

NVIDIA Performance Primitives (NPP)
Version 8.0

January 28, 2016

Contents

1	NVIDIA Performance Primitives	1
1.1	What is NPP?	2
1.2	Documentation	2
1.3	Technical Specifications	3
1.4	Files	3
1.4.1	Header Files	3
1.4.2	Library Files	3
1.5	Supported NVIDIA Hardware	4
2	General API Conventions	5
2.1	Memory Management	6
2.1.1	Scratch Buffer and Host Pointer	6
2.2	Function Naming	7
2.3	Integer Result Scaling	7
2.4	Rounding Modes	8
2.4.1	Rounding Mode Parameter	8
3	Signal-Processing Specific API Conventions	9
3.1	Signal Data	10
3.1.1	Parameter Names for Signal Data	10
3.1.1.1	Source Signal Pointer	10
3.1.1.2	Destination Signal Pointer	10
3.1.1.3	In-Place Signal Pointer	10
3.1.2	Signal Data Alignment Requirements	11
3.1.3	Signal Data Related Error Codes	11
3.2	Signal Length	11
3.2.1	Length Related Error Codes	11
4	Imaging-Processing Specific API Conventions	13

4.1	Function Naming	14
4.2	Image Data	14
4.2.1	Line Step	15
4.2.2	Parameter Names for Image Data	15
4.2.2.1	Passing Source-Image Data	15
4.2.2.2	Passing Destination-Image Data	16
4.2.2.3	Passing In-Place Image Data	18
4.2.2.4	Passing Mask-Image Data	18
4.2.2.5	Passing Channel-of-Interest Data	18
4.2.3	Image Data Alignment Requirements	18
4.2.4	Image Data Related Error Codes	19
4.3	Region-of-Interest (ROI)	19
4.3.1	ROI Related Error Codes	19
4.4	Masked Operation	20
4.5	Channel-of-Interest API	20
4.5.1	Select-Channel Source-Image Pointer	20
4.5.2	Select-Channel Source-Image	20
4.5.3	Select-Channel Destination-Image Pointer	20
4.6	Source-Image Sampling	21
4.6.1	Point-Wise Operations	21
4.6.2	Neighborhood Operations	21
4.6.2.1	Mask-Size Parameter	21
4.6.2.2	Anchor-Point Parameter	22
4.6.2.3	Sampling Beyond Image Boundaries	22
5	Module Index	23
5.1	Modules	23
6	Data Structure Index	25
6.1	Data Structures	25
7	Module Documentation	27
7.1	NPP Core	27
7.1.1	Detailed Description	28
7.1.2	Function Documentation	28
7.1.2.1	nppGetGpuComputeCapability	28
7.1.2.2	nppGetGpuDeviceProperties	28
7.1.2.3	nppGetGpuName	28

7.1.2.4	nppGetGpuNumSMs	28
7.1.2.5	nppGetLibVersion	29
7.1.2.6	nppGetMaxThreadsPerBlock	29
7.1.2.7	nppGetMaxThreadsPerSM	29
7.1.2.8	nppGetStream	29
7.1.2.9	nppGetStreamMaxThreadsPerSM	29
7.1.2.10	nppGetStreamNumSMs	29
7.1.2.11	nppSetStream	30
7.2	NPP Type Definitions and Constants	31
7.2.1	Define Documentation	37
7.2.1.1	NPP_MAX_16S	37
7.2.1.2	NPP_MAX_16U	37
7.2.1.3	NPP_MAX_32S	37
7.2.1.4	NPP_MAX_32U	37
7.2.1.5	NPP_MAX_64S	37
7.2.1.6	NPP_MAX_64U	37
7.2.1.7	NPP_MAX_8S	37
7.2.1.8	NPP_MAX_8U	37
7.2.1.9	NPP_MAXABS_32F	37
7.2.1.10	NPP_MAXABS_64F	37
7.2.1.11	NPP_MIN_16S	38
7.2.1.12	NPP_MIN_16U	38
7.2.1.13	NPP_MIN_32S	38
7.2.1.14	NPP_MIN_32U	38
7.2.1.15	NPP_MIN_64S	38
7.2.1.16	NPP_MIN_64U	38
7.2.1.17	NPP_MIN_8S	38
7.2.1.18	NPP_MIN_8U	38
7.2.1.19	NPP_MINABS_32F	38
7.2.1.20	NPP_MINABS_64F	38
7.2.2	Enumeration Type Documentation	38
7.2.2.1	NppCmpOp	38
7.2.2.2	NppGpuComputeCapability	39
7.2.2.3	NppHintAlgorithm	39
7.2.2.4	NppiAlphaOp	39
7.2.2.5	NppiAxis	40

7.2.2.6	<code>NppiBayerGridPosition</code>	40
7.2.2.7	<code>NppiBorderType</code>	40
7.2.2.8	<code>NppiDifferentialKernel</code>	40
7.2.2.9	<code>NppiHuffmanTableType</code>	41
7.2.2.10	<code>NppiInterpolationMode</code>	41
7.2.2.11	<code>NppiMaskSize</code>	41
7.2.2.12	<code>NppiNorm</code>	42
7.2.2.13	<code>NppRoundMode</code>	42
7.2.2.14	<code>NppStatus</code>	43
7.2.2.15	<code>NppsZCType</code>	45
7.3	Basic NPP Data Types	46
7.3.1	Typedef Documentation	47
7.3.1.1	<code>Npp16s</code>	47
7.3.1.2	<code>Npp16u</code>	47
7.3.1.3	<code>Npp32f</code>	47
7.3.1.4	<code>Npp32fc</code>	47
7.3.1.5	<code>Npp32s</code>	47
7.3.1.6	<code>Npp32sc</code>	48
7.3.1.7	<code>Npp32u</code>	48
7.3.1.8	<code>Npp32uc</code>	48
7.3.1.9	<code>Npp64f</code>	48
7.3.1.10	<code>Npp64fc</code>	48
7.3.1.11	<code>Npp64s</code>	48
7.3.1.12	<code>Npp64sc</code>	48
7.3.1.13	<code>Npp64u</code>	48
7.3.1.14	<code>Npp8s</code>	48
7.3.1.15	<code>Npp8u</code>	48
7.3.2	Function Documentation	48
7.3.2.1	<code>__align__</code>	48
7.3.2.2	<code>__align__</code>	49
7.3.3	Variable Documentation	49
7.3.3.1	<code>Npp16sc</code>	49
7.3.3.2	<code>Npp16uc</code>	49
7.3.3.3	<code>Npp8uc</code>	49
7.4	Statistical Operations	50
7.4.1	Detailed Description	66

