (NppiRect *srcRoi*, const double *quad*[4][2], double *coeffs*[2][3])

Calculates affine transform coefficients given source rectangular ROI and its destination quadrangle projection.

#### Parameters:

*srcRoi* Source ROI  
*quad* Destination quadrangle  
*coeffs* Affine transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_AFFINE\\_QUAD\\_INCORRECT\\_WARNING](#) Indicates a warning when *quad* does not conform to the transform properties. Fourth vertex is ignored, internally computed coordinates are used instead

### 7.10.2.4 NppStatus nppiGetPerspectiveBound (NppiRect *srcRoi*, double *bound*[2][2], const double *coeffs*[3][3])

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

**Parameters:**

*srcRoi* Source ROI  
*bound* Bounding box of the transformed source ROI  
*coeffs* Perspective transform coefficients

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

**7.10.2.5 NppStatus nppiGetPerspectiveQuad (NppiRect *srcRoi*, double *quad*[4][2], const double *coeffs*[3][3])**

Calculates perspective transform projection of given source rectangular ROI.

**Parameters:**

*srcRoi* Source ROI  
*quad* Destination quadrangle  
*coeffs* Perspective transform coefficients

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

**7.10.2.6 NppStatus nppiGetPerspectiveTransform (NppiRect *srcRoi*, const double *quad*[4][2], double *coeffs*[3][3])**

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

**Parameters:**

*srcRoi* Source ROI  
*quad* Destination quadrangle  
*coeffs* Perspective transform coefficients

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

#### 7.10.2.7 NppStatus nppiGetRotateBound (NppiRect oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)

Compute bounding-box of rotated image.

##### Parameters:

*oSrcROI* Region-of-interest of the source image.

*aBoundingBox* Two 2D points representing the bounding-box of the rotated image. All four points from nppiGetRotateQuad are contained inside the axis-aligned rectangle spanned by the two points of this bounding box.

*nAngle* The rotation angle.

*nShiftX* Post-rotation shift in x-direction.

*nShiftY* Post-rotation shift in y-direction.

##### Returns:

[ROI Related Error Codes.](#)

#### 7.10.2.8 NppStatus nppiGetRotateQuad (NppiRect oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)

Compute shape of rotated image.

##### Parameters:

*oSrcROI* Region-of-interest of the source image.

*aQuad* Array of 2D points. These points are the locations of the corners of the rotated ROI.

*nAngle* The rotation nAngle.

*nShiftX* Post-rotation shift in x-direction

*nShiftY* Post-rotation shift in y-direction

##### Returns:

[ROI Related Error Codes.](#)

#### 7.10.2.9 NppStatus nppiMirror\_16u\_AC4R (const Npp16u \*pSrc, int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oROI, NppiAxis flip)

4 channel 16-bit unsigned image mirror not affecting alpha.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.  
*oROI* Region-of-Interest (ROI).  
*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes  
 • **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.10 **NppStatus nppiMirror\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oROI* Region-of-Interest (ROI).  
*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes  
 • **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.11 **NppStatus nppiMirror\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oROI* Region-of-Interest (ROI).  
*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes  
 • **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.12 NppStatus nppiMirror\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit unsigned image mirror.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.13 NppStatus nppiMirror\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float image mirror not affecting alpha.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.14 NppStatus nppiMirror\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

32-bit float image mirror.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

**7.10.2.15 NppStatus nppiMirror\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 32-bit float image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

**7.10.2.16 NppStatus nppiMirror\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit float image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

#### 7.10.2.17 NppStatus nppiMirror\_32s\_AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit image mirror not affecting alpha.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.18 NppStatus nppiMirror\_32s\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

32-bit image mirror.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

#### 7.10.2.19 NppStatus nppiMirror\_32s\_C3R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit image mirror.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

**7.10.2.20 NppStatus nppiMirror\_32s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

**7.10.2.21 NppStatus nppiMirror\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 8-bit unsigned image mirror not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

### 7.10.2.22 NppStatus nppiMirror\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

8-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

### 7.10.2.23 NppStatus nppiMirror\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 8-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_MIRROR\_FLIP\_ERR** if flip has an illegal value.

### 7.10.2.24 NppStatus nppiMirror\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* [Region-of-Interest \(ROI\)](#).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_MIRROR\\_FLIP\\_ERR](#) if flip has an illegal value.

#### 7.10.2.25 `NppStatus nppiResize_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double xFactor, double yFactor, int eInterpolation)`

8-bit unsigned image resize.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image

*xFactor* Factors by which x dimension is changed

*yFactor* Factors by which y dimension is changed

*eInterpolation* The type of eInterpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) indicates an error condition if srcROIRect has no intersection with the source image.
- [NPP\\_RESIZE\\_NO\\_OPERATION\\_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP\\_RESIZE\\_FACTOR\\_ERROR](#) Indicates an error condition if either xFactor or yFactor is less than or equal to zero.
- [NPP\\_INTERPOLATION\\_ERROR](#) if eInterpolation has an illegal value.

#### 7.10.2.26 `NppStatus nppiResize_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double xFactor, double yFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image

*xFactor* Factors by which x dimension is changed

*yFactor* Factors by which y dimension is changed

*eInterpolation* The type of interpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) indicates an error condition if srcROIRect has no intersection with the source image.
- [NPP\\_RESIZE\\_NO\\_OPERATION\\_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP\\_RESIZE\\_FACTOR\\_ERROR](#) Indicates an error condition if either xFactor or yFactor is less than or equal to zero.
- [NPP\\_INTERPOLATION\\_ERROR](#) if eInterpolation has an illegal value.

**7.10.2.27** `NppStatus nppiRotate_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)



**7.10.2.28** `NppStatus nppiRotate_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

16-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Rotate specific Error Codes

**7.10.2.29** `NppStatus nppiRotate_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 16-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Rotate specific Error Codes

**7.10.2.30** `NppStatus nppiRotate_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.31** `NppStatus nppiRotate_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.32** `NppStatus nppiRotate_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

32-bit float image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.33** `NppStatus nppiRotate_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 32-bit float image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

#### 7.10.2.34 NppStatus nppiRotate\_32f\_C4R (const Npp32f \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp32f \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nAngle*, double *nShiftX*, double *nShiftY*, int *eInterpolation*)

4 channel 32-bit float image rotate.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

#### 7.10.2.35 NppStatus nppiRotate\_8u\_AC4R (const Npp8u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp8u \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nAngle*, double *nShiftX*, double *nShiftY*, int *eInterpolation*)

4 channel 8-bit unsigned image rotate ignoring alpha channel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.36** `NppStatus nppiRotate_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.37** `NppStatus nppiRotate_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.38** `NppStatus nppiRotate_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate specific Error Codes](#)

**7.10.2.39** `NppStatus nppiWarpAffine_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four channels RGBA).

**See also:**

[nppiWarpAffine\\_16u\\_C1R](#)

**7.10.2.40** `NppStatus nppiWarpAffine_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. The function operates on source and destination regions of interest. The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = C_{00} * x + C_{01} * y + C_{02} \quad Y_{new} = C_{10} * x + C_{11} * y + C_{12}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if *srcRoi* has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if *eInterpolation* has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.41** `NppStatus nppiWarpAffine_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, three channels).

#### See also:

[nppiWarpAffine\\_16u\\_C1R](#)

**7.10.2.42** `NppStatus nppiWarpAffine_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four channels).

See also:

[nppiWarpAffine\\_16u\\_C1R](#)

**7.10.2.43** `NppStatus nppiWarpAffine_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpAffine\\_16u\\_C1R](#)

**7.10.2.44** `NppStatus nppiWarpAffine_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpAffine\\_16u\\_C1R](#)

**7.10.2.45** `NppStatus nppiWarpAffine_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpAffine\\_32f\\_C1R](#)

**7.10.2.46** `NppStatus nppiWarpAffine_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. The function operates on source and destination regions of interest.



The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = C_{00} * x + C_{01} * y + C_{02} \quad Y_{new} = C_{10} * x + C_{11} * y + C_{12}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if `eInterpolation` has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.47** `NppStatus nppiWarpAffine_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, three channels).

#### See also:

[nppiWarpAffine\\_32f\\_C1R](#)

**7.10.2.48** `NppStatus nppiWarpAffine_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, four channels).

See also:

[nppiWarpAffine\\_32f\\_C1R](#)

**7.10.2.49** `NppStatus nppiWarpAffine_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, three planes).

See also:

[nppiWarpAffine\\_32f\\_C1R](#)

**7.10.2.50** `NppStatus nppiWarpAffine_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit float, four planes).

See also:

[nppiWarpAffine\\_32f\\_C1R](#)

**7.10.2.51** `NppStatus nppiWarpAffine_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpAffine\\_32s\\_C1R](#)

**7.10.2.52** `NppStatus nppiWarpAffine_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit signed integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. The function operates on source and destination regions of interest.

The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = C_{00} * x + C_{01} * y + C_{02} \quad Y_{new} = C_{10} * x + C_{11} * y + C_{12}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if `eInterpolation` has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.53** `NppStatus nppiWarpAffine_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int interpolation)`

Affine transform of an image (32bit signed integer, three channels).

#### See also:

[nppiWarpAffine\\_32s\\_C1R](#)

**7.10.2.54** `NppStatus nppiWarpAffine_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit signed integer, four channels).

See also:

[nppiWarpAffine\\_32s\\_C1R](#)

**7.10.2.55** `NppStatus nppiWarpAffine_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpAffine\\_32s\\_C1R](#)

**7.10.2.56** `NppStatus nppiWarpAffine_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpAffine\\_32s\\_C1R](#)

**7.10.2.57** `NppStatus nppiWarpAffine_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpAffine\\_8u\\_C1R](#)

**7.10.2.58** `NppStatus nppiWarpAffine_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. The function operates on source and destination regions of interest.

The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = C_{00} * x + C_{01} * y + C_{02} \quad Y_{new} = C_{10} * x + C_{11} * y + C_{12}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

**NPPI specific recommendation:** The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if interpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.59** `NppStatus nppiWarpAffine_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, three channels).

See also:

[nppiWarpAffine\\_8u\\_C1R](#)

**7.10.2.60** `NppStatus nppiWarpAffine_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four channels).

See also:

[nppiWarpAffine\\_8u\\_C1R](#)

**7.10.2.61** `NppStatus nppiWarpAffine_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, three planes).

See also:

[nppiWarpAffine\\_8u\\_C1R](#)

**7.10.2.62** `NppStatus nppiWarpAffine_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpAffine\\_8u\\_C1R](#)

**7.10.2.63** `NppStatus nppiWarpAffineBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (16bit unsigned integer, four channels RGBA).

See also:

[nppiWarpAffineBack\\_16u\\_C1R](#)

### 7.10.2.64 NppStatus nppiWarpAffineBack\_16u\_C1R (const Npp16u \*pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u \*pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)

Inverse affine transform of an image (16bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using nppiGetAffineTransform function or set explicitly. Thus there is no need to invert coefficients in your application before calling WarpAffineBack. The function operates on source and destination regions of interest. The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$C_{00} * X_{new} + C_{01} * Y_{new} + C_{02} = x \quad C_{10} * X_{new} + C_{11} * Y_{new} + C_{12} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions nppiGetAffineQuad and nppiGetAffineBound can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but doesn't perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values  $(\text{int})(\text{void} *) (\text{pDst} + \text{dstRoi.x})$  and  $(\text{int})(\text{void} *) (\text{pDst} + \text{dstRoi.x} + \text{dstRoi.width})$  are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return NPP\_MISALIGNED\_DST\_ROI\_WARNING warning.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*srcRoi* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.65** `NppStatus nppiWarpAffineBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (16bit unsigned integer, three channels).

See also:

[nppiWarpAffineBack\\_16u\\_C1R](#)

**7.10.2.66** `NppStatus nppiWarpAffineBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (16bit unsigned integer, four channels).

See also:

[nppiWarpAffineBack\\_16u\\_C1R](#)

**7.10.2.67** `NppStatus nppiWarpAffineBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpAffineBack\\_16u\\_C1R](#)

**7.10.2.68** `NppStatus nppiWarpAffineBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpAffineBack\\_16u\\_C1R](#)



**7.10.2.69** `NppStatus nppiWarpAffineBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpAffineBack\\_32f\\_C1R](#)

**7.10.2.70** `NppStatus nppiWarpAffineBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpAffineBack`. The function operates on source and destination regions of interest. The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$C_{00} * X_{new} + C_{01} * Y_{new} + C_{02} = x \quad C_{10} * X_{new} + C_{11} * Y_{new} + C_{12} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if `eInterpolation` has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.71** `NppStatus nppiWarpAffineBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, three channels).

See also:

[nppiWarpAffineBack\\_32f\\_C1R](#)

**7.10.2.72** `NppStatus nppiWarpAffineBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, four channels).

See also:

[nppiWarpAffineBack\\_32f\\_C1R](#)

**7.10.2.73** `NppStatus nppiWarpAffineBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, three planes).

See also:

[nppiWarpAffineBack\\_32f\\_C1R](#)

**7.10.2.74** `NppStatus nppiWarpAffineBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit float, four planes).

See also:

[nppiWarpAffineBack\\_32f\\_C1R](#)

**7.10.2.75** `NppStatus nppiWarpAffineBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpAffineBack\\_32s\\_C1R](#)

**7.10.2.76** `NppStatus nppiWarpAffineBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpAffineBack`. The function operates on source and destination regions of interest. The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$C_{00} * X_{new} + C_{01} * Y_{new} + C_{02} = x \quad C_{10} * X_{new} + C_{11} * Y_{new} + C_{12} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*srcRoi* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if *srcRoi* has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if *eInterpolation* has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.77** `NppStatus nppiWarpAffineBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, three channels).

See also:

[nppiWarpAffineBack\\_32s\\_C1R](#)

**7.10.2.78** `NppStatus nppiWarpAffineBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, four channels).

See also:

[nppiWarpAffineBack\\_32s\\_C1R](#)

**7.10.2.79** `NppStatus nppiWarpAffineBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpAffineBack\\_32s\\_C1R](#)

**7.10.2.80** `NppStatus nppiWarpAffineBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpAffineBack\\_32s\\_C1R](#)

**7.10.2.81** `NppStatus nppiWarpAffineBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpAffineBack\\_8u\\_C1R](#)

**7.10.2.82** `NppStatus nppiWarpAffineBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetAffineTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpAffineBack`. The function operates on source and destination regions of interest. The affine warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$C_{00} * X_{new} + C_{01} * Y_{new} + C_{02} = x \quad C_{10} * X_{new} + C_{11} * Y_{new} + C_{12} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetAffineQuad` and `nppiGetAffineBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but doesn't perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void *) (pDst + dstRoi.x))` and `(int)((void *) (pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if interpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.83** `NppStatus nppiWarpAffineBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, three channels).