7.4.2	Function Documentation	66
7.4.2.1	nppiAverageErrorGetBufferHostSize_16s_C1R	66
7.4.2.2	nppiAverageErrorGetBufferHostSize_16s_C2R	66
7.4.2.3	nppiAverageErrorGetBufferHostSize_16s_C3R	66
7.4.2.4	nppiAverageErrorGetBufferHostSize_16s_C4R	67
7.4.2.5	nppiAverageErrorGetBufferHostSize_16sc_C1R	67
7.4.2.6	nppiAverageErrorGetBufferHostSize_16sc_C2R	67
7.4.2.7	nppiAverageErrorGetBufferHostSize_16sc_C3R	68
7.4.2.8	nppiAverageErrorGetBufferHostSize_16sc_C4R	68
7.4.2.9	nppiAverageErrorGetBufferHostSize_16u_C1R	68
7.4.2.10	nppiAverageErrorGetBufferHostSize_16u_C2R	68
7.4.2.11	nppiAverageErrorGetBufferHostSize_16u_C3R	69
7.4.2.12	nppiAverageErrorGetBufferHostSize_16u_C4R	69
7.4.2.13	nppiAverageErrorGetBufferHostSize_32f_C1R	69
7.4.2.14	nppiAverageErrorGetBufferHostSize_32f_C2R	70
7.4.2.15	nppiAverageErrorGetBufferHostSize_32f_C3R	70
7.4.2.16	nppiAverageErrorGetBufferHostSize_32f_C4R	70
7.4.2.17	nppiAverageErrorGetBufferHostSize_32fc_C1R	70
7.4.2.18	nppiAverageErrorGetBufferHostSize_32fc_C2R	71
7.4.2.19	nppiAverageErrorGetBufferHostSize_32fc_C3R	71
7.4.2.20	nppiAverageErrorGetBufferHostSize_32fc_C4R	71
7.4.2.21	nppiAverageErrorGetBufferHostSize_32s_C1R	72
7.4.2.22	nppiAverageErrorGetBufferHostSize_32s_C2R	72
7.4.2.23	nppiAverageErrorGetBufferHostSize_32s_C3R	72
7.4.2.24	nppiAverageErrorGetBufferHostSize_32s_C4R	72
7.4.2.25	nppiAverageErrorGetBufferHostSize_32sc_C1R	73
7.4.2.26	nppiAverageErrorGetBufferHostSize_32sc_C2R	73
7.4.2.27	nppiAverageErrorGetBufferHostSize_32sc_C3R	73
7.4.2.28	nppiAverageErrorGetBufferHostSize_32sc_C4R	74
7.4.2.29	nppiAverageErrorGetBufferHostSize_32u_C1R	74
7.4.2.30	nppiAverageErrorGetBufferHostSize_32u_C2R	74
7.4.2.31	nppiAverageErrorGetBufferHostSize_32u_C3R	74
7.4.2.32	nppiAverageErrorGetBufferHostSize_32u_C4R	75
7.4.2.33	nppiAverageErrorGetBufferHostSize_64f_C1R	75
7.4.2.34	nppiAverageErrorGetBufferHostSize_64f_C2R	75
7.4.2.35	nppiAverageErrorGetBufferHostSize_64f_C3R	76

7.4.2.36	<code>nppiAverageErrorGetBufferHostSize_64f_C4R</code>	76
7.4.2.37	<code>nppiAverageErrorGetBufferHostSize_8s_C1R</code>	76
7.4.2.38	<code>nppiAverageErrorGetBufferHostSize_8s_C2R</code>	76
7.4.2.39	<code>nppiAverageErrorGetBufferHostSize_8s_C3R</code>	77
7.4.2.40	<code>nppiAverageErrorGetBufferHostSize_8s_C4R</code>	77
7.4.2.41	<code>nppiAverageErrorGetBufferHostSize_8u_C1R</code>	77
7.4.2.42	<code>nppiAverageErrorGetBufferHostSize_8u_C2R</code>	78
7.4.2.43	<code>nppiAverageErrorGetBufferHostSize_8u_C3R</code>	78
7.4.2.44	<code>nppiAverageErrorGetBufferHostSize_8u_C4R</code>	78
7.4.2.45	<code>nppiAverageRelativeErrorGetBufferHostSize_16s_C1R</code>	78
7.4.2.46	<code>nppiAverageRelativeErrorGetBufferHostSize_16s_C2R</code>	79
7.4.2.47	<code>nppiAverageRelativeErrorGetBufferHostSize_16s_C3R</code>	79
7.4.2.48	<code>nppiAverageRelativeErrorGetBufferHostSize_16s_C4R</code>	79
7.4.2.49	<code>nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R</code>	80
7.4.2.50	<code>nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R</code>	80
7.4.2.51	<code>nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R</code>	80
7.4.2.52	<code>nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R</code>	80
7.4.2.53	<code>nppiAverageRelativeErrorGetBufferHostSize_16u_C1R</code>	81
7.4.2.54	<code>nppiAverageRelativeErrorGetBufferHostSize_16u_C2R</code>	81
7.4.2.55	<code>nppiAverageRelativeErrorGetBufferHostSize_16u_C3R</code>	81
7.4.2.56	<code>nppiAverageRelativeErrorGetBufferHostSize_16u_C4R</code>	82
7.4.2.57	<code>nppiAverageRelativeErrorGetBufferHostSize_32f_C1R</code>	82
7.4.2.58	<code>nppiAverageRelativeErrorGetBufferHostSize_32f_C2R</code>	82
7.4.2.59	<code>nppiAverageRelativeErrorGetBufferHostSize_32f_C3R</code>	82
7.4.2.60	<code>nppiAverageRelativeErrorGetBufferHostSize_32f_C4R</code>	83
7.4.2.61	<code>nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R</code>	83
7.4.2.62	<code>nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R</code>	83
7.4.2.63	<code>nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R</code>	84
7.4.2.64	<code>nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R</code>	84
7.4.2.65	<code>nppiAverageRelativeErrorGetBufferHostSize_32s_C1R</code>	84
7.4.2.66	<code>nppiAverageRelativeErrorGetBufferHostSize_32s_C2R</code>	84
7.4.2.67	<code>nppiAverageRelativeErrorGetBufferHostSize_32s_C3R</code>	85
7.4.2.68	<code>nppiAverageRelativeErrorGetBufferHostSize_32s_C4R</code>	85
7.4.2.69	<code>nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R</code>	85
7.4.2.70	<code>nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R</code>	86
7.4.2.71	<code>nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R</code>	86

7.4.2.72	nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R	86
7.4.2.73	nppiAverageRelativeErrorGetBufferHostSize_32u_C1R	86
7.4.2.74	nppiAverageRelativeErrorGetBufferHostSize_32u_C2R	87
7.4.2.75	nppiAverageRelativeErrorGetBufferHostSize_32u_C3R	87
7.4.2.76	nppiAverageRelativeErrorGetBufferHostSize_32u_C4R	87
7.4.2.77	nppiAverageRelativeErrorGetBufferHostSize_64f_C1R	88
7.4.2.78	nppiAverageRelativeErrorGetBufferHostSize_64f_C2R	88
7.4.2.79	nppiAverageRelativeErrorGetBufferHostSize_64f_C3R	88
7.4.2.80	nppiAverageRelativeErrorGetBufferHostSize_64f_C4R	88
7.4.2.81	nppiAverageRelativeErrorGetBufferHostSize_8s_C1R	89
7.4.2.82	nppiAverageRelativeErrorGetBufferHostSize_8s_C2R	89
7.4.2.83	nppiAverageRelativeErrorGetBufferHostSize_8s_C3R	89
7.4.2.84	nppiAverageRelativeErrorGetBufferHostSize_8s_C4R	90
7.4.2.85	nppiAverageRelativeErrorGetBufferHostSize_8u_C1R	90
7.4.2.86	nppiAverageRelativeErrorGetBufferHostSize_8u_C2R	90
7.4.2.87	nppiAverageRelativeErrorGetBufferHostSize_8u_C3R	90
7.4.2.88	nppiAverageRelativeErrorGetBufferHostSize_8u_C4R	91
7.4.2.89	nppiMaximumErrorGetBufferHostSize_16s_C1R	91
7.4.2.90	nppiMaximumErrorGetBufferHostSize_16s_C2R	91
7.4.2.91	nppiMaximumErrorGetBufferHostSize_16s_C3R	92
7.4.2.92	nppiMaximumErrorGetBufferHostSize_16s_C4R	92
7.4.2.93	nppiMaximumErrorGetBufferHostSize_16sc_C1R	92
7.4.2.94	nppiMaximumErrorGetBufferHostSize_16sc_C2R	92
7.4.2.95	nppiMaximumErrorGetBufferHostSize_16sc_C3R	93
7.4.2.96	nppiMaximumErrorGetBufferHostSize_16sc_C4R	93
7.4.2.97	nppiMaximumErrorGetBufferHostSize_16u_C1R	93
7.4.2.98	nppiMaximumErrorGetBufferHostSize_16u_C2R	94
7.4.2.99	nppiMaximumErrorGetBufferHostSize_16u_C3R	94
7.4.2.100	nppiMaximumErrorGetBufferHostSize_16u_C4R	94
7.4.2.101	nppiMaximumErrorGetBufferHostSize_32f_C1R	94
7.4.2.102	nppiMaximumErrorGetBufferHostSize_32f_C2R	95
7.4.2.103	nppiMaximumErrorGetBufferHostSize_32f_C3R	95
7.4.2.104	nppiMaximumErrorGetBufferHostSize_32f_C4R	95
7.4.2.105	nppiMaximumErrorGetBufferHostSize_32fc_C1R	96
7.4.2.106	nppiMaximumErrorGetBufferHostSize_32fc_C2R	96
7.4.2.107	nppiMaximumErrorGetBufferHostSize_32fc_C3R	96

7.4.2.108	<code>nppiMaximumErrorGetBufferHostSize_32fc_C4R</code>	96
7.4.2.109	<code>nppiMaximumErrorGetBufferHostSize_32s_C1R</code>	97
7.4.2.110	<code>nppiMaximumErrorGetBufferHostSize_32s_C2R</code>	97
7.4.2.111	<code>nppiMaximumErrorGetBufferHostSize_32s_C3R</code>	97
7.4.2.112	<code>nppiMaximumErrorGetBufferHostSize_32s_C4R</code>	98
7.4.2.113	<code>nppiMaximumErrorGetBufferHostSize_32sc_C1R</code>	98
7.4.2.114	<code>nppiMaximumErrorGetBufferHostSize_32sc_C2R</code>	98
7.4.2.115	<code>nppiMaximumErrorGetBufferHostSize_32sc_C3R</code>	98
7.4.2.116	<code>nppiMaximumErrorGetBufferHostSize_32sc_C4R</code>	99
7.4.2.117	<code>nppiMaximumErrorGetBufferHostSize_32u_C1R</code>	99
7.4.2.118	<code>nppiMaximumErrorGetBufferHostSize_32u_C2R</code>	99
7.4.2.119	<code>nppiMaximumErrorGetBufferHostSize_32u_C3R</code>	100
7.4.2.120	<code>nppiMaximumErrorGetBufferHostSize_32u_C4R</code>	100
7.4.2.121	<code>nppiMaximumErrorGetBufferHostSize_64f_C1R</code>	100
7.4.2.122	<code>nppiMaximumErrorGetBufferHostSize_64f_C2R</code>	100
7.4.2.123	<code>nppiMaximumErrorGetBufferHostSize_64f_C3R</code>	101
7.4.2.124	<code>nppiMaximumErrorGetBufferHostSize_64f_C4R</code>	101
7.4.2.125	<code>nppiMaximumErrorGetBufferHostSize_8s_C1R</code>	101
7.4.2.126	<code>nppiMaximumErrorGetBufferHostSize_8s_C2R</code>	102
7.4.2.127	<code>nppiMaximumErrorGetBufferHostSize_8s_C3R</code>	102
7.4.2.128	<code>nppiMaximumErrorGetBufferHostSize_8s_C4R</code>	102
7.4.2.129	<code>nppiMaximumErrorGetBufferHostSize_8u_C1R</code>	102
7.4.2.130	<code>nppiMaximumErrorGetBufferHostSize_8u_C2R</code>	103
7.4.2.131	<code>nppiMaximumErrorGetBufferHostSize_8u_C3R</code>	103
7.4.2.132	<code>nppiMaximumErrorGetBufferHostSize_8u_C4R</code>	103
7.4.2.133	<code>nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R</code>	104
7.4.2.134	<code>nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R</code>	104
7.4.2.135	<code>nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R</code>	104
7.4.2.136	<code>nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R</code>	104
7.4.2.137	<code>nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R</code>	105
7.4.2.138	<code>nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R</code>	105
7.4.2.139	<code>nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R</code>	105
7.4.2.140	<code>nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R</code>	106
7.4.2.141	<code>nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R</code>	106
7.4.2.142	<code>nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R</code>	106
7.4.2.143	<code>nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R</code>	106