**See also:**

[nppiWarpAffineBack\\_8u\\_C1R](#)

**7.10.2.84** `NppStatus nppiWarpAffineBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, four channels).

**See also:**

[nppiWarpAffineBack\\_8u\\_C1R](#)

**7.10.2.85** `NppStatus nppiWarpAffineBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, three planes).

**See also:**

[nppiWarpAffineBack\\_8u\\_C1R](#)

**7.10.2.86** `NppStatus nppiWarpAffineBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[2][3], int eInterpolation)`

Inverse affine transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpAffineBack\\_8u\\_C1R](#)

**7.10.2.87** `NppStatus nppiWarpAffineQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four channels RGBA).

See also:

[nppiWarpAffineQuad\\_16u\\_C1R](#)

**7.10.2.88** `NppStatus nppiWarpAffineQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, single channel).

This function performs affine warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpAffineQuad` uses the same formulas for pixel mapping as in `nppiWarpAffine` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but doesn't perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void *) (pDst + dstRoi.x))` and `(int)((void *) (pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes in each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*srcRoi* Source ROI  
*srcQuad* Source quadrangle  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.89** `NppStatus nppiWarpAffineQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, three channels).

See also:

[nppiWarpAffineQuad\\_16u\\_C1R](#)

**7.10.2.90** `NppStatus nppiWarpAffineQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four channels).

See also:

[nppiWarpAffineQuad\\_16u\\_C1R](#)

**7.10.2.91** `NppStatus nppiWarpAffineQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpAffineQuad\\_16u\\_C1R](#)



**7.10.2.92** `NppStatus nppiWarpAffineQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpAffineQuad\\_16u\\_C1R](#)

**7.10.2.93** `NppStatus nppiWarpAffineQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpAffineQuad\\_32f\\_C1R](#)

**7.10.2.94** `NppStatus nppiWarpAffineQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, single channel).

This function performs affine warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpAffineQuad` uses the same formulas for pixel mapping as in `nppiWarpAffine` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*srcQuad* Source quadrangle

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.95** `NppStatus nppiWarpAffineQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, three channels).

See also:

[nppiWarpAffineQuad\\_32f\\_C1R](#)

**7.10.2.96** `NppStatus nppiWarpAffineQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, four channels).

See also:

[nppiWarpAffineQuad\\_32f\\_C1R](#)

**7.10.2.97** `NppStatus nppiWarpAffineQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, three planes).

See also:

[nppiWarpAffineQuad\\_32f\\_C1R](#)

**7.10.2.98** `NppStatus nppiWarpAffineQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit float, four planes).

See also:

[nppiWarpAffineQuad\\_32f\\_C1R](#)

**7.10.2.99** `NppStatus nppiWarpAffineQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpAffineQuad\\_32s\\_C1R](#)

**7.10.2.100** `NppStatus nppiWarpAffineQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, single channel).

This function performs affine warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpAffineQuad` uses the same formulas for pixel mapping as in `nppiWarpAffine` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*srcQuad* Source quadrangle

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if `eInterpolation` has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment

**7.10.2.101** `NppStatus nppiWarpAffineQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, three channels).

See also:

[nppiWarpAffineQuad\\_32s\\_C1R](#)

**7.10.2.102** `NppStatus nppiWarpAffineQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, four channels).

See also:

[nppiWarpAffineQuad\\_32s\\_C1R](#)

**7.10.2.103** `NppStatus nppiWarpAffineQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpAffineQuad\\_32s\\_C1R](#)

**7.10.2.104** `NppStatus nppiWarpAffineQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpAffineQuad\\_32s\\_C1R](#)

**7.10.2.105** `NppStatus nppiWarpAffineQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpAffineQuad\\_8u\\_C1R](#)

### 7.10.2.106 **NppStatus nppiWarpAffineQuad\_8u\_C1R** (const Npp8u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *srcRoi*, const double *srcQuad*[4][2], Npp8u \* *pDst*, int *nDstStep*, NppiRect *dstRoi*, const double *dstQuad*[4][2], int *eInterpolation*)

Affine transform of an image (8bit unsigned integer, single channel).

This function performs affine warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpAffineQuad` uses the same formulas for pixel mapping as in `nppiWarpAffine` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified `eInterpolation` method and written to the destination ROI.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*srcQuad* Source quadrangle

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if `srcRoi` has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if `eInterpolation` has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI misalignment ignored, internally computed coordinates are used instead

**7.10.2.107** `NppStatus nppiWarpAffineQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, three channels).

See also:

[nppiWarpAffineQuad\\_8u\\_C1R](#)

**7.10.2.108** `NppStatus nppiWarpAffineQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four channels).

See also:

[nppiWarpAffineQuad\\_8u\\_C1R](#)

**7.10.2.109** `NppStatus nppiWarpAffineQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, three planes).

See also:

[nppiWarpAffineQuad\\_8u\\_C1R](#)

**7.10.2.110** `NppStatus nppiWarpAffineQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Affine transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpAffineQuad\\_8u\\_C1R](#)

**7.10.2.111** `NppStatus nppiWarpPerspective_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspective\\_16u\\_C1R](#)

### 7.10.2.112 **NppStatus nppiWarpPerspective\_16u\_C1R** (const Npp16u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *srcRoi*, Npp16u \* *pDst*, int *nDstStep*, NppiRect *dstRoi*, const double *coeffs*[3][3], int *eInterpolation*)

Perspective transform of an image (16bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates ( $x, y$ ) according to the following formulas:

$$X_{new} = \frac{C_{00} * x + C_{01} * y + C_{02}}{C_{20} * x + C_{21} * y + C_{22}} \quad Y_{new} = \frac{C_{10} * x + C_{11} * y + C_{12}}{C_{20} * x + C_{21} * y + C_{22}}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*srcRoi* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if *srcRoi* has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if *eInterpolation* has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.113** `NppStatus nppiWarpPerspective_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, three channels).

See also:

[nppiWarpPerspective\\_16u\\_C1R](#)

**7.10.2.114** `NppStatus nppiWarpPerspective_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four channels).

See also:

[nppiWarpPerspective\\_16u\\_C1R](#)

**7.10.2.115** `NppStatus nppiWarpPerspective_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpPerspective\\_16u\\_C1R](#)

**7.10.2.116** `NppStatus nppiWarpPerspective_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpPerspective\\_16u\\_C1R](#)



**7.10.2.117** `NppStatus nppiWarpPerspective_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpPerspective\\_32f\\_C1R](#)

**7.10.2.118** `NppStatus nppiWarpPerspective_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = \frac{C_{00} * x + C_{01} * y + C_{02}}{C_{20} * x + C_{21} * y + C_{22}} \quad Y_{new} = \frac{C_{10} * x + C_{11} * y + C_{12}}{C_{20} * x + C_{21} * y + C_{22}}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `srcRoi` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if `eInterpolation` has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.119** `NppStatus nppiWarpPerspective_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, three channels).

See also:

[nppiWarpPerspective\\_32f\\_C1R](#)

**7.10.2.120** `NppStatus nppiWarpPerspective_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, four channels).

See also:

[nppiWarpPerspective\\_32f\\_C1R](#)

**7.10.2.121** `NppStatus nppiWarpPerspective_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, three planes).

See also:

[nppiWarpPerspective\\_32f\\_C1R](#)

**7.10.2.122** `NppStatus nppiWarpPerspective_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit float, four planes).

See also:

[nppiWarpPerspective\\_32f\\_C1R](#)

**7.10.2.123** `NppStatus nppiWarpPerspective_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpPerspective\\_32s\\_C1R](#)

**7.10.2.124** `NppStatus nppiWarpPerspective_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = \frac{C_{00} * x + C_{01} * y + C_{02}}{C_{20} * x + C_{21} * y + C_{22}} \quad Y_{new} = \frac{C_{10} * x + C_{11} * y + C_{12}}{C_{20} * x + C_{21} * y + C_{22}}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `srcRoi` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if `eInterpolation` has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.125** `NppStatus nppiWarpPerspective_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, three channels).

See also:

[nppiWarpPerspective\\_32s\\_C1R](#)

**7.10.2.126** `NppStatus nppiWarpPerspective_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, four channels).

See also:

[nppiWarpPerspective\\_32s\\_C1R](#)

**7.10.2.127** `NppStatus nppiWarpPerspective_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpPerspective\\_32s\\_C1R](#)

**7.10.2.128** `NppStatus nppiWarpPerspective_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpPerspective\\_32s\\_C1R](#)

**7.10.2.129** `NppStatus nppiWarpPerspective_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspective\\_8u\\_C1R](#)

**7.10.2.130** `NppStatus nppiWarpPerspective_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$X_{new} = \frac{C_{00} * x + C_{01} * y + C_{02}}{C_{20} * x + C_{21} * y + C_{22}} \quad Y_{new} = \frac{C_{10} * x + C_{11} * y + C_{12}}{C_{20} * x + C_{21} * y + C_{22}}$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void *) (pDst + dstRoi.x))` and `(int)((void *) (pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*srcRoi* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.131** `NppStatus nppiWarpPerspective_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, three channels).

**See also:**

[nppiWarpPerspective\\_8u\\_C1R](#)

**7.10.2.132** `NppStatus nppiWarpPerspective_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four channels).

**See also:**

[nppiWarpPerspective\\_8u\\_C1R](#)

**7.10.2.133** `NppStatus nppiWarpPerspective_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, three planes).

**See also:**

[nppiWarpPerspective\\_8u\\_C1R](#)

**7.10.2.134** `NppStatus nppiWarpPerspective_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpPerspective\\_8u\\_C1R](#)

**7.10.2.135** `NppStatus nppiWarpPerspectiveBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (16bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspectiveBack\\_16u\\_C1R](#)

**7.10.2.136** `NppStatus nppiWarpPerspectiveBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (16bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpPerspectiveBack`. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$\frac{C_{00} * X_{new} + C_{01} * Y_{new} + C_{02}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = x \quad \frac{C_{10} * X_{new} + C_{11} * Y_{new} + C_{12}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void *) (pDst + dstRoi.x))` and `(int)((void *) (pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.137** `NppStatus nppiWarpPerspectiveBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (16bit unsigned integer, three channels).

#### See also:

[npippiWarpPerspectiveBack\\_16u\\_C1R](#)

**7.10.2.138** `NppStatus nppiWarpPerspectiveBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (16bit unsigned integer, four channels).

#### See also:

[npippiWarpPerspectiveBack\\_16u\\_C1R](#)



**7.10.2.139** `NppStatus nppiWarpPerspectiveBack_16u_P3R` (`const Npp16u * pSrc[3]`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `Npp16u * pDst[3]`, `int nDstStep`, `NppiRect dstRoi`, `const double coeffs[3][3]`, `int eInterpolation`)

Inverse perspective transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpPerspectiveBack\\_16u\\_C1R](#)

**7.10.2.140** `NppStatus nppiWarpPerspectiveBack_16u_P4R` (`const Npp16u * pSrc[4]`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `Npp16u * pDst[4]`, `int nDstStep`, `NppiRect dstRoi`, `const double coeffs[3][3]`, `int eInterpolation`)

Inverse perspective transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpPerspectiveBack\\_16u\\_C1R](#)

**7.10.2.141** `NppStatus nppiWarpPerspectiveBack_32f_AC4R` (`const Npp32f * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `Npp32f * pDst`, `int nDstStep`, `NppiRect dstRoi`, `const double coeffs[3][3]`, `int eInterpolation`)

Inverse perspective transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpPerspectiveBack\\_32f\\_C1R](#)

**7.10.2.142** `NppStatus nppiWarpPerspectiveBack_32f_C1R` (`const Npp32f * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `Npp32f * pDst`, `int nDstStep`, `NppiRect dstRoi`, `const double coeffs[3][3]`, `int eInterpolation`)

Inverse perspective transform of an image (32bit float, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpPerspectiveBack`. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$\frac{C_{00} * X_{new} + C_{01} * Y_{new} + C_{02}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = x \quad \frac{C_{10} * X_{new} + C_{11} * Y_{new} + C_{12}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.143** `NppStatus nppiWarpPerspectiveBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit float, three channels).

#### See also:

[npippiWarpPerspectiveBack\\_32f\\_C1R](#)

**7.10.2.144** `NppStatus nppiWarpPerspectiveBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit float, four channels).

#### See also:

[npippiWarpPerspectiveBack\\_32f\\_C1R](#)

**7.10.2.145** `NppStatus nppiWarpPerspectiveBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit float, three planes).

See also:

[nppiWarpPerspectiveBack\\_32f\\_C1R](#)

**7.10.2.146** `NppStatus nppiWarpPerspectiveBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32f * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit float, four planes).

See also:

[nppiWarpPerspectiveBack\\_32f\\_C1R](#)

**7.10.2.147** `NppStatus nppiWarpPerspectiveBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpPerspectiveBack\\_32s\\_C1R](#)

**7.10.2.148** `NppStatus nppiWarpPerspectiveBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpPerspectiveBack`. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$\frac{C_{00} * X_{new} + C_{01} * Y_{new} + C_{02}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = x \quad \frac{C_{10} * X_{new} + C_{11} * Y_{new} + C_{12}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.149** `NppStatus nppiWarpPerspectiveBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, three channels).

#### See also:

[npippiWarpPerspectiveBack\\_32s\\_C1R](#)

**7.10.2.150** `NppStatus nppiWarpPerspectiveBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, four channels).

#### See also:

[npippiWarpPerspectiveBack\\_32s\\_C1R](#)

**7.10.2.151** `NppStatus nppiWarpPerspectiveBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpPerspectiveBack\\_32s\\_C1R](#)

**7.10.2.152** `NppStatus nppiWarpPerspectiveBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpPerspectiveBack\\_32s\\_C1R](#)

**7.10.2.153** `NppStatus nppiWarpPerspectiveBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspectiveBack\\_8u\\_C1R](#)

**7.10.2.154** `NppStatus nppiWarpPerspectiveBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, single channel).

This function operates using given transform coefficients that can be obtained by using `nppiGetPerspectiveTransform` function or set explicitly. Thus there is no need to invert coefficients in your application before calling `WarpPerspectiveBack`. The function operates on source and destination regions of interest. The perspective warp function transforms the source image pixel coordinates  $(x, y)$  according to the following formulas:

$$\frac{C_{00} * X_{new} + C_{01} * Y_{new} + C_{02}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = x \quad \frac{C_{10} * X_{new} + C_{11} * Y_{new} + C_{12}}{C_{20} * X_{new} + C_{21} * Y_{new} + C_{22}} = y$$

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI. The functions `nppiGetPerspectiveQuad` and `nppiGetPerspectiveBound` can help with destination ROI specification.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical

stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*coeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if *srcRoi* has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if *eInterpolation* has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.155** `NppStatus nppiWarpPerspectiveBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, three channels).

#### See also:

[nppiWarpPerspectiveBack\\_8u\\_C1R](#)

**7.10.2.156** `NppStatus nppiWarpPerspectiveBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, four channels).

See also:

[nppiWarpPerspectiveBack\\_8u\\_C1R](#)

**7.10.2.157** `NppStatus nppiWarpPerspectiveBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, three planes).

See also:

[nppiWarpPerspectiveBack\\_8u\\_C1R](#)

**7.10.2.158** `NppStatus nppiWarpPerspectiveBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double coeffs[3][3], int eInterpolation)`

Inverse perspective transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpPerspectiveBack\\_8u\\_C1R](#)

**7.10.2.159** `NppStatus nppiWarpPerspectiveQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspectiveQuad\\_16u\\_C1R](#)

**7.10.2.160** `NppStatus nppiWarpPerspectiveQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, single channel).

This function performs perspective warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpPerspectiveQuad` uses the same formulas for pixel mapping as in `nppiWarpPerspective` function. The transform coefficients are computed internally.

The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void*)(pDst + dstRoi.x))` and `(int)((void*)(pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*srcRoi* Source ROI

*srcQuad* Source quadrangle

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *srcRoi* and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if *srcRoi* has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if *eInterpolation* has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.161** `NppStatus nppiWarpPerspectiveQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, three channels).



See also:

[nppiWarpPerspectiveQuad\\_16u\\_C1R](#)

**7.10.2.162** `NppStatus nppiWarpPerspectiveQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four channels).

See also:

[nppiWarpPerspectiveQuad\\_16u\\_C1R](#)

**7.10.2.163** `NppStatus nppiWarpPerspectiveQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, three planes).

See also:

[nppiWarpPerspectiveQuad\\_16u\\_C1R](#)

**7.10.2.164** `NppStatus nppiWarpPerspectiveQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (16bit unsigned integer, four planes).

See also:

[nppiWarpPerspectiveQuad\\_16u\\_C1R](#)

**7.10.2.165** `NppStatus nppiWarpPerspectiveQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit float, four channels RGBA).

See also:

[nppiWarpPerspectiveQuad\\_32f\\_C1R](#)

**7.10.2.166** `NppStatus nppiWarpPerspectiveQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit float, single channel).

This function performs perspective warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpPerspectiveQuad` uses the same formulas for pixel mapping as in `nppiWarpPerspective` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*srcRoi* Source ROI

*srcQuad* Source quadrangle

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*dstRoi* Destination ROI

*dstQuad* Destination quadrangle

*eInterpolation* Interpolation mode: can be `NPPI_INTER_NN`, `NPPI_INTER_LINEAR` or `NPPI_INTER_CUBIC`

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `srcRoi` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `srcRoi` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if `eInterpolation` has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- `NPP_MISALIGNED_DST_ROI_WARNING` Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.167** `NppStatus nppiWarpPerspectiveQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit float, three channels).

**See also:**

[nppiWarpPerspectiveQuad\\_32f\\_C1R](#)

**7.10.2.168** `NppStatus nppiWarpPerspectiveQuad_32f_C4R` (`const Npp32f * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `const double srcQuad[4][2]`, `Npp32f * pDst`, `int nDstStep`, `NppiRect dstRoi`, `const double dstQuad[4][2]`, `int eInterpolation`)

Perspective transform of an image (32bit float, four channels).

See also:

[nppiWarpPerspectiveQuad\\_32f\\_C1R](#)

**7.10.2.169** `NppStatus nppiWarpPerspectiveQuad_32f_P3R` (`const Npp32f * pSrc[3]`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `const double srcQuad[4][2]`, `Npp32f * pDst[3]`, `int nDstStep`, `NppiRect dstRoi`, `const double dstQuad[4][2]`, `int eInterpolation`)

Perspective transform of an image (32bit float, three planes).

See also:

[nppiWarpPerspectiveQuad\\_32f\\_C1R](#)

**7.10.2.170** `NppStatus nppiWarpPerspectiveQuad_32f_P4R` (`const Npp32f * pSrc[4]`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `const double srcQuad[4][2]`, `Npp32f * pDst[4]`, `int nDstStep`, `NppiRect dstRoi`, `const double dstQuad[4][2]`, `int eInterpolation`)

Perspective transform of an image (32bit float, four planes).

See also:

[nppiWarpPerspectiveQuad\\_32f\\_C1R](#)

**7.10.2.171** `NppStatus nppiWarpPerspectiveQuad_32s_AC4R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `const double srcQuad[4][2]`, `Npp32s * pDst`, `int nDstStep`, `NppiRect dstRoi`, `const double dstQuad[4][2]`, `int eInterpolation`)

Perspective transform of an image (32bit signed integer, four channels RGBA).

See also:

[nppiWarpPerspectiveQuad\\_32s\\_C1R](#)

**7.10.2.172** `NppStatus nppiWarpPerspectiveQuad_32s_C1R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect srcRoi`, `const double srcQuad[4][2]`, `Npp32s * pDst`, `int nDstStep`, `NppiRect dstRoi`, `const double dstQuad[4][2]`, `int eInterpolation`)

Perspective transform of an image (32bit signed integer, single channel).

This function performs perspective warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpPerspectiveQuad` uses the same formulas for pixel mapping as in `nppiWarpPerspective` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*srcRoi* Source ROI  
*srcQuad* Source quadrangle  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstRoi* Destination ROI  
*dstQuad* Destination quadrangle  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.173** `NppStatus nppiWarpPerspectiveQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit signed integer, three channels).

**See also:**

[npippiWarpPerspectiveQuad\\_32s\\_C1R](#)

**7.10.2.174** `NppStatus nppiWarpPerspectiveQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit signed integer, four channels).

**See also:**

[npippiWarpPerspectiveQuad\\_32s\\_C1R](#)

**7.10.2.175** `NppStatus nppiWarpPerspectiveQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit signed integer, three planes).

See also:

[nppiWarpPerspectiveQuad\\_32s\\_C1R](#)

**7.10.2.176** `NppStatus nppiWarpPerspectiveQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (32bit signed integer, four planes).

See also:

[nppiWarpPerspectiveQuad\\_32s\\_C1R](#)

**7.10.2.177** `NppStatus nppiWarpPerspectiveQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four channels RGBA).

See also:

[nppiWarpPerspectiveQuad\\_8u\\_C1R](#)

**7.10.2.178** `NppStatus nppiWarpPerspectiveQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, single channel).

This function performs perspective warping of a the specified quadrangle in the source image to the specified quadrangle in the destination image. The function `nppiWarpPerspectiveQuad` uses the same formulas for pixel mapping as in `nppiWarpPerspective` function. The transform coefficients are computed internally. The transformed part of the source image is resampled using the specified interpolation method and written to the destination ROI.

NPPI specific recommendation: The function operates using 2 types of kernels: fast and accurate. The fast method is about 4 times faster than its accurate variant, but does not perform memory access checks and requires the destination ROI to be 64 bytes aligned. Hence any destination ROI is chunked into 3 vertical stripes: the first and the third are processed by accurate kernels and the central one is processed by the fast one. In order to get the maximum available speed of execution, the projection of destination ROI onto image addresses must be 64 bytes aligned. This is always true if the values `(int)((void *) (pDst + dstRoi.x))` and `(int)((void *) (pDst + dstRoi.x + dstRoi.width))` are multiples of 64. Another rule of thumb is to specify destination ROI in such way that left and right sides of the projected image are separated from the ROI by at least 63 bytes from each side. However, this requires the whole ROI to be part of allocated memory. In case when the conditions above are not satisfied, the function may decrease in speed slightly and will return `NPP_MISALIGNED_DST_ROI_WARNING` warning.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*srcRoi* Source ROI  
*srcQuad* Source quadrangle  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstRoi* Destination ROI  
*dstQuad* Destination quadrangle  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the srcRoi and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if srcRoi has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if eInterpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI
- [NPP\\_MISALIGNED\\_DST\\_ROI\\_WARNING](#) Indicates a warning that the speed of primitive execution was reduced due to destination ROI

**7.10.2.179** `NppStatus nppiWarpPerspectiveQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, three channels).

**See also:**

[npippiWarpPerspectiveQuad\\_8u\\_C1R](#)

**7.10.2.180** `NppStatus nppiWarpPerspectiveQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four channels).

**See also:**

[npippiWarpPerspectiveQuad\\_8u\\_C1R](#)

**7.10.2.181** `NppStatus nppiWarpPerspectiveQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, three planes).

See also:

[nppiWarpPerspectiveQuad\\_8u\\_C1R](#)

**7.10.2.182** `NppStatus nppiWarpPerspectiveQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect srcRoi, const double srcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect dstRoi, const double dstQuad[4][2], int eInterpolation)`

Perspective transform of an image (8bit unsigned integer, four planes).

See also:

[nppiWarpPerspectiveQuad\\_8u\\_C1R](#)

## 7.11 NPP Signal Processing

### Memory Allocation

Signal-allocator methods for allocating 1D arrays of data in device memory.

All allocators have size parameters to specify the size of the signal (1D array) being allocated.

The allocator methods return a pointer to the newly allocated memory of appropriate type. If device-memory allocation is not possible due to resource constraints the allocators return 0 (i.e. NULL pointer).

All signal allocators allocate memory aligned such that it is beneficial to the performance of the majority of the signal-processing primitives. It is no mandatory however to use these allocators. Any valid CUDA device-memory pointers can be passed to NPP primitives.

- `Npp8u * nppsMalloc_8u` (int nSize)  
*8-bit unsigned signal allocator.*
- `Npp16u * nppsMalloc_16u` (int nSize)  
*16-bit unsigned signal allocator.*
- `Npp16s * nppsMalloc_16s` (int nSize)  
*16-bit signal allocator.*
- `Npp16sc * nppsMalloc_16sc` (int nSize)  
*16-bit complex-value signal allocator.*
- `Npp32u * nppsMalloc_32u` (int nSize)  
*32-bit unsigned signal allocator.*
- `Npp32s * nppsMalloc_32s` (int nSize)  
*32-bit integer signal allocator.*
- `Npp32sc * nppsMalloc_32sc` (int nSize)  
*32-bit complex integer signal allocator.*
- `Npp32f * nppsMalloc_32f` (int nSize)  
*32-bit float signal allocator.*
- `Npp32fc * nppsMalloc_32fc` (int nSize)  
*32-bit complex float signal allocator.*
- `Npp64s * nppsMalloc_64s` (int nSize)  
*64-bit long integer signal allocator.*
- `Npp64sc * nppsMalloc_64sc` (int nSize)  
*64-bit complex long integer signal allocator.*
- `Npp64f * nppsMalloc_64f` (int nSize)  
*64-bit float (double) signal allocator.*
- `Npp64fc * nppsMalloc_64fc` (int nSize)



*64-bit complex complex signal allocator.*

- void `nppsFree` (void \*pValues)  
*Free method for any 2D allocated memory.*

## Set

Set methods for 1D vectors of various types.

The copy methods operate on vector data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to `Npp8u` type) and length of the vectors, i.e. the number of items.

- `NppStatus nppsSet_8u` (`Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char, vector set method.*
- `NppStatus nppsSet_16s` (`Npp16s` nValue, `Npp16s` \*pDst, int nLength)  
*16-bit integer, vector set method.*
- `NppStatus nppsSet_16sc` (`Npp16sc` nValue, `Npp16sc` \*pDst, int nLength)  
*16-bit integer complex, vector set method.*
- `NppStatus nppsSet_32s` (`Npp32s` nValue, `Npp32s` \*pDst, int nLength)  
*32-bit integer, vector set method.*
- `NppStatus nppsSet_32sc` (`Npp32sc` nValue, `Npp32sc` \*pDst, int nLength)  
*32-bit integer complex, vector set method.*
- `NppStatus nppsSet_32f` (`Npp32f` nValue, `Npp32f` \*pDst, int nLength)  
*32-bit float, vector set method.*
- `NppStatus nppsSet_32fc` (`Npp32fc` nValue, `Npp32fc` \*pDst, int nLength)  
*32-bit float complex, vector set method.*
- `NppStatus nppsSet_64s` (`Npp64s` nValue, `Npp64s` \*pDst, int nLength)  
*64-bit long long integer, vector set method.*
- `NppStatus nppsSet_64sc` (`Npp64sc` nValue, `Npp64sc` \*pDst, int nLength)  
*64-bit long long integer complex, vector set method.*
- `NppStatus nppsSet_64f` (`Npp64f` nValue, `Npp64f` \*pDst, int nLength)  
*64-bit double, vector set method.*
- `NppStatus nppsSet_64fc` (`Npp64fc` nValue, `Npp64fc` \*pDst, int nLength)  
*64-bit double complex, vector set method.*

## Zero

Set signals to zero.

- `NppStatus nppsZero_8u (Npp8u *pDst, int nLength)`  
*8-bit unsigned char, vector zero method.*
- `NppStatus nppsZero_16s (Npp16s *pDst, int nLength)`  
*16-bit integer, vector zero method.*
- `NppStatus nppsZero_16sc (Npp16sc *pDst, int nLength)`  
*16-bit integer complex, vector zero method.*
- `NppStatus nppsZero_32s (Npp32s *pDst, int nLength)`  
*32-bit integer, vector zero method.*
- `NppStatus nppsZero_32sc (Npp32sc *pDst, int nLength)`  
*32-bit integer complex, vector zero method.*
- `NppStatus nppsZero_32f (Npp32f *pDst, int nLength)`  
*32-bit float, vector zero method.*
- `NppStatus nppsZero_32fc (Npp32fc *pDst, int nLength)`  
*32-bit float complex, vector zero method.*
- `NppStatus nppsZero_64s (Npp64s *pDst, int nLength)`  
*64-bit long long integer, vector zero method.*
- `NppStatus nppsZero_64sc (Npp64sc *pDst, int nLength)`  
*64-bit long long integer complex, vector zero method.*
- `NppStatus nppsZero_64f (Npp64f *pDst, int nLength)`  
*64-bit double, vector zero method.*
- `NppStatus nppsZero_64fc (Npp64fc *pDst, int nLength)`  
*64-bit double complex, vector zero method.*

## Copy

Copy methods for various type signals.

Copy methods operate on signal data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to `Npp8u` type) and length of the vectors, i.e. the number of items.

- `NppStatus nppsCopy_8u (const Npp8u *pSrc, Npp8u *pDst, int len)`  
*8-bit unsigned char, vector copy method*
- `NppStatus nppsCopy_16s (const Npp16s *pSrc, Npp16s *pDst, int len)`  
*16-bit signed short, vector copy method.*

- **NppStatus nppsCopy\_32s** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer, vector copy method.*
- **NppStatus nppsCopy\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int len)  
*32-bit float, vector copy method.*
- **NppStatus nppsCopy\_64s** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int len)  
*64-bit signed integer, vector copy method.*
- **NppStatus nppsCopy\_16sc** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int len)  
*16-bit complex short, vector copy method.*
- **NppStatus nppsCopy\_32sc** (const **Npp32sc** \*pSrc, **Npp32sc** \*pDst, int len)  
*32-bit complex signed integer, vector copy method.*
- **NppStatus nppsCopy\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int len)  
*32-bit complex float, vector copy method.*
- **NppStatus nppsCopy\_64sc** (const **Npp64sc** \*pSrc, **Npp64sc** \*pDst, int len)  
*64-bit complex signed integer, vector copy method.*
- **NppStatus nppsCopy\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int len)  
*64-bit complex double, vector copy method.*

## AddC

Adds a constant value to each sample of a signal.