7.4.2.144	<code>nppiMaximumRelativeErrorGetBufferSize_16u_C4R</code>	107
7.4.2.145	<code>nppiMaximumRelativeErrorGetBufferSize_32f_C1R</code>	107
7.4.2.146	<code>nppiMaximumRelativeErrorGetBufferSize_32f_C2R</code>	107
7.4.2.147	<code>nppiMaximumRelativeErrorGetBufferSize_32f_C3R</code>	108
7.4.2.148	<code>nppiMaximumRelativeErrorGetBufferSize_32f_C4R</code>	108
7.4.2.149	<code>nppiMaximumRelativeErrorGetBufferSize_32fc_C1R</code>	108
7.4.2.150	<code>nppiMaximumRelativeErrorGetBufferSize_32fc_C2R</code>	108
7.4.2.151	<code>nppiMaximumRelativeErrorGetBufferSize_32fc_C3R</code>	109
7.4.2.152	<code>nppiMaximumRelativeErrorGetBufferSize_32fc_C4R</code>	109
7.4.2.153	<code>nppiMaximumRelativeErrorGetBufferSize_32s_C1R</code>	109
7.4.2.154	<code>nppiMaximumRelativeErrorGetBufferSize_32s_C2R</code>	110
7.4.2.155	<code>nppiMaximumRelativeErrorGetBufferSize_32s_C3R</code>	110
7.4.2.156	<code>nppiMaximumRelativeErrorGetBufferSize_32s_C4R</code>	110
7.4.2.157	<code>nppiMaximumRelativeErrorGetBufferSize_32sc_C1R</code>	110
7.4.2.158	<code>nppiMaximumRelativeErrorGetBufferSize_32sc_C2R</code>	111
7.4.2.159	<code>nppiMaximumRelativeErrorGetBufferSize_32sc_C3R</code>	111
7.4.2.160	<code>nppiMaximumRelativeErrorGetBufferSize_32sc_C4R</code>	111
7.4.2.161	<code>nppiMaximumRelativeErrorGetBufferSize_32u_C1R</code>	112
7.4.2.162	<code>nppiMaximumRelativeErrorGetBufferSize_32u_C2R</code>	112
7.4.2.163	<code>nppiMaximumRelativeErrorGetBufferSize_32u_C3R</code>	112
7.4.2.164	<code>nppiMaximumRelativeErrorGetBufferSize_32u_C4R</code>	112
7.4.2.165	<code>nppiMaximumRelativeErrorGetBufferSize_64f_C1R</code>	113
7.4.2.166	<code>nppiMaximumRelativeErrorGetBufferSize_64f_C2R</code>	113
7.4.2.167	<code>nppiMaximumRelativeErrorGetBufferSize_64f_C3R</code>	113
7.4.2.168	<code>nppiMaximumRelativeErrorGetBufferSize_64f_C4R</code>	114
7.4.2.169	<code>nppiMaximumRelativeErrorGetBufferSize_8s_C1R</code>	114
7.4.2.170	<code>nppiMaximumRelativeErrorGetBufferSize_8s_C2R</code>	114
7.4.2.171	<code>nppiMaximumRelativeErrorGetBufferSize_8s_C3R</code>	114
7.4.2.172	<code>nppiMaximumRelativeErrorGetBufferSize_8s_C4R</code>	115
7.4.2.173	<code>nppiMaximumRelativeErrorGetBufferSize_8u_C1R</code>	115
7.4.2.174	<code>nppiMaximumRelativeErrorGetBufferSize_8u_C2R</code>	115
7.4.2.175	<code>nppiMaximumRelativeErrorGetBufferSize_8u_C3R</code>	116
7.4.2.176	<code>nppiMaximumRelativeErrorGetBufferSize_8u_C4R</code>	116
7.5	Sum	117
7.5.1	Detailed Description	119
7.5.2	Function Documentation	120

7.5.2.1	nppiSum_16s_AC4R	120
7.5.2.2	nppiSum_16s_C1R	120
7.5.2.3	nppiSum_16s_C3R	120
7.5.2.4	nppiSum_16s_C4R	121
7.5.2.5	nppiSum_16u_AC4R	121
7.5.2.6	nppiSum_16u_C1R	121
7.5.2.7	nppiSum_16u_C3R	122
7.5.2.8	nppiSum_16u_C4R	122
7.5.2.9	nppiSum_32f_AC4R	123
7.5.2.10	nppiSum_32f_C1R	123
7.5.2.11	nppiSum_32f_C3R	123
7.5.2.12	nppiSum_32f_C4R	124
7.5.2.13	nppiSum_8u64s_C1R	124
7.5.2.14	nppiSum_8u64s_C4R	124
7.5.2.15	nppiSum_8u_AC4R	125
7.5.2.16	nppiSum_8u_C1R	125
7.5.2.17	nppiSum_8u_C3R	126
7.5.2.18	nppiSum_8u_C4R	126
7.5.2.19	nppiSumGetBufferHostSize_16s_AC4R	126
7.5.2.20	nppiSumGetBufferHostSize_16s_C1R	127
7.5.2.21	nppiSumGetBufferHostSize_16s_C3R	127
7.5.2.22	nppiSumGetBufferHostSize_16s_C4R	127
7.5.2.23	nppiSumGetBufferHostSize_16u_AC4R	127
7.5.2.24	nppiSumGetBufferHostSize_16u_C1R	128
7.5.2.25	nppiSumGetBufferHostSize_16u_C3R	128
7.5.2.26	nppiSumGetBufferHostSize_16u_C4R	128
7.5.2.27	nppiSumGetBufferHostSize_32f_AC4R	128
7.5.2.28	nppiSumGetBufferHostSize_32f_C1R	129
7.5.2.29	nppiSumGetBufferHostSize_32f_C3R	129
7.5.2.30	nppiSumGetBufferHostSize_32f_C4R	129
7.5.2.31	nppiSumGetBufferHostSize_8u64s_C1R	130
7.5.2.32	nppiSumGetBufferHostSize_8u64s_C4R	130
7.5.2.33	nppiSumGetBufferHostSize_8u_AC4R	130
7.5.2.34	nppiSumGetBufferHostSize_8u_C1R	130
7.5.2.35	nppiSumGetBufferHostSize_8u_C3R	131
7.5.2.36	nppiSumGetBufferHostSize_8u_C4R	131

7.6	Min	132
7.6.1	Detailed Description	134
7.6.2	Function Documentation	134
7.6.2.1	nppiMin_16s_AC4R	134
7.6.2.2	nppiMin_16s_C1R	135
7.6.2.3	nppiMin_16s_C3R	135
7.6.2.4	nppiMin_16s_C4R	135
7.6.2.5	nppiMin_16u_AC4R	136
7.6.2.6	nppiMin_16u_C1R	136
7.6.2.7	nppiMin_16u_C3R	136
7.6.2.8	nppiMin_16u_C4R	137
7.6.2.9	nppiMin_32f_AC4R	137
7.6.2.10	nppiMin_32f_C1R	137
7.6.2.11	nppiMin_32f_C3R	138
7.6.2.12	nppiMin_32f_C4R	138
7.6.2.13	nppiMin_8u_AC4R	139
7.6.2.14	nppiMin_8u_C1R	139
7.6.2.15	nppiMin_8u_C3R	139
7.6.2.16	nppiMin_8u_C4R	140
7.6.2.17	nppiMinGetBufferHostSize_16s_AC4R	140
7.6.2.18	nppiMinGetBufferHostSize_16s_C1R	140
7.6.2.19	nppiMinGetBufferHostSize_16s_C3R	141
7.6.2.20	nppiMinGetBufferHostSize_16s_C4R	141
7.6.2.21	nppiMinGetBufferHostSize_16u_AC4R	141
7.6.2.22	nppiMinGetBufferHostSize_16u_C1R	141
7.6.2.23	nppiMinGetBufferHostSize_16u_C3R	142
7.6.2.24	nppiMinGetBufferHostSize_16u_C4R	142
7.6.2.25	nppiMinGetBufferHostSize_32f_AC4R	142
7.6.2.26	nppiMinGetBufferHostSize_32f_C1R	142
7.6.2.27	nppiMinGetBufferHostSize_32f_C3R	143
7.6.2.28	nppiMinGetBufferHostSize_32f_C4R	143
7.6.2.29	nppiMinGetBufferHostSize_8u_AC4R	143
7.6.2.30	nppiMinGetBufferHostSize_8u_C1R	143
7.6.2.31	nppiMinGetBufferHostSize_8u_C3R	144
7.6.2.32	nppiMinGetBufferHostSize_8u_C4R	144
7.7	MinIdx	145