- **NppStatus nppsAddC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal add constant, scale, then clamp to saturated value*
- **NppStatus nppsAddC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned charvector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short vector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal add constant, scale, then clamp to saturated value.*

- **NppStatus nppsAddC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.*

- **NppStatus nppsAddC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)

*16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.*

- **NppStatus nppsAddC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal add constant and scale.*

- **NppStatus nppsAddC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integers signal add constant and scale.*

- **NppStatus nppsAddC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.*

- **NppStatus nppsAddC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.*

- **NppStatus nppsAddC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)

*32-bit floating point in place signal add constant.*

- **NppStatus nppsAddC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)

*32-bit floating point signal add constant.*

- **NppStatus nppsAddC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)

*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.*

- **NppStatus nppsAddC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)

*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.*

- **NppStatus nppsAddC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point, in place signal add constant.*

- **NppStatus nppsAddC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)

*64-bit floating point signal add constant.*

- **NppStatus nppsAddC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)

*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.*

- **NppStatus nppsAddC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)

*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.*

## AddProductC

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

- **NppStatus nppsAddProductC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value*
- **NppStatus nppsAddProductC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal add product of signal times constant to destination signal, scale, then clamp to saturated value.*
- **NppStatus nppsAddProductC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal add product of signal times constant to destination signal and scale.*
- **NppStatus nppsAddProductC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal add product of signal times constant to destination signal and scale.*

- **NppStatus nppsAddProductC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add product of signal times constant to destination signal and scale.*
- **NppStatus nppsAddProductC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal add product of signal times constant to destination signal and scale.*
- **NppStatus nppsAddProductC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point, in place signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add product of signal times constant to destination signal.*
- **NppStatus nppsAddProductC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add product of signal times constant to destination signal.*

## MulC

Multiplies each sample of a signal by a constant value.

- **NppStatus nppsMulC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal times constant, scale, then clamp to saturated value*

- **NppStatus nppsMulC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times constant and scale.*
- **NppStatus nppsMulC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.*
- **NppStatus nppsMulC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal times constant.*
- **NppStatus nppsMulC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal times constant.*

- `NppStatus nppsMulC_Low_32f16s` (const `Npp32f` \*pSrc, `Npp32f` nValue, `Npp16s` \*pDst, int nLength)  
*32-bit floating point signal times constant with output converted to 16-bit signed integer.*
- `NppStatus nppsMulC_32f16s_Sfs` (const `Npp32f` \*pSrc, `Npp32f` nValue, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.*
- `NppStatus nppsMulC_32fc_I` (`Npp32fc` nValue, `Npp32fc` \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.*
- `NppStatus nppsMulC_32fc` (const `Npp32fc` \*pSrc, `Npp32fc` nValue, `Npp32fc` \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.*
- `NppStatus nppsMulC_64f_I` (`Npp64f` nValue, `Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point, in place signal times constant.*
- `NppStatus nppsMulC_64f` (const `Npp64f` \*pSrc, `Npp64f` nValue, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal times constant.*
- `NppStatus nppsMulC_64f64s_ISfs` (`Npp64f` nValue, `Npp64s` \*pDst, int nLength, int nScaleFactor)  
*64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.*
- `NppStatus nppsMulC_64fc_I` (`Npp64fc` nValue, `Npp64fc` \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.*
- `NppStatus nppsMulC_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` nValue, `Npp64fc` \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.*

## SubC

Subtracts a constant from each sample of a signal.

- `NppStatus nppsSubC_8u_ISfs` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value*
- `NppStatus nppsSubC_8u_Sfs` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppsSubC_16u_ISfs` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppsSubC_16u_Sfs` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.*



- **NppStatus nppsSubC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.*
- **NppStatus nppsSubC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal subtract constant.*
- **NppStatus nppsSubC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract constant.*
- **NppStatus nppsSubC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.*
- **NppStatus nppsSubC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.*
- **NppStatus nppsSubC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point, in place signal subtract constant.*
- **NppStatus nppsSubC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract constant.*
- **NppStatus nppsSubC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)

*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.*

- **NppStatus nppsSubC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.*

## SubCRev

Subtracts each sample of a signal from a constant.

- **NppStatus nppsSubCRev\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value*
- **NppStatus nppsSubCRev\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubCRev\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract from constant and scale.*
- **NppStatus nppsSubCRev\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integersignal subtract from constant and scale.*

- **NppStatus nppsSubCRev\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.*
- **NppStatus nppsSubCRev\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.*
- **NppStatus nppsSubCRev\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract from constant.*
- **NppStatus nppsSubCRev\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.*
- **NppStatus nppsSubCRev\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point, in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract from constant.*
- **NppStatus nppsSubCRev\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.*

## DivC

Divides each sample of a signal by a constant.

- **NppStatus nppsDivC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value*
- **NppStatus nppsDivC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.*

- **NppStatus nppsDivC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal divided by constant and scale.*
- **NppStatus nppsDivC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal divided by constant and scale.*
- **NppStatus nppsDivC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal divided by constant and scale.*
- **NppStatus nppsDivC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal divided by constant and scale.*
- **NppStatus nppsDivC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal divided by constant.*
- **NppStatus nppsDivC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal divided by constant.*
- **NppStatus nppsDivC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.*
- **NppStatus nppsDivC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.*
- **NppStatus nppsDivC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point in place signal divided by constant.*

- `NppStatus nppsDivC_64f` (const `Npp64f` \*pSrc, `Npp64f` nValue, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal divided by constant.*
- `NppStatus nppsDivC_64fc_I` (`Npp64fc` nValue, `Npp64fc` \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.*
- `NppStatus nppsDivC_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` nValue, `Npp64fc` \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.*

## DivCRev

Divides a constant by each sample of a signal.

- `NppStatus nppsDivCRev_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char signal in place constant divided by signal, scale, then clamp to saturated value*
- `NppStatus nppsDivCRev_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal divided by constant, then clamp to saturated value.*
- `NppStatus nppsDivCRev_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place constant divided by signal, then clamp to saturated value.*
- `NppStatus nppsDivCRev_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal divided by constant, then clamp to saturated value.*
- `NppStatus nppsDivCRev_16s_I` (`Npp16s` nValue, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short in place constant divided by signal, then clamp to saturated value.*
- `NppStatus nppsDivCRev_16s` (const `Npp16s` \*pSrc, `Npp16s` nValue, `Npp16s` \*pDst, int nLength)  
*16-bit signed short constant divided by signal, then clamp to saturated value.*
- `NppStatus nppsDivCRev_32s_I` (`Npp32s` nValue, `Npp32s` \*pSrcDst, int nLength)  
*32-bit signed integer in place constant divided by signal.*
- `NppStatus nppsDivCRev_32s` (const `Npp32s` \*pSrc, `Npp32s` nValue, `Npp32s` \*pDst, int nLength)  
*32-bit signed integer constant divided by signal.*
- `NppStatus nppsDivCRev_32f_I` (`Npp32f` nValue, `Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point in place constant divided by signal.*
- `NppStatus nppsDivCRev_32f` (const `Npp32f` \*pSrc, `Npp32f` nValue, `Npp32f` \*pDst, int nLength)  
*32-bit floating point constant divided by signal.*
- `NppStatus nppsDivCRev_64f_I` (`Npp64f` nValue, `Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point in place constant divided by signal.*
- `NppStatus nppsDivCRev_64f` (const `Npp64f` \*pSrc, `Npp64f` nValue, `Npp64f` \*pDst, int nLength)  
*64-bit floating point constant divided by signal.*

## Add Signal

Sample by sample addition of two signals.

- **NppStatus** **nppsAdd\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned int signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal add signal, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_8u16u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char add signal, scale, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short add signal, scale, then clamp to saturated value.*
- **NppStatus** **nppsAdd\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_32s_Sfs` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, `Npp32s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_64s_Sfs` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, `Npp64s` \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_16sc_Sfs` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, `Npp16sc` \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_32sc_Sfs` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, `Npp32sc` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed complex integer add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short in place signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point in place signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_64f_I` (const `Npp64f` \*pSrc, `Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point in place signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_32fc_I` (const `Npp32fc` \*pSrc, `Npp32fc` \*pSrcDst, int nLength)  
*32-bit complex floating point in place signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_64fc_I` (const `Npp64fc` \*pSrc, `Npp64fc` \*pSrcDst, int nLength)  
*64-bit complex floating point in place signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_16s32s_I` (const `Npp16s` \*pSrc, `Npp32s` \*pSrcDst, int nLength)  
*16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.*
- `NppStatus nppsAdd_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.*
- `NppStatus nppsAdd_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.*
- `NppStatus nppsAdd_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.*
- `NppStatus nppsAdd_32s_ISfs` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAdd\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAdd\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.*

## AddProduct Signal

Adds sample by sample product of two signals to the destination signal.

- **NppStatus** **nppsAddProduct\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)

*32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)

*64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)

*32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)

*64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.*



## Mul Signal

Sample by sample multiplication the samples of two signals.

- **NppStatus nppsMul\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_8u16u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.*
- **NppStatus nppsMul\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_32f32fc** (const **Npp32f** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal time signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_32sc\_Sfs** (const **Npp32sc** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed complex integer signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16u16s\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.*
- **NppStatus nppsMul\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.*
- **NppStatus nppsMul\_32s32sc\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.*
- **NppStatus nppsMul\_Low\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_I** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short in place signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point in place signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point in place signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point in place signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32f32fc\_I** (const **Npp32f** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.*
- **NppStatus nppsMul\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsMul\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s32sc\_ISfs** (const **Npp32s** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.*

## Sub Signal

Sample by sample subtraction of the samples of two signals.

- **NppStatus nppsSub\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.*
- **NppStatus nppsSub\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)

*8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_16u_Sfs` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength, int nScaleFactor)

*16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_16s_Sfs` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, `Npp16s` \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_32s_Sfs` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, `Npp32s` \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_16sc_Sfs` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, `Npp16sc` \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_32sc_Sfs` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, `Npp32sc` \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.*

- `NppStatus nppsSub_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)

*16-bit signed short in place signal subtract signal, then clamp to saturated value.*

- `NppStatus nppsSub_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)

*32-bit floating point in place signal subtract signal, then clamp to saturated value.*

- `NppStatus nppsSub_64f_I` (const `Npp64f` \*pSrc, `Npp64f` \*pSrcDst, int nLength)

*64-bit floating point in place signal subtract signal, then clamp to saturated value.*

- `NppStatus nppsSub_32fc_I` (const `Npp32fc` \*pSrc, `Npp32fc` \*pSrcDst, int nLength)

*32-bit complex floating point in place signal subtract signal, then clamp to saturated value.*

- `NppStatus nppsSub_64fc_I` (const `Npp64fc` \*pSrc, `Npp64fc` \*pSrcDst, int nLength)

*64-bit complex floating point in place signal subtract signal, then clamp to saturated value.*

- `NppStatus nppsSub_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, int nScaleFactor)

*8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsSub_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsSub_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsSub_32s_ISfs` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsSub\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsSub\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.*

## Div Signal

Sample by sample division of the samples of two signals.

- **NppStatus** **nppsDiv\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)

*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)

*16-bit unsigned short signal divide signal, scale, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal divide signal, scale, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal divide signal, scale, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short signal divide signal, scale, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_32s16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.*

- **NppStatus** **nppsDiv\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)

*32-bit floating point signal divide signal, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)

*64-bit floating point signal divide signal, then clamp to saturated value.*

- **NppStatus** **nppsDiv\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)

*32-bit complex floating point signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point in place signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point in place signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point in place signal divide signal, then clamp to saturated value.*

## Div\_Round Signal

Sample by sample division of the samples of two signals with rounding.

- **NppStatus nppsDiv\_Round\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)

*16-bit signed short signal divide signal, scale, round, then clamp to saturated value.*

- `NppStatus nppsDiv_Round_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)

*8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.*

- `NppStatus nppsDiv_Round_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)

*16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.*

- `NppStatus nppsDiv_Round_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)

*16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.*

## Absolute Value Signal

Absolute value of each sample of a signal.

- `NppStatus nppsAbs_16s` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength)

*16-bit signed short signal absolute value.*

- `NppStatus nppsAbs_32s` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength)

*32-bit signed integer signal absolute value.*

- `NppStatus nppsAbs_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)

*32-bit floating point signal absolute value.*

- `NppStatus nppsAbs_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)

*64-bit floating point signal absolute value.*

- `NppStatus nppsAbs_16s_I` (`Npp16s` \*pSrcDst, int nLength)

*16-bit signed short signal absolute value.*

- `NppStatus nppsAbs_32s_I` (`Npp32s` \*pSrcDst, int nLength)

*32-bit signed integer signal absolute value.*

- `NppStatus nppsAbs_32f_I` (`Npp32f` \*pSrcDst, int nLength)

*32-bit floating point signal absolute value.*

- `NppStatus nppsAbs_64f_I` (`Npp64f` \*pSrcDst, int nLength)

*64-bit floating point signal absolute value.*

## Square Signal

Squares each sample of a signal.

- `NppStatus nppsSqr_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)

*32-bit floating point signal squared.*

- **NppStatus nppsSqr\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal squared.*
- **NppStatus nppsSqr\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16sc\_ISfs** (**Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*



## Square Root Signal

Square root of each sample of a signal.

- **NppStatus nppsSqrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal square root.*
- **NppStatus nppsSqrt\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal square root.*
- **NppStatus nppsSqrt\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal square root.*
- **NppStatus nppsSqrt\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal square root.*
- **NppStatus nppsSqrt\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_64s\_Sfs** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*
- **NppStatus nppsSqrt\_64s16s\_Sfs** (const **Npp64s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*

*64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*

- **NppStatus nppsSqrt\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16sc\_ISfs** (**Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_64s\_ISfs** (**Npp64s** \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*

## Cube Root Signal

Cube root of each sample of a signal.

- **NppStatus nppsCubrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal cube root.*
- **NppStatus nppsCubrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.*

## Exponent Signal

E raised to the power of each sample of a signal.

- **NppStatus nppsExp\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal exponent.*
- **NppStatus nppsExp\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal exponent.*
- **NppStatus nppsExp\_32f64f** (const **Npp32f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*32-bit floating point signal exponent with 64-bit floating point result.*
- **NppStatus nppsExp\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal exponent.*
- **NppStatus nppsExp\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal exponent.*

- **NppStatus nppsExp\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_64s\_Sfs** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_32s\_ISfs** (**Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_64s\_ISfs** (**Npp64s** \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*

## Natural Logarithm Signal

Natural logarithm of each sample of a signal.