7.7.1	Detailed Description	147
7.7.2	Function Documentation	147
7.7.2.1	nppiMinIndx_16s_AC4R	147
7.7.2.2	nppiMinIndx_16s_C1R	148
7.7.2.3	nppiMinIndx_16s_C3R	148
7.7.2.4	nppiMinIndx_16s_C4R	149
7.7.2.5	nppiMinIndx_16u_AC4R	149
7.7.2.6	nppiMinIndx_16u_C1R	149
7.7.2.7	nppiMinIndx_16u_C3R	150
7.7.2.8	nppiMinIndx_16u_C4R	150
7.7.2.9	nppiMinIndx_32f_AC4R	151
7.7.2.10	nppiMinIndx_32f_C1R	151
7.7.2.11	nppiMinIndx_32f_C3R	151
7.7.2.12	nppiMinIndx_32f_C4R	152
7.7.2.13	nppiMinIndx_8u_AC4R	152
7.7.2.14	nppiMinIndx_8u_C1R	153
7.7.2.15	nppiMinIndx_8u_C3R	153
7.7.2.16	nppiMinIndx_8u_C4R	153
7.7.2.17	nppiMinIndxGetBufferHostSize_16s_AC4R	154
7.7.2.18	nppiMinIndxGetBufferHostSize_16s_C1R	154
7.7.2.19	nppiMinIndxGetBufferHostSize_16s_C3R	154
7.7.2.20	nppiMinIndxGetBufferHostSize_16s_C4R	155
7.7.2.21	nppiMinIndxGetBufferHostSize_16u_AC4R	155
7.7.2.22	nppiMinIndxGetBufferHostSize_16u_C1R	155
7.7.2.23	nppiMinIndxGetBufferHostSize_16u_C3R	156
7.7.2.24	nppiMinIndxGetBufferHostSize_16u_C4R	156
7.7.2.25	nppiMinIndxGetBufferHostSize_32f_AC4R	156
7.7.2.26	nppiMinIndxGetBufferHostSize_32f_C1R	156
7.7.2.27	nppiMinIndxGetBufferHostSize_32f_C3R	157
7.7.2.28	nppiMinIndxGetBufferHostSize_32f_C4R	157
7.7.2.29	nppiMinIndxGetBufferHostSize_8u_AC4R	157
7.7.2.30	nppiMinIndxGetBufferHostSize_8u_C1R	158
7.7.2.31	nppiMinIndxGetBufferHostSize_8u_C3R	158
7.7.2.32	nppiMinIndxGetBufferHostSize_8u_C4R	158
7.8	Max	159
7.8.1	Detailed Description	161

7.8.2	Function Documentation	161
7.8.2.1	nppiMax_16s_AC4R	161
7.8.2.2	nppiMax_16s_C1R	162
7.8.2.3	nppiMax_16s_C3R	162
7.8.2.4	nppiMax_16s_C4R	162
7.8.2.5	nppiMax_16u_AC4R	163
7.8.2.6	nppiMax_16u_C1R	163
7.8.2.7	nppiMax_16u_C3R	163
7.8.2.8	nppiMax_16u_C4R	164
7.8.2.9	nppiMax_32f_AC4R	164
7.8.2.10	nppiMax_32f_C1R	164
7.8.2.11	nppiMax_32f_C3R	165
7.8.2.12	nppiMax_32f_C4R	165
7.8.2.13	nppiMax_8u_AC4R	166
7.8.2.14	nppiMax_8u_C1R	166
7.8.2.15	nppiMax_8u_C3R	166
7.8.2.16	nppiMax_8u_C4R	167
7.8.2.17	nppiMaxGetBufferHostSize_16s_AC4R	167
7.8.2.18	nppiMaxGetBufferHostSize_16s_C1R	167
7.8.2.19	nppiMaxGetBufferHostSize_16s_C3R	168
7.8.2.20	nppiMaxGetBufferHostSize_16s_C4R	168
7.8.2.21	nppiMaxGetBufferHostSize_16u_AC4R	168
7.8.2.22	nppiMaxGetBufferHostSize_16u_C1R	168
7.8.2.23	nppiMaxGetBufferHostSize_16u_C3R	169
7.8.2.24	nppiMaxGetBufferHostSize_16u_C4R	169
7.8.2.25	nppiMaxGetBufferHostSize_32f_AC4R	169
7.8.2.26	nppiMaxGetBufferHostSize_32f_C1R	170
7.8.2.27	nppiMaxGetBufferHostSize_32f_C3R	170
7.8.2.28	nppiMaxGetBufferHostSize_32f_C4R	170
7.8.2.29	nppiMaxGetBufferHostSize_8u_AC4R	170
7.8.2.30	nppiMaxGetBufferHostSize_8u_C1R	171
7.8.2.31	nppiMaxGetBufferHostSize_8u_C3R	171
7.8.2.32	nppiMaxGetBufferHostSize_8u_C4R	171
7.9	MaxIndx	172
7.9.1	Detailed Description	174
7.9.2	Function Documentation	174

7.9.2.1	nppiMaxIndx_16s_AC4R	174
7.9.2.2	nppiMaxIndx_16s_C1R	175
7.9.2.3	nppiMaxIndx_16s_C3R	175
7.9.2.4	nppiMaxIndx_16s_C4R	176
7.9.2.5	nppiMaxIndx_16u_AC4R	176
7.9.2.6	nppiMaxIndx_16u_C1R	176
7.9.2.7	nppiMaxIndx_16u_C3R	177
7.9.2.8	nppiMaxIndx_16u_C4R	177
7.9.2.9	nppiMaxIndx_32f_AC4R	178
7.9.2.10	nppiMaxIndx_32f_C1R	178
7.9.2.11	nppiMaxIndx_32f_C3R	178
7.9.2.12	nppiMaxIndx_32f_C4R	179
7.9.2.13	nppiMaxIndx_8u_AC4R	179
7.9.2.14	nppiMaxIndx_8u_C1R	180
7.9.2.15	nppiMaxIndx_8u_C3R	180
7.9.2.16	nppiMaxIndx_8u_C4R	180
7.9.2.17	nppiMaxIndxGetBufferHostSize_16s_AC4R	181
7.9.2.18	nppiMaxIndxGetBufferHostSize_16s_C1R	181
7.9.2.19	nppiMaxIndxGetBufferHostSize_16s_C3R	181
7.9.2.20	nppiMaxIndxGetBufferHostSize_16s_C4R	182
7.9.2.21	nppiMaxIndxGetBufferHostSize_16u_AC4R	182
7.9.2.22	nppiMaxIndxGetBufferHostSize_16u_C1R	182
7.9.2.23	nppiMaxIndxGetBufferHostSize_16u_C3R	183
7.9.2.24	nppiMaxIndxGetBufferHostSize_16u_C4R	183
7.9.2.25	nppiMaxIndxGetBufferHostSize_32f_AC4R	183
7.9.2.26	nppiMaxIndxGetBufferHostSize_32f_C1R	183
7.9.2.27	nppiMaxIndxGetBufferHostSize_32f_C3R	184
7.9.2.28	nppiMaxIndxGetBufferHostSize_32f_C4R	184
7.9.2.29	nppiMaxIndxGetBufferHostSize_8u_AC4R	184
7.9.2.30	nppiMaxIndxGetBufferHostSize_8u_C1R	185
7.9.2.31	nppiMaxIndxGetBufferHostSize_8u_C3R	185
7.9.2.32	nppiMaxIndxGetBufferHostSize_8u_C4R	185
7.10	MinMax	186
7.10.1	Detailed Description	188
7.10.2	Function Documentation	188
7.10.2.1	nppiMinMax_16s_AC4R	188