- **NppStatus nppsLn\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- **NppStatus nppsLn\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- **NppStatus nppsLn\_64f32f** (const **Npp64f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*64-bit floating point signal natural logarithm with 32-bit floating point result.*
- **NppStatus nppsLn\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- **NppStatus nppsLn\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- **NppStatus nppsLn\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*
- **NppStatus nppsLn\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*
- **NppStatus nppsLn\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.*
- **NppStatus nppsLn\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*

- [NppStatus nppsLn\\_32s\\_ISfs](#) ([Npp32s](#) \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*

## Ten Times Base Ten Logarithm Signal

Ten times the decimal logarithm of each sample of a signal.

- [NppStatus npps10Log10\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc, [Npp32s](#) \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*
- [NppStatus npps10Log10\\_32s\\_ISfs](#) ([Npp32s](#) \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*

## Inverse Tangent Signal

Inverse tangent of each sample of a signal.

- [NppStatus nppsArctan\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- [NppStatus nppsArctan\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength)  
*64-bit floating point signal inverse tangent.*
- [NppStatus nppsArctan\\_32f\\_I](#) ([Npp32f](#) \*pSrcDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- [NppStatus nppsArctan\\_64f\\_I](#) ([Npp64f](#) \*pSrcDst, int nLength)  
*64-bit floating point signal inverse tangent.*

## Normalize Signal

Normalize each sample of a real or complex signal using offset and division operations.

- [NppStatus nppsNormalize\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength, [Npp32f](#) vSub, [Npp32f](#) vDiv)  
*32-bit floating point signal normalize.*
- [NppStatus nppsNormalize\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) \*pDst, int nLength, [Npp32fc](#) vSub, [Npp32fc](#) vDiv)  
*32-bit complex floating point signal normalize.*
- [NppStatus nppsNormalize\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength, [Npp64f](#) vSub, [Npp64f](#) vDiv)

*64-bit floating point signal normalize.*

- `NppStatus nppsNormalize_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` \*pDst, int nLength, `Npp64fc` vSub, `Npp64f` vDiv)

*64-bit complex floating point signal normalize.*

- `NppStatus nppsNormalize_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, `Npp16s` vSub, int vDiv, int nScaleFactor)

*16-bit signed short signal normalize, scale, then clamp to saturated value.*

- `NppStatus nppsNormalize_16sc_Sfs` (const `Npp16sc` \*pSrc, `Npp16sc` \*pDst, int nLength, `Npp16sc` vSub, int vDiv, int nScaleFactor)

*16-bit complex signed short signal normalize, scale, then clamp to saturated value.*

## Cauchy, CauchyD, and CauchyDD2 Signal

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

- `NppStatus nppsCauchy_32f_I` (`Npp32f` \*pSrcDst, int nLength, `Npp32f` nParam)

*32-bit floating point signal Cauchy error calculation.*

- `NppStatus nppsCauchyD_32f_I` (`Npp32f` \*pSrcDst, int nLength, `Npp32f` nParam)

*32-bit floating point signal Cauchy first derivative.*

- `NppStatus nppsCauchyDD2_32f_I` (`Npp32f` \*pSrcDst, `Npp32f` \*pD2FVal, int nLength, `Npp32f` nParam)

*32-bit floating point signal Cauchy first and second derivatives.*

## AndC

Bitwise AND of a constant and each sample of a signal.

- `NppStatus nppsAndC_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)

*8-bit unsigned char signal and with constant.*

- `NppStatus nppsAndC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)

*16-bit unsigned short signal and with constant.*

- `NppStatus nppsAndC_32u` (const `Npp32u` \*pSrc, `Npp32u` nValue, `Npp32u` \*pDst, int nLength)

*32-bit unsigned integer signal and with constant.*

- `NppStatus nppsAndC_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)

*8-bit unsigned char in place signal and with constant.*

- `NppStatus nppsAndC_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)

*16-bit unsigned short in place signal and with constant.*

- `NppStatus nppsAndC_32u_I` (`Npp32u` nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal and with constant.*

## And

Sample by sample bitwise AND of samples from two signals.

- `NppStatus nppsAnd_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal and with signal.*
- `NppStatus nppsAnd_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal and with signal.*
- `NppStatus nppsAnd_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal and with signal.*
- `NppStatus nppsAnd_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal and with signal.*
- `NppStatus nppsAnd_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal and with signal.*
- `NppStatus nppsAnd_32u_I` (const `Npp32u` \*pSrc, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal and with signal.*

## OrC

Bitwise OR of a constant and each sample of a signal.

- `NppStatus nppsOrC_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal or with constant.*
- `NppStatus nppsOrC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal or with constant.*
- `NppStatus nppsOrC_32u` (const `Npp32u` \*pSrc, `Npp32u` nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal or with constant.*
- `NppStatus nppsOrC_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with constant.*
- `NppStatus nppsOrC_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with constant.*
- `NppStatus nppsOrC_32u_I` (`Npp32u` nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal or with constant.*

## Or

Sample by sample bitwise OR of the samples from two signals.

- **NppStatus nppsOr\_8u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal or with signal.*
- **NppStatus nppsOr\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal or with signal.*
- **NppStatus nppsOr\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal or with signal.*
- **NppStatus nppsOr\_8u\_I** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with signal.*
- **NppStatus nppsOr\_16u\_I** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with signal.*
- **NppStatus nppsOr\_32u\_I** (const **Npp32u** \*pSrc, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal or with signal.*

## XorC

Bitwise XOR of a constant and each sample of a signal.

- **NppStatus nppsXorC\_8u** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with constant.*
- **NppStatus nppsXorC\_16u** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with constant.*
- **NppStatus nppsXorC\_32u** (const **Npp32u** \*pSrc, **Npp32u** nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with constant.*
- **NppStatus nppsXorC\_8u\_I** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with constant.*
- **NppStatus nppsXorC\_16u\_I** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with constant.*
- **NppStatus nppsXorC\_32u\_I** (**Npp32u** nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal exclusive or with constant.*

## Xor

Sample by sample bitwise XOR of the samples from two signals.

- `NppStatus nppsXor_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with signal.*
- `NppStatus nppsXor_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with signal.*
- `NppStatus nppsXor_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with signal.*
- `NppStatus nppsXor_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with signal.*
- `NppStatus nppsXor_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with signal.*
- `NppStatus nppsXor_32u_I` (const `Npp32u` \*pSrc, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal exclusive or with signal.*

## Not

Bitwise NOT of each sample of a signal.

- `NppStatus nppsNot_8u` (const `Npp8u` \*pSrc, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char not signal.*
- `NppStatus nppsNot_16u` (const `Npp16u` \*pSrc, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short not signal.*
- `NppStatus nppsNot_32u` (const `Npp32u` \*pSrc, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer not signal.*
- `NppStatus nppsNot_8u_I` (`Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place not signal.*
- `NppStatus nppsNot_16u_I` (`Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place not signal.*
- `NppStatus nppsNot_32u_I` (`Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place not signal.*



## LShiftC

Left shifts the bits of each sample of a signal by a constant amount.

- **NppStatus nppsLShiftC\_8u** (const **Npp8u** \*pSrc, int nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal left shift with constant.*
- **NppStatus nppsLShiftC\_16u** (const **Npp16u** \*pSrc, int nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal left shift with constant.*
- **NppStatus nppsLShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal left shift with constant.*
- **NppStatus nppsLShiftC\_32u** (const **Npp32u** \*pSrc, int nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal left shift with constant.*
- **NppStatus nppsLShiftC\_32s** (const **Npp32s** \*pSrc, int nValue, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer signal left shift with constant.*
- **NppStatus nppsLShiftC\_8u\_I** (int nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_16u\_I** (int nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_16s\_I** (int nValue, **Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_32u\_I** (int nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_32s\_I** (int nValue, **Npp32s** \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal left shift with constant.*

## RShiftC

Right shifts the bits of each sample of a signal by a constant amount.

- **NppStatus nppsRShiftC\_8u** (const **Npp8u** \*pSrc, int nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal right shift with constant.*
- **NppStatus nppsRShiftC\_16u** (const **Npp16u** \*pSrc, int nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal right shift with constant.*
- **NppStatus nppsRShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal right shift with constant.*
- **NppStatus nppsRShiftC\_32u** (const **Npp32u** \*pSrc, int nValue, **Npp32u** \*pDst, int nLength)

*32-bit unsigned integer signal right shift with constant.*

- [NppStatus nppsRShiftC\\_32s](#) (const [Npp32s](#) \*pSrc, int nValue, [Npp32s](#) \*pDst, int nLength)  
*32-bit signed integer signal right shift with constant.*
- [NppStatus nppsRShiftC\\_8u\\_I](#) (int nValue, [Npp8u](#) \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal right shift with constant.*
- [NppStatus nppsRShiftC\\_16u\\_I](#) (int nValue, [Npp16u](#) \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal right shift with constant.*
- [NppStatus nppsRShiftC\\_16s\\_I](#) (int nValue, [Npp16s](#) \*pSrcDst, int nLength)  
*16-bit signed short in place signal right shift with constant.*
- [NppStatus nppsRShiftC\\_32u\\_I](#) (int nValue, [Npp32u](#) \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal right shift with constant.*
- [NppStatus nppsRShiftC\\_32s\\_I](#) (int nValue, [Npp32s](#) \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal right shift with constant.*

## Statistical Functions

Functions that provide global signal statistics like: average, standard deviation, minimum, etc.

- [NppStatus nppsSumGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f Sum.*
- [NppStatus nppsMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f Max.*
- [NppStatus nppsMinGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f Min.*
- [NppStatus nppsSumGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s32s\_Sfs.*
- [NppStatus nppsMaxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s Max.*
- [NppStatus nppsMinGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s Min.*
- [NppStatus nppsSumGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64fc Sum.*
- [NppStatus nppsSumGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s\_Sfs Sum.*
- [NppStatus nppsSumGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for 16sc\_Sfs Sum.*

- [NppStatus nppsSumGetBufferSize\\_16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16sc32sc\_Sfs Sum.*
- [NppStatus nppsSumGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64f Sum.*
- [NppStatus nppsMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64f Max.*
- [NppStatus nppsMinGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64f Min.*
- [NppStatus nppsSumGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32s\_Sfs Sum.*
- [NppStatus nppsMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32s Max.*
- [NppStatus nppsMinGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32s Min.*
- [NppStatus nppsSumGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32fc Sum.*
- [NppStatus nppsSum\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pSum, [NppHintAlgorithm](#) eHint, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector sum method*
- [NppStatus nppsSum\\_32fc](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32fc](#) \*pSum, [NppHintAlgorithm](#) eHint, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector sum method*
- [NppStatus nppsSum\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector sum method*
- [NppStatus nppsSum\\_64fc](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64fc](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double complex vector sum method*
- [NppStatus nppsSum\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pSum, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit short vector sum with integer scaling method*
- [NppStatus nppsSum\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pSum, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector sum with integer scaling method*

- **NppStatus nppsSum\_16sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp16sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum with integer scaling method*
- **NppStatus nppsSum\_16sc32sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp32sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum (32bit int complex) with integer scaling method*
- **NppStatus nppsSum\_16s32s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit integer vector sum (32bit) with integer scaling method*
- **NppStatus nppsMax\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMax, **Npp8u** \*pDeviceBuffer)  
*16-bit integer vector max method*
- **NppStatus nppsMax\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMax, **Npp8u** \*pDeviceBuffer)  
*32-bit integer vector max method*
- **NppStatus nppsMax\_32f** (const **Npp32f** \*pSrc, int nLength, **Npp32f** \*pMax, **Npp8u** \*pDeviceBuffer)  
*32-bit float vector max method*
- **NppStatus nppsMax\_64f** (const **Npp64f** \*pSrc, int nLength, **Npp64f** \*pMax, **Npp8u** \*pDeviceBuffer)  
*64-bit float vector max method*
- **NppStatus nppsMin\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMin, **Npp8u** \*pDeviceBuffer)  
*16-bit integer vector min method*
- **NppStatus nppsMin\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMin, **Npp8u** \*pDeviceBuffer)  
*32-bit integer vector min method*
- **NppStatus nppsMin\_32f** (const **Npp32f** \*pSrc, int nLength, **Npp32f** \*pMin, **Npp8u** \*pDeviceBuffer)  
*32-bit integer vector min method*
- **NppStatus nppsMin\_64f** (const **Npp64f** \*pSrc, int nLength, **Npp64f** \*pMin, **Npp8u** \*pDeviceBuffer)  
*64-bit integer vector min method*
- **NppStatus nppsMinMaxGetBufferSize\_8u** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_8u.*
- **NppStatus nppsMinMaxGetBufferSize\_16s** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_16s.*
- **NppStatus nppsMinMaxGetBufferSize\_16u** (int nLength, int \*hpBufferSize)

*Device-buffer size (in bytes) for nppsMinMax\_16u.*

- [NppStatus nppsMinMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32s.*
- [NppStatus nppsMinMaxGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32u.*
- [NppStatus nppsMinMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32f.*
- [NppStatus nppsMinMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_64f.*
- [NppStatus nppsMinMax\\_8u](#) (const [Npp8u](#) \*pSrc, int nLength, [Npp8u](#) \*pMin, [Npp8u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*8-bit char vector min and max method*
- [NppStatus nppsMinMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short vector min and max method*
- [NppStatus nppsMinMax\\_16u](#) (const [Npp16u](#) \*pSrc, int nLength, [Npp16u](#) \*pMin, [Npp16u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short vector min and max method*
- [NppStatus nppsMinMax\\_32u](#) (const [Npp32u](#) \*pSrc, int nLength, [Npp32u](#) \*pMin, [Npp32u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned int vector min and max method*
- [NppStatus nppsMinMax\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, [Npp32s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed int vector min and max method*
- [NppStatus nppsMinMax\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, [Npp32f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector min and max method*
- [NppStatus nppsMinMax\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, [Npp64f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector min and max method*

## Filtering Functions

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

- [NppStatus nppsIntegralGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)
- [NppStatus nppsIntegral\\_32s](#) (const [Npp32s](#) \*pSrc, [Npp32s](#) \*pDst, int nLength, [Npp8u](#) \*pDeviceBuffer)

### 7.11.1 Function Documentation

#### 7.11.1.1 NppStatus npps10Log10\_32s\_ISfs (Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.2 NppStatus npps10Log10\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.3 NppStatus nppsAbs\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*)

16-bit signed short signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.4 NppStatus nppsAbs\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.5 NppStatus nppsAbs\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.6 NppStatus nppsAbs\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.7 NppStatus nppsAbs\_32s (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.8 NppStatus nppsAbs\_32s\_I (Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.9 NppStatus nppsAbs\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.10 NppStatus nppsAbs\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.11 NppStatus nppsAdd\_16s (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*)**