7.10.2.2	nppiMinMax_16s_C1R	189
7.10.2.3	nppiMinMax_16s_C3R	189
7.10.2.4	nppiMinMax_16s_C4R	189
7.10.2.5	nppiMinMax_16u_AC4R	190
7.10.2.6	nppiMinMax_16u_C1R	190
7.10.2.7	nppiMinMax_16u_C3R	191
7.10.2.8	nppiMinMax_16u_C4R	191
7.10.2.9	nppiMinMax_32f_AC4R	191
7.10.2.10	nppiMinMax_32f_C1R	192
7.10.2.11	nppiMinMax_32f_C3R	192
7.10.2.12	nppiMinMax_32f_C4R	193
7.10.2.13	nppiMinMax_8u_AC4R	193
7.10.2.14	nppiMinMax_8u_C1R	193
7.10.2.15	nppiMinMax_8u_C3R	194
7.10.2.16	nppiMinMax_8u_C4R	194
7.10.2.17	nppiMinMaxGetBufferHostSize_16s_AC4R	195
7.10.2.18	nppiMinMaxGetBufferHostSize_16s_C1R	195
7.10.2.19	nppiMinMaxGetBufferHostSize_16s_C3R	195
7.10.2.20	nppiMinMaxGetBufferHostSize_16s_C4R	195
7.10.2.21	nppiMinMaxGetBufferHostSize_16u_AC4R	196
7.10.2.22	nppiMinMaxGetBufferHostSize_16u_C1R	196
7.10.2.23	nppiMinMaxGetBufferHostSize_16u_C3R	196
7.10.2.24	nppiMinMaxGetBufferHostSize_16u_C4R	197
7.10.2.25	nppiMinMaxGetBufferHostSize_32f_AC4R	197
7.10.2.26	nppiMinMaxGetBufferHostSize_32f_C1R	197
7.10.2.27	nppiMinMaxGetBufferHostSize_32f_C3R	197
7.10.2.28	nppiMinMaxGetBufferHostSize_32f_C4R	198
7.10.2.29	nppiMinMaxGetBufferHostSize_8u_AC4R	198
7.10.2.30	nppiMinMaxGetBufferHostSize_8u_C1R	198
7.10.2.31	nppiMinMaxGetBufferHostSize_8u_C3R	199
7.10.2.32	nppiMinMaxGetBufferHostSize_8u_C4R	199
7.11	MinMaxIndx	200
7.11.1	Detailed Description	203
7.11.2	Function Documentation	203
7.11.2.1	nppiMinMaxIndx_16u_C1MR	203
7.11.2.2	nppiMinMaxIndx_16u_C1R	204

7.11.2.3	nppiMinMaxIndx_16u_C3CMR	204
7.11.2.4	nppiMinMaxIndx_16u_C3CR	205
7.11.2.5	nppiMinMaxIndx_32f_C1MR	206
7.11.2.6	nppiMinMaxIndx_32f_C1R	206
7.11.2.7	nppiMinMaxIndx_32f_C3CMR	207
7.11.2.8	nppiMinMaxIndx_32f_C3CR	207
7.11.2.9	nppiMinMaxIndx_8s_C1MR	208
7.11.2.10	nppiMinMaxIndx_8s_C1R	208
7.11.2.11	nppiMinMaxIndx_8s_C3CMR	209
7.11.2.12	nppiMinMaxIndx_8s_C3CR	209
7.11.2.13	nppiMinMaxIndx_8u_C1MR	210
7.11.2.14	nppiMinMaxIndx_8u_C1R	211
7.11.2.15	nppiMinMaxIndx_8u_C3CMR	211
7.11.2.16	nppiMinMaxIndx_8u_C3CR	212
7.11.2.17	nppiMinMaxIndxGetBufferHostSize_16u_C1MR	212
7.11.2.18	nppiMinMaxIndxGetBufferHostSize_16u_C1R	212
7.11.2.19	nppiMinMaxIndxGetBufferHostSize_16u_C3CMR	213
7.11.2.20	nppiMinMaxIndxGetBufferHostSize_16u_C3CR	213
7.11.2.21	nppiMinMaxIndxGetBufferHostSize_32f_C1MR	213
7.11.2.22	nppiMinMaxIndxGetBufferHostSize_32f_C1R	213
7.11.2.23	nppiMinMaxIndxGetBufferHostSize_32f_C3CMR	214
7.11.2.24	nppiMinMaxIndxGetBufferHostSize_32f_C3CR	214
7.11.2.25	nppiMinMaxIndxGetBufferHostSize_8s_C1MR	214
7.11.2.26	nppiMinMaxIndxGetBufferHostSize_8s_C1R	215
7.11.2.27	nppiMinMaxIndxGetBufferHostSize_8s_C3CMR	215
7.11.2.28	nppiMinMaxIndxGetBufferHostSize_8s_C3CR	215
7.11.2.29	nppiMinMaxIndxGetBufferHostSize_8u_C1MR	215
7.11.2.30	nppiMinMaxIndxGetBufferHostSize_8u_C1R	216
7.11.2.31	nppiMinMaxIndxGetBufferHostSize_8u_C3CMR	216
7.11.2.32	nppiMinMaxIndxGetBufferHostSize_8u_C3CR	216
7.12	Mean	217
7.12.1	Detailed Description	220
7.12.2	Function Documentation	221
7.12.2.1	nppiMean_16s_AC4R	221
7.12.2.2	nppiMean_16s_C1R	221
7.12.2.3	nppiMean_16s_C3R	221

7.12.2.4	nppiMean_16s_C4R	222
7.12.2.5	nppiMean_16u_AC4R	222
7.12.2.6	nppiMean_16u_C1MR	222
7.12.2.7	nppiMean_16u_C1R	223
7.12.2.8	nppiMean_16u_C3CMR	223
7.12.2.9	nppiMean_16u_C3R	224
7.12.2.10	nppiMean_16u_C4R	224
7.12.2.11	nppiMean_32f_AC4R	224
7.12.2.12	nppiMean_32f_C1MR	225
7.12.2.13	nppiMean_32f_C1R	225
7.12.2.14	nppiMean_32f_C3CMR	226
7.12.2.15	nppiMean_32f_C3R	226
7.12.2.16	nppiMean_32f_C4R	226
7.12.2.17	nppiMean_8s_C1MR	227
7.12.2.18	nppiMean_8s_C3CMR	227
7.12.2.19	nppiMean_8u_AC4R	228
7.12.2.20	nppiMean_8u_C1MR	228
7.12.2.21	nppiMean_8u_C1R	229
7.12.2.22	nppiMean_8u_C3CMR	229
7.12.2.23	nppiMean_8u_C3R	229
7.12.2.24	nppiMean_8u_C4R	230
7.12.2.25	nppiMeanGetBufferHostSize_16s_AC4R	230
7.12.2.26	nppiMeanGetBufferHostSize_16s_C1R	230
7.12.2.27	nppiMeanGetBufferHostSize_16s_C3R	231
7.12.2.28	nppiMeanGetBufferHostSize_16s_C4R	231
7.12.2.29	nppiMeanGetBufferHostSize_16u_AC4R	231
7.12.2.30	nppiMeanGetBufferHostSize_16u_C1MR	232
7.12.2.31	nppiMeanGetBufferHostSize_16u_C1R	232
7.12.2.32	nppiMeanGetBufferHostSize_16u_C3CMR	232
7.12.2.33	nppiMeanGetBufferHostSize_16u_C3R	232
7.12.2.34	nppiMeanGetBufferHostSize_16u_C4R	233
7.12.2.35	nppiMeanGetBufferHostSize_32f_AC4R	233
7.12.2.36	nppiMeanGetBufferHostSize_32f_C1MR	233
7.12.2.37	nppiMeanGetBufferHostSize_32f_C1R	234
7.12.2.38	nppiMeanGetBufferHostSize_32f_C3CMR	234
7.12.2.39	nppiMeanGetBufferHostSize_32f_C3R	234

7.12.2.40	<code>nppiMeanGetBufferHostSize_32f_C4R</code>	234
7.12.2.41	<code>nppiMeanGetBufferHostSize_8s_C1MR</code>	235
7.12.2.42	<code>nppiMeanGetBufferHostSize_8s_C3CMR</code>	235
7.12.2.43	<code>nppiMeanGetBufferHostSize_8u_AC4R</code>	235
7.12.2.44	<code>nppiMeanGetBufferHostSize_8u_C1MR</code>	236
7.12.2.45	<code>nppiMeanGetBufferHostSize_8u_C1R</code>	236
7.12.2.46	<code>nppiMeanGetBufferHostSize_8u_C3CMR</code>	236
7.12.2.47	<code>nppiMeanGetBufferHostSize_8u_C3R</code>	236
7.12.2.48	<code>nppiMeanGetBufferHostSize_8u_C4R</code>	237
7.13	<code>Mean_StdDev</code>	238
7.13.1	Detailed Description	241
7.13.2	Function Documentation	241
7.13.2.1	<code>nppiMean_StdDev_16u_C1MR</code>	241
7.13.2.2	<code>nppiMean_StdDev_16u_C1R</code>	242
7.13.2.3	<code>nppiMean_StdDev_16u_C3CMR</code>	242
7.13.2.4	<code>nppiMean_StdDev_16u_C3CR</code>	243
7.13.2.5	<code>nppiMean_StdDev_32f_C1MR</code>	243
7.13.2.6	<code>nppiMean_StdDev_32f_C1R</code>	244
7.13.2.7	<code>nppiMean_StdDev_32f_C3CMR</code>	244
7.13.2.8	<code>nppiMean_StdDev_32f_C3CR</code>	245
7.13.2.9	<code>nppiMean_StdDev_8s_C1MR</code>	245
7.13.2.10	<code>nppiMean_StdDev_8s_C1R</code>	246
7.13.2.11	<code>nppiMean_StdDev_8s_C3CMR</code>	246
7.13.2.12	<code>nppiMean_StdDev_8s_C3CR</code>	247
7.13.2.13	<code>nppiMean_StdDev_8u_C1MR</code>	247
7.13.2.14	<code>nppiMean_StdDev_8u_C1R</code>	248
7.13.2.15	<code>nppiMean_StdDev_8u_C3CMR</code>	248
7.13.2.16	<code>nppiMean_StdDev_8u_C3CR</code>	249
7.13.2.17	<code>nppiMeanStdDevGetBufferHostSize_16u_C1MR</code>	249
7.13.2.18	<code>nppiMeanStdDevGetBufferHostSize_16u_C1R</code>	249
7.13.2.19	<code>nppiMeanStdDevGetBufferHostSize_16u_C3CMR</code>	250
7.13.2.20	<code>nppiMeanStdDevGetBufferHostSize_16u_C3CR</code>	250
7.13.2.21	<code>nppiMeanStdDevGetBufferHostSize_32f_C1MR</code>	250
7.13.2.22	<code>nppiMeanStdDevGetBufferHostSize_32f_C1R</code>	250
7.13.2.23	<code>nppiMeanStdDevGetBufferHostSize_32f_C3CMR</code>	251
7.13.2.24	<code>nppiMeanStdDevGetBufferHostSize_32f_C3CR</code>	251

7.13.2.25	<code>nppiMeanStdDevGetBufferHostSize_8s_C1MR</code>	251
7.13.2.26	<code>nppiMeanStdDevGetBufferHostSize_8s_C1R</code>	252
7.13.2.27	<code>nppiMeanStdDevGetBufferHostSize_8s_C3CMR</code>	252
7.13.2.28	<code>nppiMeanStdDevGetBufferHostSize_8s_C3CR</code>	252
7.13.2.29	<code>nppiMeanStdDevGetBufferHostSize_8u_C1MR</code>	252
7.13.2.30	<code>nppiMeanStdDevGetBufferHostSize_8u_C1R</code>	253
7.13.2.31	<code>nppiMeanStdDevGetBufferHostSize_8u_C3CMR</code>	253
7.13.2.32	<code>nppiMeanStdDevGetBufferHostSize_8u_C3CR</code>	253
7.14	Image Norms	254
7.14.1	Detailed Description	254
7.15	<code>Norm_Inf</code>	256
7.15.1	Detailed Description	260
7.15.2	Function Documentation	260
7.15.2.1	<code>nppiNorm_Inf_16s_AC4R</code>	260
7.15.2.2	<code>nppiNorm_Inf_16s_C1R</code>	260
7.15.2.3	<code>nppiNorm_Inf_16s_C3R</code>	260
7.15.2.4	<code>nppiNorm_Inf_16s_C4R</code>	261
7.15.2.5	<code>nppiNorm_Inf_16u_AC4R</code>	261
7.15.2.6	<code>nppiNorm_Inf_16u_C1MR</code>	262
7.15.2.7	<code>nppiNorm_Inf_16u_C1R</code>	262
7.15.2.8	<code>nppiNorm_Inf_16u_C3CMR</code>	262
7.15.2.9	<code>nppiNorm_Inf_16u_C3R</code>	263
7.15.2.10	<code>nppiNorm_Inf_16u_C4R</code>	263
7.15.2.11	<code>nppiNorm_Inf_32f_AC4R</code>	264
7.15.2.12	<code>nppiNorm_Inf_32f_C1MR</code>	264
7.15.2.13	<code>nppiNorm_Inf_32f_C1R</code>	264
7.15.2.14	<code>nppiNorm_Inf_32f_C3CMR</code>	265
7.15.2.15	<code>nppiNorm_Inf_32f_C3R</code>	265
7.15.2.16	<code>nppiNorm_Inf_32f_C4R</code>	266
7.15.2.17	<code>nppiNorm_Inf_32s_C1R</code>	266
7.15.2.18	<code>nppiNorm_Inf_8s_C1MR</code>	266
7.15.2.19	<code>nppiNorm_Inf_8s_C3CMR</code>	267
7.15.2.20	<code>nppiNorm_Inf_8u_AC4R</code>	267
7.15.2.21	<code>nppiNorm_Inf_8u_C1MR</code>	268
7.15.2.22	<code>nppiNorm_Inf_8u_C1R</code>	268
7.15.2.23	<code>nppiNorm_Inf_8u_C3CMR</code>	268