16-bit signed short signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.11.1.12 NppStatus nppsAdd\_16s32f (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)**

16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.13 NppStatus nppsAdd\_16s32s\_I (const Npp16s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*)**

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.14 NppStatus nppsAdd\_16s\_I (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.15 NppStatus nppsAdd\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.16 NppStatus nppsAdd\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.17 NppStatus nppsAdd\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.18 NppStatus nppsAdd\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short add signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.19 NppStatus nppsAdd\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)

16-bit unsigned short signal add signal, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.20 NppStatus nppsAdd\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.21 **NppStatus nppsAdd\_16u\_Sfs** (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short add signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.22 **NppStatus nppsAdd\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal add signal, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.23 **NppStatus nppsAdd\_32f\_I** (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)

32-bit floating point in place signal add signal, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal2 elements to be added to signal1 elements

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.24 NppStatus nppsAdd\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.25 NppStatus nppsAdd\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.26 NppStatus nppsAdd\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.27 **NppStatus nppsAdd\_32s\_Sfs** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer add signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.28 **NppStatus nppsAdd\_32sc\_ISfs** (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.29 **NppStatus nppsAdd\_32sc\_Sfs** (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer add signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.30 NppStatus nppsAdd\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned int signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.31 NppStatus nppsAdd\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.32 NppStatus nppsAdd\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.33 **NppStatus nppsAdd\_64fc** (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)

64-bit complex floating point signal add signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be added to signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.34 **NppStatus nppsAdd\_64fc\_I** (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)

64-bit complex floating point in place signal add signal, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.35 **NppStatus nppsAdd\_64s\_Sfs** (const Npp64s \* *pSrc1*, const Npp64s \* *pSrc2*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer add signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.36 NppStatus nppsAdd\_8u16u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.37 NppStatus nppsAdd\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.38 NppStatus nppsAdd\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.39 NppStatus nppsAddC\_16s\_ISfs (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short in place signal add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.40 NppStatus nppsAddC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.41 NppStatus nppsAddC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.42 NppStatus nppsAddC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.43 NppStatus nppsAddC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.44 NppStatus nppsAddC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short vector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.45 NppStatus nppsAddC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal add constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.46 NppStatus nppsAddC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.47 NppStatus nppsAddC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.48 NppStatus nppsAddC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.49 NppStatus nppsAddC\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal add constant and scale.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.50 NppStatus nppsAddC\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integersignal add constant and scale.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.51 **NppStatus nppsAddC\_32sc\_ISfs** (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.52 **NppStatus nppsAddC\_32sc\_Sfs** (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

#### Parameters:

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.53 **NppStatus nppsAddC\_64f** (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)

64-bit floating pointsignal add constant.

#### Parameters:

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.54 NppStatus nppsAddC\_64f\_I (Npp64f nValue, Npp64f \* pSrcDst, int nLength)**

64-bit floating point, in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.55 NppStatus nppsAddC\_64fc (const Npp64fc \* pSrc, Npp64fc nValue, Npp64fc \* pDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.56 NppStatus nppsAddC\_64fc\_I (Npp64fc nValue, Npp64fc \* pSrcDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.57 NppStatus nppsAddC\_8u\_ISfs (Npp8u nValue, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal add constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.58 `NppStatus nppsAddC_8u_Sfs (const Npp8u *pSrc, Npp8u nValue, Npp8u *pDst, int nLength, int nScaleFactor)`

8-bit unsigned charvector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.59 `NppStatus nppsAddProduct_16s32s_Sfs (const Npp16s *pSrc1, const Npp16s *pSrc2, Npp32s *pDst, int nLength, int nScaleFactor)`

16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#).  
*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



### 7.11.1.60 **NppStatus nppsAddProduct\_16s\_Sfs** (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.61 **NppStatus nppsAddProduct\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.62 **NppStatus nppsAddProduct\_32fc** (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)

32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.63 **NppStatus nppsAddProduct\_32s\_Sfs** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.64 **NppStatus nppsAddProduct\_64f** (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.65 **NppStatus nppsAddProduct\_64fc** (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)

64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.66 NppStatus nppsAddProductC\_16s\_ISfs (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.67 NppStatus nppsAddProductC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal add product of signal times constant to destination signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.68 NppStatus nppsAddProductC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add product of signal times constant to destination signal, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.69 **NppStatus nppsAddProductC\_16sc\_Sfs** (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add product of signal times constant to destination signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.70 **NppStatus nppsAddProductC\_16u\_ISfs** (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.71 **NppStatus nppsAddProductC\_16u\_Sfs** (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short add product of signal times constant to destination signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.72 NppStatus nppsAddProductC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal add product of signal times constant to destination signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.73 NppStatus nppsAddProductC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal add product of signal times constant to destination signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.74 NppStatus nppsAddProductC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add product of signal times constant to destination signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.75 **NppStatus nppsAddProductC\_32fc\_I** (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add product of signal times constant to destination signal.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be multiplied by each vector element  
*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.76 **NppStatus nppsAddProductC\_32s\_ISfs** (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal add product of signal times constant to destination signal and scale.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be multiplied by each vector element  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.77 **NppStatus nppsAddProductC\_32s\_Sfs** (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal add product of signal times constant to destination signal and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be multiplied by each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.78 NppStatus nppsAddProductC\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add product of signal times constant to destination signal and scale.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.79 NppStatus nppsAddProductC\_32sc\_Sfs (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add product of signal times constant to destination signal and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.80 NppStatus nppsAddProductC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal add product of signal times constant to destination signal.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.81 NppStatus nppsAddProductC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point, in place signal add product of signal times constant to destination signal.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.82 NppStatus nppsAddProductC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add product of signal times constant to destination signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.83 NppStatus nppsAddProductC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add product of signal times constant to destination signal.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



#### 7.11.1.84 NppStatus nppsAddProductC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal add product of signal times constant to destination signal, scale, then clamp to saturated value

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.85 NppStatus nppsAddProductC\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char add product of signal times constant to destination signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.86 NppStatus nppsAnd\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)

16-bit unsigned short signal and with signal.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.87 NppStatus nppsAnd\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal and with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.88 NppStatus nppsAnd\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal and with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.89 NppStatus nppsAnd\_32u\_I (const Npp32u \* *pSrc*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned integer in place signal and with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.90 NppStatus nppsAnd\_8u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal and with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal2 elements to be anded with signal1 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.91 NppStatus nppsAnd\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal and with signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer. signal2 elements to be anded with signal1 elements  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.92 NppStatus nppsAndC\_16u (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal and with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be anded with each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.93 NppStatus nppsAndC\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be anded with each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.94 NppStatus nppsAndC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal and with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be anded with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.95 NppStatus nppsAndC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be anded with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.96 NppStatus nppsAndC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal and with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be anded with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.97 NppStatus nppsAndC\_8u\_I (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be anded with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.98 NppStatus nppsArctan\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal inverse tangent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.99 NppStatus nppsArctan\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal inverse tangent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.100 NppStatus nppsArctan\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.101 NppStatus nppsArctan\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.102 NppStatus nppsCauchy\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nParam)**

32-bit floating point signal Cauchy error calculation.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nParam* constant used in Cauchy formula

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.103 NppStatus nppsCauchyD\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nParam)**

32-bit floating point signal Cauchy first derivative.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nParam* constant used in Cauchy formula

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.104 NppStatus nppsCauchyDD2\_32f\_I (Npp32f \* pSrcDst, Npp32f \* pD2FVal, int nLength, Npp32f nParam)**

32-bit floating point signal Cauchy first and second derivatives.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*pD2FVal* Source Signal Pointer. This signal contains the second derivative of the source signal.

*nLength* Signal Length.

*nParam* constant used in Cauchy formula

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.105 NppStatus nppsCopy\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *len*)**

16-bit signed short, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.106 NppStatus nppsCopy\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *len*)**

16-bit complex short, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.107 NppStatus nppsCopy\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *len*)**

32-bit float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.108 NppStatus nppsCopy\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *len*)**

32-bit complex float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.109 NppStatus nppsCopy\_32s (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.110 NppStatus nppsCopy\_32sc (const Npp32sc \* *pSrc*, Npp32sc \* *pDst*, int *len*)**

32-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.111 NppStatus nppsCopy\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *len*)**

64-bit complex double, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.



*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.112 NppStatus nppsCopy\_64s (const Npp64s \* *pSrc*, Npp64s \* *pDst*, int *len*)**

64-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.113 NppStatus nppsCopy\_64sc (const Npp64sc \* *pSrc*, Npp64sc \* *pDst*, int *len*)**

64-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.114 NppStatus nppsCopy\_8u (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *len*)**

8-bit unsigned char, vector copy method

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*len* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.115 NppStatus nppsCubrt\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal cube root.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.116 NppStatus nppsCubrt\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.117 NppStatus nppsDiv\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.118 NppStatus nppsDiv\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal divide signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.119 NppStatus nppsDiv\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.120 NppStatus nppsDiv\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal divide signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.121 **NppStatus nppsDiv\_16u\_ISfs** (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.122 **NppStatus nppsDiv\_16u\_Sfs** (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal divide signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.123 **NppStatus nppsDiv\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal divide signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.124 NppStatus nppsDiv\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.125 NppStatus nppsDiv\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.126 NppStatus nppsDiv\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.127 NppStatus nppsDiv\_32s16s\_Sfs (const Npp16s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.128** `NppStatus nppsDiv_32s_ISfs (const Npp32s * pSrc, Npp32s * pSrcDst, int nLength, int nScaleFactor)`

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.129** `NppStatus nppsDiv_32s_Sfs (const Npp32s * pSrc1, const Npp32s * pSrc2, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed integer signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.130 NppStatus nppsDiv\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.131 NppStatus nppsDiv\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.132 NppStatus nppsDiv\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.133 NppStatus nppsDiv\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.134 NppStatus nppsDiv\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.135 NppStatus nppsDiv\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.136 NppStatus nppsDiv\_Round\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.137 NppStatus nppsDiv\_Round\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

16-bit signed short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.138 NppStatus nppsDiv\_Round\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.139 NppStatus nppsDiv\_Round\_16u\_Sfs (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nRndMode* various rounding modes.  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.140 NppStatus nppsDiv\_Round\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements  
*nLength* [Signal Length](#).  
*nRndMode* various rounding modes.  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.141 NppStatus nppsDiv\_Round\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nRndMode* various rounding modes.  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.142 NppStatus nppsDivC\_16s\_ISfs (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.143 NppStatus nppsDivC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.144 NppStatus nppsDivC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.145 NppStatus nppsDivC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.146 NppStatus nppsDivC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.147 NppStatus nppsDivC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.148 NppStatus nppsDivC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.149 NppStatus nppsDivC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.150 NppStatus nppsDivC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.151 NppStatus nppsDivC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.152 NppStatus nppsDivC\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal divided by constant and scale.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.153 NppStatus nppsDivC\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal divided by constant and scale.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.154 NppStatus nppsDivC\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal divided by constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.155 NppStatus nppsDivC\_32sc\_Sfs (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) signal divided by constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.156 NppStatus nppsDivC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.157 NppStatus nppsDivC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.158 NppStatus nppsDivC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.159 NppStatus nppsDivC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.160 NppStatus nppsDivC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value



**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.161** `NppStatus nppsDivC_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.162** `NppStatus nppsDivCRev_16s (const Npp16s * pSrc, Npp16s nValue, Npp16s * pDst, int nLength)`

16-bit signed short constant divided by signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.163 NppStatus nppsDivCRev\_16s\_I (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place constant divided by signal, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.164 NppStatus nppsDivCRev\_16u (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal divided by constant, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.165 NppStatus nppsDivCRev\_16u\_I (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place constant divided by signal, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.166 NppStatus nppsDivCRev\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point constant divided by signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.167 NppStatus nppsDivCRev\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place constant divided by signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.168 NppStatus nppsDivCRev\_32s (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer constant divided by signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.169 NppStatus nppsDivCRev\_32s\_I (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer in place constant divided by signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.170 NppStatus nppsDivCRev\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point constant divided by signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.171 NppStatus nppsDivCRev\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place constant divided by signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided by each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.172 NppStatus nppsDivCRev\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal divided by constant, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.173 NppStatus nppsDivCRev\_8u\_I (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char signal in place constant divided by signal, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.174 NppStatus nppsExp\_16s\_ISfs (Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.175 NppStatus nppsExp\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.176 NppStatus nppsExp\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.177 NppStatus nppsExp\_32f64f (const Npp32f \* pSrc, Npp64f \* pDst, int nLength)**

32-bit floating point signal exponent with 64-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.178 NppStatus nppsExp\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.179 NppStatus nppsExp\_32s\_ISfs (Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.180 NppStatus nppsExp\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.181 NppStatus nppsExp\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.182 NppStatus nppsExp\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.183 NppStatus nppsExp\_64s\_ISfs (Npp64s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.184** `NppStatus nppsExp_64s_Sfs (const Npp64s * pSrc, Npp64s * pDst, int nLength, int nScaleFactor)`

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.185** `void nppsFree (void * pValues)`

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the `nppiMalloc_<modifier>` methods.

**Parameters:**

*pValues* A pointer to memory allocated using `nppiMalloc_<modifier>`.

**7.11.1.186** `NppStatus nppsIntegral_32s (const Npp32s * pSrc, Npp32s * pDst, int nLength, Npp8u * pDeviceBuffer)`

**7.11.1.187** `NppStatus nppsIntegralGetBufferSize_32s (int nLength, int * hpBufferSize)`

**7.11.1.188** `NppStatus nppsLn_16s_ISfs (Npp16s * pSrcDst, int nLength, int nScaleFactor)`

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.189 NppStatus nppsLn\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.190 NppStatus nppsLn\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.191 NppStatus nppsLn\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.192 NppStatus nppsLn\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.193 NppStatus nppsLn\_32s\_ISfs (Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.194 NppStatus nppsLn\_32s\_Sfs (const Npp32s \* pSrc, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.195 NppStatus nppsLn\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.196 NppStatus nppsLn\_64f32f (const Npp64f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

64-bit floating point signal natural logarithm with 32-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.197 NppStatus nppsLn\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.198 NppStatus nppsLShiftC\_16s (const Npp16s \* *pSrc*, int *nValue*, Npp16s \* *pDst*, int *nLength*)**

16-bit signed short signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.199 NppStatus nppsLShiftC\_16s\_I (int *nValue*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.200 NppStatus nppsLShiftC\_16u (const Npp16u \* pSrc, int nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal left shift with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be used to left shift each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.201 NppStatus nppsLShiftC\_16u\_I (int nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal left shift with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be used to left shift each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.202 NppStatus nppsLShiftC\_32s (const Npp32s \* pSrc, int nValue, Npp32s \* pDst, int nLength)**

32-bit signed integer signal left shift with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be used to left shift each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.203 NppStatus nppsLShiftC\_32s\_I (int *nValue*, Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.204 NppStatus nppsLShiftC\_32u (const Npp32u \* *pSrc*, int *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.205 NppStatus nppsLShiftC\_32u\_I (int *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.206 NppStatus nppsLShiftC\_8u (const Npp8u \* *pSrc*, int *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal left shift with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be used to left shift each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.207 NppStatus nppsLShiftC\_8u\_I (int nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal left shift with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be used to left shift each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.208 Npp16s\* nppsMalloc\_16s (int nSize)**

16-bit signal allocator.

**Parameters:**

*nSize* Number of shorts in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.209 Npp16sc\* nppsMalloc\_16sc (int nSize)**

16-bit complex-value signal allocator.

**Parameters:**

*nSize* Number of 16-bit complex numbers in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.210 Npp16u\* nppsMalloc\_16u (int nSize)**

16-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned shorts in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.211 Npp32f\* nppsMalloc\_32f (int nSize)**

32-bit float signal allocator.

**Parameters:**

*nSize* Number of floats in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.212 Npp32fc\* nppsMalloc\_32fc (int nSize)**

32-bit complex float signal allocator.

**Parameters:**

*nSize* Number of complex float values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.213 Npp32s\* nppsMalloc\_32s (int nSize)**

32-bit integer signal allocator.

**Parameters:**

*nSize* Number of ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.214 Npp32sc\* nppsMalloc\_32sc (int *nSize*)**

32-bit complex integer signal allocator.

**Parameters:**

*nSize* Number of complex integner values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.215 Npp32u\* nppsMalloc\_32u (int *nSize*)**

32-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.216 Npp64f\* nppsMalloc\_64f (int *nSize*)**

64-bit float (double) signal allocator.

**Parameters:**

*nSize* Number of doubles in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.217 Npp64fc\* nppsMalloc\_64fc (int *nSize*)**

64-bit complex complex signal allocator.

**Parameters:**

*nSize* Number of complex double valuess in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.



**7.11.1.218 Npp64s\* nppsMalloc\_64s (int nSize)**

64-bit long integer signal allocator.

**Parameters:**

*nSize* Number of long ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.219 Npp64sc\* nppsMalloc\_64sc (int nSize)**

64-bit complex long integer signal allocator.

**Parameters:**

*nSize* Number of complex long int values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.220 Npp8u\* nppsMalloc\_8u (int nSize)**

8-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned chars in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.11.1.221 NppStatus nppsMax\_16s (const Npp16s \*pSrc, int nLength, Npp16s \*pMax, Npp8u \*pDeviceBuffer)**

16-bit integer vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.222 NppStatus nppsMax\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMax*, Npp8u \* *pDeviceBuffer*)**

32-bit float vector max method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.223 NppStatus nppsMax\_32s (const Npp32s \* *pSrc*, int *nLength*, Npp32s \* *pMax*, Npp8u \* *pDeviceBuffer*)**

32-bit integer vector max method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.224 NppStatus nppsMax\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMax*, Npp8u \* *pDeviceBuffer*)**

64-bit float vector max method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.225 NppStatus nppsMaxGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16s Max.

This primitive provides the correct buffer size for nppsMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.226 NppStatus nppsMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32f Max.

This primitive provides the correct buffer size for nppsMax\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.227 NppStatus nppsMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32s Max.

This primitive provides the correct buffer size for nppsMax\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.228 NppStatus nppsMaxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64f Max.

This primitive provides the correct buffer size for nppsMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

#### 7.11.1.229 NppStatus nppsMin\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMin*, Npp8u \* *pDeviceBuffer*)

16-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.230 NppStatus nppsMin\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMin*, Npp8u \* *pDeviceBuffer*)

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.231 NppStatus nppsMin\_32s (const Npp32s \* *pSrc*, int *nLength*, Npp32s \* *pMin*, Npp8u \* *pDeviceBuffer*)

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.232 NppStatus nppsMin\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, Npp8u \* pDeviceBuffer)**

64-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.233 NppStatus nppsMinGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for 16s Min.

This primitive provides the correct buffer size for nppsMin\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.234 NppStatus nppsMinGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for 32f Min.

This primitive provides the correct buffer size for nppsMin\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.235 NppStatus nppsMinGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32s Min.

This primitive provides the correct buffer size for nppsMin\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.236 NppStatus nppsMinGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64f Min.

This primitive provides the correct buffer size for nppsMin\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.237 NppStatus nppsMinMax\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMin*, Npp16s \* *pMax*, Npp8u \* *pDeviceBuffer*)**

16-bit signed short vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.238 NppStatus nppsMinMax\_16u (const Npp16u \* *pSrc*, int *nLength*, Npp16u \* *pMin*, Npp16u \* *pMax*, Npp8u \* *pDeviceBuffer*)**

16-bit unsigned short vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.239 NppStatus nppsMinMax\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMin*, Npp32f \* *pMax*, Npp8u \* *pDeviceBuffer*)**

32-bit float vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.240 NppStatus nppsMinMax\_32s (const Npp32s \* *pSrc*, int *nLength*, Npp32s \* *pMin*, Npp32s \* *pMax*, Npp8u \* *pDeviceBuffer*)**

32-bit signed int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.241 NppStatus nppsMinMax\_32u (const Npp32u \* *pSrc*, int *nLength*, Npp32u \* *pMin*, Npp32u \* *pMax*, Npp8u \* *pDeviceBuffer*)**

32-bit unsigned int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.242 NppStatus nppsMinMax\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMin*, Npp64f \* *pMax*, Npp8u \* *pDeviceBuffer*)**

64-bit double vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.243 NppStatus nppsMinMax\_8u (const Npp8u \* *pSrc*, int *nLength*, Npp8u \* *pMin*, Npp8u \* *pMax*, Npp8u \* *pDeviceBuffer*)**

8-bit char vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.244 NppStatus nppsMinMaxGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.245 NppStatus nppsMinMaxGetBufferSize\_16u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.246 NppStatus nppsMinMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.247 NppStatus nppsMinMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.248 NppStatus nppsMinMaxGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.249 NppStatus nppsMinMaxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.250 NppStatus nppsMinMaxGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.11.1.251 NppStatus nppsMul\_16s (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*)**

16-bit signed short signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.252 NppStatus nppsMul\_16s32f (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)**

16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.253 NppStatus nppsMul\_16s32s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.254 NppStatus nppsMul\_16s\_I (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.255 NppStatus nppsMul\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.256 NppStatus nppsMul\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.257 NppStatus nppsMul\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.258 NppStatus nppsMul\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.259 NppStatus nppsMul\_16u16s\_Sfs (const Npp16u \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.260 NppStatus nppsMul\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.261 **NppStatus nppsMul\_16u\_Sfs** (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal time signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.262 **NppStatus nppsMul\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal times signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.263 **NppStatus nppsMul\_32f32fc** (const Npp32f \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)

32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.264 NppStatus nppsMul\_32f32fc\_I (const Npp32f \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.265 NppStatus nppsMul\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.266 NppStatus nppsMul\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.267 NppStatus nppsMul\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.268 NppStatus nppsMul\_32s32sc\_ISfs (const Npp32s \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.269 NppStatus nppsMul\_32s32sc\_Sfs (const Npp32s \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.270 NppStatus nppsMul\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.271 NppStatus nppsMul\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.272 NppStatus nppsMul\_32sc\_ISfs (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.273 **NppStatus nppsMul\_32sc\_Sfs** (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.274 **NppStatus nppsMul\_64f** (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal times signal, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.275 **NppStatus nppsMul\_64f\_I** (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point in place signal times signal, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.276 NppStatus nppsMul\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.277 NppStatus nppsMul\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.278 NppStatus nppsMul\_8u16u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.279 **NppStatus nppsMul\_8u\_ISfs** (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.280 **NppStatus nppsMul\_8u\_Sfs** (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal times signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.281 **NppStatus nppsMul\_Low\_32s\_Sfs** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.282 NppStatus nppsMulC\_16s\_ISfs (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.283 NppStatus nppsMulC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.284 NppStatus nppsMulC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.285 NppStatus nppsMulC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.286 NppStatus nppsMulC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.287 NppStatus nppsMulC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.288 NppStatus nppsMulC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.289 NppStatus nppsMulC\_32f16s\_Sfs (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nScaleFactor* [Integer Result Scaling](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.290 NppStatus nppsMulC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.291 **NppStatus nppsMulC\_32fc** (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.292 **NppStatus nppsMulC\_32fc\_I** (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.293 **NppStatus nppsMulC\_32s\_ISfs** (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal times constant and scale.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.294 NppStatus nppsMulC\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.295 NppStatus nppsMulC\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.296 NppStatus nppsMulC\_32sc\_Sfs (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.297 NppStatus nppsMulC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.298 NppStatus nppsMulC\_64f64s\_ISfs (Npp64f *nValue*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.

**Parameters:**

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.299 NppStatus nppsMulC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point, in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.300 NppStatus nppsMulC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.301 NppStatus nppsMulC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.302 NppStatus nppsMulC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal times constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.303 **NppStatus nppsMulC\_8u\_Sfs** (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal times constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.304 **NppStatus nppsMulC\_Low\_32f16s** (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp16s \* *pDst*, int *nLength*)

32-bit floating point signal times constant with output converted to 16-bit signed integer.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.305 **NppStatus nppsNormalize\_16s\_Sfs** (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, Npp16s *vSub*, int *vDiv*, int *nScaleFactor*)

16-bit signed short signal normalize, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.306 NppStatus nppsNormalize\_16sc\_Sfs** (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16sc *vSub*, int *vDiv*, int *nScaleFactor*)

16-bit complex signed short signal normalize, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.307 NppStatus nppsNormalize\_32f** (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *vSub*, Npp32f *vDiv*)

32-bit floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.308 NppStatus nppsNormalize\_32fc** (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32fc *vSub*, Npp32fc *vDiv*)

32-bit complex floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.309 NppStatus nppsNormalize\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *vSub*, Npp64f *vDiv*)**

64-bit floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.310 NppStatus nppsNormalize\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*, Npp64fc *vSub*, Npp64fc *vDiv*)**

64-bit complex floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.311 NppStatus nppsNot\_16u (const Npp16u \* *pSrc*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short not signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.312 NppStatus nppsNot\_16u\_I (Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.313 NppStatus nppsNot\_32u (const Npp32u \* pSrc, Npp32u \* pDst, int nLength)**

32-bit unsigned integer not signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.314 NppStatus nppsNot\_32u\_I (Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.315 NppStatus nppsNot\_8u (const Npp8u \* pSrc, Npp8u \* pDst, int nLength)**

8-bit unsigned char not signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.316 NppStatus nppsNot\_8u\_I (Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place not signal.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.317 NppStatus nppsOr\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.318 NppStatus nppsOr\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.319 NppStatus nppsOr\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).



*pSrc2* [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.320 NppStatus nppsOr\_32u\_I (const Npp32u \* pSrc, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned integer in place signal or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.321 NppStatus nppsOr\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.322 NppStatus nppsOr\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.323 NppStatus nppsOrC\_16u (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.324 NppStatus nppsOrC\_16u\_I (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.325 NppStatus nppsOrC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.326 NppStatus nppsOrC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.327 NppStatus nppsOrC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.328 NppStatus nppsOrC\_8u\_I (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.329 NppStatus nppsRShiftC\_16s (const Npp16s \* *pSrc*, int *nValue*, Npp16s \* *pDst*, int *nLength*)**

16-bit signed short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.330 NppStatus nppsRShiftC\_16s\_I (int nValue, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.331 NppStatus nppsRShiftC\_16u (const Npp16u \* pSrc, int nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.332 NppStatus nppsRShiftC\_16u\_I (int nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.333 NppStatus nppsRShiftC\_32s (const Npp32s \* *pSrc*, int *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.334 NppStatus nppsRShiftC\_32s\_I (int *nValue*, Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.335 NppStatus nppsRShiftC\_32u (const Npp32u \* *pSrc*, int *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.336 NppStatus nppsRShiftC\_32u\_I (int *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal right shift with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be used to right shift each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.337 NppStatus nppsRShiftC\_8u (const Npp8u \* *pSrc*, int *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal right shift with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be used to right shift each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.338 NppStatus nppsRShiftC\_8u\_I (int *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal right shift with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be used to right shift each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.339 NppStatus nppsSet\_16s (Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*)**

16-bit integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector *pDst*.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.340 NppStatus nppsSet\_16sc (Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*)**

16-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.341 NppStatus nppsSet\_32f (Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit float, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.342 NppStatus nppsSet\_32fc (Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit float complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.343 NppStatus nppsSet\_32s (Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.344 NppStatus nppsSet\_32sc (Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*)**

32-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.345 NppStatus nppsSet\_64f (Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit double, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.346 NppStatus nppsSet\_64fc (Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit double complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).



*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.347 NppStatus nppsSet\_64s (Npp64s *nValue*, Npp64s \* *pDst*, int *nLength*)**

64-bit long long integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.348 NppStatus nppsSet\_64sc (Npp64sc *nValue*, Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.349 NppStatus nppsSet\_8u (Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.350 NppStatus nppsSqr\_16s\_ISfs (Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.351 NppStatus nppsSqr\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.352 NppStatus nppsSqr\_16sc\_ISfs (Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.353 NppStatus nppsSqr\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.354 NppStatus nppsSqr\_16u\_ISfs (Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.355 NppStatus nppsSqr\_16u\_Sfs (const Npp16u \* pSrc, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.356 NppStatus nppsSqr\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.357 NppStatus nppsSqr\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.358 NppStatus nppsSqr\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.359 NppStatus nppsSqr\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.360 NppStatus nppsSqr\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.361 NppStatus nppsSqr\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.362 NppStatus nppsSqr\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.363 NppStatus nppsSqr\_64fc\_I (Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.364 NppStatus nppsSqr\_8u\_ISfs (Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.365 NppStatus nppsSqr\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.366 NppStatus nppsSqrt\_16s\_ISfs (Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.367 NppStatus nppsSqrt\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.368 NppStatus nppsSqrt\_16sc\_ISfs (Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.369 NppStatus nppsSqrt\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.370 NppStatus nppsSqrt\_16u\_ISfs (Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.371 NppStatus nppsSqrt\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.372 NppStatus nppsSqrt\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.373 NppStatus nppsSqrt\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.374 NppStatus nppsSqrt\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.11.1.375 NppStatus nppsSqrt\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.376 NppStatus nppsSqrt\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.377 NppStatus nppsSqrt\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.378 NppStatus nppsSqrt\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.379 NppStatus nppsSqrt\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.380 NppStatus nppsSqrt\_64fc\_I (Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.381 NppStatus nppsSqrt\_64s16s\_Sfs (const Npp64s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.382 NppStatus nppsSqrt\_64s\_ISfs (Npp64s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.383 NppStatus nppsSqrt\_64s\_Sfs (const Npp64s \* *pSrc*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.384 NppStatus nppsSqrt\_8u\_ISfs (Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.385 NppStatus nppsSqrt\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.386 **NppStatus nppsSub\_16s** (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*)

16-bit signed short signal subtract signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.387 **NppStatus nppsSub\_16s32f** (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.11.1.388 **NppStatus nppsSub\_16s\_I** (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)

16-bit signed short in place signal subtract signal, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.389 NppStatus nppsSub\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.390 NppStatus nppsSub\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.391 NppStatus nppsSub\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.392 NppStatus nppsSub\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.393 NppStatus nppsSub\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.11.1.394 NppStatus nppsSub\_16u\_Sfs (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.395 NppStatus nppsSub\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.396 NppStatus nppsSub\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.397 NppStatus nppsSub\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.398 NppStatus nppsSub\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.399 NppStatus nppsSub\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.400 NppStatus nppsSub\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.401 NppStatus nppsSub\_32sc\_ISfs (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.402 NppStatus nppsSub\_32sc\_Sfs (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.403 NppStatus nppsSub\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.404 NppStatus nppsSub\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.405 NppStatus nppsSub\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.406 NppStatus nppsSub\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.407 NppStatus nppsSub\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.408** `NppStatus nppsSub_8u_Sfs (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.409** `NppStatus nppsSubC_16s_ISfs (Npp16s nValue, Npp16s * pSrcDst, int nLength, int nScaleFactor)`

16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.410 **NppStatus nppsSubC\_16s\_Sfs** (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal subtract constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.411 **NppStatus nppsSubC\_16sc\_ISfs** (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.412 **NppStatus nppsSubC\_16sc\_Sfs** (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.413 NppStatus nppsSubC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.414 NppStatus nppsSubC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.415 NppStatus nppsSubC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.416 NppStatus nppsSubC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal subtract constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.417 NppStatus nppsSubC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.418 NppStatus nppsSubC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.419 NppStatus nppsSubC\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.420** `NppStatus nppsSubC_32s_Sfs (const Npp32s * pSrc, Npp32s nValue, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed integer signal subtract constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.421** `NppStatus nppsSubC_32sc_ISfs (Npp32sc nValue, Npp32sc * pSrcDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.422 **NppStatus nppsSubC\_32sc\_Sfs** (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.423 **NppStatus nppsSubC\_64f** (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal subtract constant.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.424 **NppStatus nppsSubC\_64f\_I** (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point, in place signal subtract constant.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* Length of the vectors, number of items.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.425 NppStatus nppsSubC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.426 NppStatus nppsSubC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.427 NppStatus nppsSubC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.428 NppStatus nppsSubC\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.429 NppStatus nppsSubCRev\_16s\_ISfs (Npp16s *nValue*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.430 NppStatus nppsSubCRev\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.431 NppStatus nppsSubCRev\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.432 NppStatus nppsSubCRev\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.433 NppStatus nppsSubCRev\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.434 NppStatus nppsSubCRev\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.435 NppStatus nppsSubCRev\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.436 NppStatus nppsSubCRev\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.437 NppStatus nppsSubCRev\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.438 NppStatus nppsSubCRev\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.439 NppStatus nppsSubCRev\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract from constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.11.1.440 **NppStatus nppsSubCRev\_32s\_Sfs** (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integersignal subtract from constant and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.441 **NppStatus nppsSubCRev\_32sc\_ISfs** (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.11.1.442 **NppStatus nppsSubCRev\_32sc\_Sfs** (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.443 NppStatus nppsSubCRev\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.444 NppStatus nppsSubCRev\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point, in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.445 NppStatus nppsSubCRev\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.446 NppStatus nppsSubCRev\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.447 NppStatus nppsSubCRev\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.448 NppStatus nppsSubCRev\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.11.1.449 NppStatus nppsSum\_16s32s\_Sfs (const Npp16s \* *pSrc*, int *nLength*, Npp32s \* *pSum*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit integer vector sum (32bit) with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

*nScaleFactor* Integer-result scale factor.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.450 NppStatus nppsSum\_16s\_Sfs (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pSum*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit short vector sum with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

*nScaleFactor* Integer-result scale factor.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.451 NppStatus nppsSum\_16sc32sc\_Sfs (const Npp16sc \* *pSrc*, int *nLength*, Npp32sc \* *pSum*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit short complex vector sum (32bit int complex) with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

*nScaleFactor* Integer-result scale factor.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.452 NppStatus nppsSum\_16sc\_Sfs (const Npp16sc \* *pSrc*, int *nLength*, Npp16sc \* *pSum*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit short complex vector sum with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

*nScaleFactor* Integer-result scale factor.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.453 NppStatus nppsSum\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pSum*, NppHintAlgorithm *eHint*, Npp8u \* *pDeviceBuffer*)**

32-bit float vector sum method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*eHint* Suggests using specific code.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.454 NppStatus nppsSum\_32fc (const Npp32fc \* *pSrc*, int *nLength*, Npp32fc \* *pSum*, NppHintAlgorithm *eHint*, Npp8u \* *pDeviceBuffer*)**

32-bit float complex vector sum method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*eHint* Suggests using specific code.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.455** `NppStatus nppsSum_32s_Sfs (const Npp32s * pSrc, int nLength, Npp32s * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

32-bit integer vector sum with integer scaling method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

*nScaleFactor* Integer-result scale factor.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.456** `NppStatus nppsSum_64f (const Npp64f * pSrc, int nLength, Npp64f * pSum, Npp8u * pDeviceBuffer)`

64-bit double vector sum method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.457** `NppStatus nppsSum_64fc (const Npp64fc * pSrc, int nLength, Npp64fc * pSum, Npp8u * pDeviceBuffer)`

64-bit double complex vector sum method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.458 NppStatus nppsSumGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16s32s\_Sfs.

This primitive provides the correct buffer size for nppsSum\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.459 NppStatus nppsSumGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16s\_Sfs Sum.

This primitive provides the correct buffer size for nppsSum\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.460 NppStatus nppsSumGetBufferSize\_16sc32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16sc32sc\_Sfs Sum.

This primitive provides the correct buffer size for nppsSum\_16sc32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.461 NppStatus nppsSumGetBufferSize\_16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16sc\_Sfs Sum.

This primitive provides the correct buffer size for nppsSum\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.462 NppStatus nppsSumGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32f Sum.

This primitive provides the correct buffer size for `nppsSum_32f`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.463 NppStatus nppsSumGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32fc Sum.

This primitive provides the correct buffer size for `nppsSum_32fc`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.464 NppStatus nppsSumGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32s\_Sfs Sum.

This primitive provides the correct buffer size for `nppsSum_32s_Sfs`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.465 NppStatus nppsSumGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64f Sum.

This primitive provides the correct buffer size for nppsSum\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.466 NppStatus nppsSumGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64fc Sum.

This primitive provides the correct buffer size for nppsSum\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.11.1.467 NppStatus nppsXor\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal exclusive or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.468 NppStatus nppsXor\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal exclusive or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.469 NppStatus nppsXor\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal exclusive or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.470 NppStatus nppsXor\_32u\_I (const Npp32u \* *pSrc*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned integer in place signal exclusive or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.11.1.471 NppStatus nppsXor\_8u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal exclusive or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal2 elements to be exclusive ored with signal1 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.472 NppStatus nppsXor\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal exclusive or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer. signal2 elements to be exclusive ored with signal1 elements  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.473 NppStatus nppsXorC\_16u (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal exclusive or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be exclusive ored with each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.474 NppStatus nppsXorC\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be exclusive ored with each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.11.1.475 NppStatus nppsXorC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal exclusive or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.476 NppStatus nppsXorC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.477 NppStatus nppsXorC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal exclusive or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.478 NppStatus nppsXorC\_8u\_I (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be exclusive ored with each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.479 NppStatus nppsZero\_16s (Npp16s \* *pDst*, int *nLength*)**

16-bit integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.480 NppStatus nppsZero\_16sc (Npp16sc \* *pDst*, int *nLength*)**

16-bit integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.481 NppStatus nppsZero\_32f (Npp32f \* *pDst*, int *nLength*)**

32-bit float, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.482 NppStatus nppsZero\_32fc (Npp32fc \* *pDst*, int *nLength*)**

32-bit float complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.483 NppStatus nppsZero\_32s (Npp32s \* *pDst*, int *nLength*)**

32-bit integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.484 NppStatus nppsZero\_32sc (Npp32sc \* *pDst*, int *nLength*)**

32-bit integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.485 NppStatus nppsZero\_64f (Npp64f \* *pDst*, int *nLength*)**

64-bit double, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.486 NppStatus nppsZero\_64fc (Npp64fc \* *pDst*, int *nLength*)**

64-bit double complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.487 NppStatus nppsZero\_64s (Npp64s \* *pDst*, int *nLength*)**

64-bit long long integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.488 NppStatus nppsZero\_64sc (Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.11.1.489 NppStatus nppsZero\_8u (Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

# Chapter 8

## Data Structure Documentation

### 8.1 Npp16sc Struct Reference

Complex Number This struct represents a short complex number.

```
#include <nppdefs.h>
```

#### Data Fields

- [Npp16s re](#)  
*Real part.*
- [Npp16s im](#)  
*Imaginary part.*

#### 8.1.1 Detailed Description

Complex Number This struct represents a short complex number.

#### 8.1.2 Field Documentation

##### 8.1.2.1 Npp16s Npp16sc::im

Imaginary part.

##### 8.1.2.2 Npp16s Npp16sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.2 Npp32fc Struct Reference

Complex Number This struct represents a single floating-point complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32f re](#)  
*Real part.*
- [Npp32f im](#)  
*Imaginary part.*

### 8.2.1 Detailed Description

Complex Number This struct represents a single floating-point complex number.

### 8.2.2 Field Documentation

#### 8.2.2.1 Npp32f Npp32fc::im

Imaginary part.

#### 8.2.2.2 Npp32f Npp32fc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.3 Npp32sc Struct Reference

Complex Number This struct represents a signed int complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32s re](#)  
*Real part.*
- [Npp32s im](#)  
*Imaginary part.*

### 8.3.1 Detailed Description

Complex Number This struct represents a signed int complex number.

### 8.3.2 Field Documentation

#### 8.3.2.1 Npp32s Npp32sc::im

Imaginary part.

#### 8.3.2.2 Npp32s Npp32sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.4 Npp64fc Struct Reference

Complex Number This struct represents a double floating-point complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp64f re](#)  
*Real part.*
- [Npp64f im](#)  
*Imaginary part.*

### 8.4.1 Detailed Description

Complex Number This struct represents a double floating-point complex number.

### 8.4.2 Field Documentation

#### 8.4.2.1 Npp64f Npp64fc::im

Imaginary part.

#### 8.4.2.2 Npp64f Npp64fc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h



## 8.5 Npp64sc Struct Reference

Complex Number This struct represents a long long complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp64s re](#)  
*Real part.*
- [Npp64s im](#)  
*Imaginary part.*

### 8.5.1 Detailed Description

Complex Number This struct represents a long long complex number.

### 8.5.2 Field Documentation

#### 8.5.2.1 Npp64s Npp64sc::im

Imaginary part.

#### 8.5.2.2 Npp64s Npp64sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.6 NppiHaarBuffer Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [haarBufferSize](#)  
*size of the buffer*
- [Npp32s](#) \* [haarBuffer](#)  
*buffer*

### 8.6.1 Field Documentation

#### 8.6.1.1 [Npp32s](#)\* [NppiHaarBuffer::haarBuffer](#)

*buffer*

#### 8.6.1.2 [int](#) [NppiHaarBuffer::haarBufferSize](#)

*size of the buffer*

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.7 NppiHaarClassifier\_32f Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [numClassifiers](#)  
*number of classifiers*
- [Npp32s](#) \* [classifiers](#)  
*packed classifier data 40 bytes each*
- [size\\_t](#) [classifierStep](#)
- [NppiSize](#) [classifierSize](#)
- [Npp32s](#) \* [counterDevice](#)

### 8.7.1 Field Documentation

#### 8.7.1.1 [Npp32s](#)\* [NppiHaarClassifier\\_32f::classifiers](#)

packed classifier data 40 bytes each

#### 8.7.1.2 [NppiSize](#) [NppiHaarClassifier\\_32f::classifierSize](#)

#### 8.7.1.3 [size\\_t](#) [NppiHaarClassifier\\_32f::classifierStep](#)

#### 8.7.1.4 [Npp32s](#)\* [NppiHaarClassifier\\_32f::counterDevice](#)

#### 8.7.1.5 [int](#) [NppiHaarClassifier\\_32f::numClassifiers](#)

number of classifiers

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.8 NppiPoint Struct Reference

2D Point

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate.*
- `int y`  
*y-coordinate.*

### 8.8.1 Detailed Description

2D Point

### 8.8.2 Field Documentation

#### 8.8.2.1 `int NppiPoint::x`

x-coordinate.

#### 8.8.2.2 `int NppiPoint::y`

y-coordinate.

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h`

## 8.9 NppiRect Struct Reference

2D Rectangle This struct contains position and size information of a rectangle in two space.

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate of upper left corner.*
- `int y`  
*y-coordinate of upper left corner.*
- `int width`  
*Rectangle width.*
- `int height`  
*Rectangle height.*

### 8.9.1 Detailed Description

2D Rectangle This struct contains position and size information of a rectangle in two space.

The rectangle's position is usually signified by the coordinate of its upper-left corner.

### 8.9.2 Field Documentation

#### 8.9.2.1 `int NppiRect::height`

Rectangle height.

#### 8.9.2.2 `int NppiRect::width`

Rectangle width.

#### 8.9.2.3 `int NppiRect::x`

x-coordinate of upper left corner.

#### 8.9.2.4 `int NppiRect::y`

y-coordinate of upper left corner.

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h`

## 8.10 NppiSize Struct Reference

2D Size This struct typically represents the size of a rectangular region in two space.

```
#include <nppdefs.h>
```

### Data Fields

- int [width](#)  
*Rectangle width.*
- int [height](#)  
*Rectangle height.*

### 8.10.1 Detailed Description

2D Size This struct typically represents the size of a rectangular region in two space.

### 8.10.2 Field Documentation

#### 8.10.2.1 int NppiSize::height

Rectangle height.

#### 8.10.2.2 int NppiSize::width

Rectangle width.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h

## 8.11 NppLibraryVersion Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [major](#)  
*Major version number.*
- int [minor](#)  
*Minor version number.*
- int [build](#)  
*Build number. This reflects the nightly build this release was made from.*

### 8.11.1 Field Documentation

#### 8.11.1.1 int NppLibraryVersion::build

Build number. This reflects the nightly build this release was made from.

#### 8.11.1.2 int NppLibraryVersion::major

Major version number.

#### 8.11.1.3 int NppLibraryVersion::minor

Minor version number.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r4.1/NPP/npp/include/nppdefs.h