7.15.2.24	<code>nppiNorm_Inf_8u_C3R</code>	269
7.15.2.25	<code>nppiNorm_Inf_8u_C4R</code>	269
7.15.2.26	<code>nppiNormInfGetBufferHostSize_16s_AC4R</code>	270
7.15.2.27	<code>nppiNormInfGetBufferHostSize_16s_C1R</code>	270
7.15.2.28	<code>nppiNormInfGetBufferHostSize_16s_C3R</code>	270
7.15.2.29	<code>nppiNormInfGetBufferHostSize_16s_C4R</code>	270
7.15.2.30	<code>nppiNormInfGetBufferHostSize_16u_AC4R</code>	271
7.15.2.31	<code>nppiNormInfGetBufferHostSize_16u_C1MR</code>	271
7.15.2.32	<code>nppiNormInfGetBufferHostSize_16u_C1R</code>	271
7.15.2.33	<code>nppiNormInfGetBufferHostSize_16u_C3CMR</code>	272
7.15.2.34	<code>nppiNormInfGetBufferHostSize_16u_C3R</code>	272
7.15.2.35	<code>nppiNormInfGetBufferHostSize_16u_C4R</code>	272
7.15.2.36	<code>nppiNormInfGetBufferHostSize_32f_AC4R</code>	272
7.15.2.37	<code>nppiNormInfGetBufferHostSize_32f_C1MR</code>	273
7.15.2.38	<code>nppiNormInfGetBufferHostSize_32f_C1R</code>	273
7.15.2.39	<code>nppiNormInfGetBufferHostSize_32f_C3CMR</code>	273
7.15.2.40	<code>nppiNormInfGetBufferHostSize_32f_C3R</code>	274
7.15.2.41	<code>nppiNormInfGetBufferHostSize_32f_C4R</code>	274
7.15.2.42	<code>nppiNormInfGetBufferHostSize_32s_C1R</code>	274
7.15.2.43	<code>nppiNormInfGetBufferHostSize_8s_C1MR</code>	274
7.15.2.44	<code>nppiNormInfGetBufferHostSize_8s_C3CMR</code>	275
7.15.2.45	<code>nppiNormInfGetBufferHostSize_8u_AC4R</code>	275
7.15.2.46	<code>nppiNormInfGetBufferHostSize_8u_C1MR</code>	275
7.15.2.47	<code>nppiNormInfGetBufferHostSize_8u_C1R</code>	276
7.15.2.48	<code>nppiNormInfGetBufferHostSize_8u_C3CMR</code>	276
7.15.2.49	<code>nppiNormInfGetBufferHostSize_8u_C3R</code>	276
7.15.2.50	<code>nppiNormInfGetBufferHostSize_8u_C4R</code>	276
7.16	<code>Norm_L1</code>	278
7.16.1	Detailed Description	281
7.16.2	Function Documentation	282
7.16.2.1	<code>nppiNorm_L1_16s_AC4R</code>	282
7.16.2.2	<code>nppiNorm_L1_16s_C1R</code>	282
7.16.2.3	<code>nppiNorm_L1_16s_C3R</code>	282
7.16.2.4	<code>nppiNorm_L1_16s_C4R</code>	283
7.16.2.5	<code>nppiNorm_L1_16u_AC4R</code>	283
7.16.2.6	<code>nppiNorm_L1_16u_C1MR</code>	283

7.16.2.7	nppiNorm_L1_16u_C1R	284
7.16.2.8	nppiNorm_L1_16u_C3CMR	284
7.16.2.9	nppiNorm_L1_16u_C3R	285
7.16.2.10	nppiNorm_L1_16u_C4R	285
7.16.2.11	nppiNorm_L1_32f_AC4R	285
7.16.2.12	nppiNorm_L1_32f_C1MR	286
7.16.2.13	nppiNorm_L1_32f_C1R	286
7.16.2.14	nppiNorm_L1_32f_C3CMR	287
7.16.2.15	nppiNorm_L1_32f_C3R	287
7.16.2.16	nppiNorm_L1_32f_C4R	287
7.16.2.17	nppiNorm_L1_8s_C1MR	288
7.16.2.18	nppiNorm_L1_8s_C3CMR	288
7.16.2.19	nppiNorm_L1_8u_AC4R	289
7.16.2.20	nppiNorm_L1_8u_C1MR	289
7.16.2.21	nppiNorm_L1_8u_C1R	289
7.16.2.22	nppiNorm_L1_8u_C3CMR	290
7.16.2.23	nppiNorm_L1_8u_C3R	290
7.16.2.24	nppiNorm_L1_8u_C4R	291
7.16.2.25	nppiNormL1GetBufferHostSize_16s_AC4R	291
7.16.2.26	nppiNormL1GetBufferHostSize_16s_C1R	291
7.16.2.27	nppiNormL1GetBufferHostSize_16s_C3R	292
7.16.2.28	nppiNormL1GetBufferHostSize_16s_C4R	292
7.16.2.29	nppiNormL1GetBufferHostSize_16u_AC4R	292
7.16.2.30	nppiNormL1GetBufferHostSize_16u_C1MR	292
7.16.2.31	nppiNormL1GetBufferHostSize_16u_C1R	293
7.16.2.32	nppiNormL1GetBufferHostSize_16u_C3CMR	293
7.16.2.33	nppiNormL1GetBufferHostSize_16u_C3R	293
7.16.2.34	nppiNormL1GetBufferHostSize_16u_C4R	294
7.16.2.35	nppiNormL1GetBufferHostSize_32f_AC4R	294
7.16.2.36	nppiNormL1GetBufferHostSize_32f_C1MR	294
7.16.2.37	nppiNormL1GetBufferHostSize_32f_C1R	294
7.16.2.38	nppiNormL1GetBufferHostSize_32f_C3CMR	295
7.16.2.39	nppiNormL1GetBufferHostSize_32f_C3R	295
7.16.2.40	nppiNormL1GetBufferHostSize_32f_C4R	295
7.16.2.41	nppiNormL1GetBufferHostSize_8s_C1MR	296
7.16.2.42	nppiNormL1GetBufferHostSize_8s_C3CMR	296

7.16.2.43	nppiNormL1GetBufferHostSize_8u_AC4R	296
7.16.2.44	nppiNormL1GetBufferHostSize_8u_C1MR	296
7.16.2.45	nppiNormL1GetBufferHostSize_8u_C1R	297
7.16.2.46	nppiNormL1GetBufferHostSize_8u_C3CMR	297
7.16.2.47	nppiNormL1GetBufferHostSize_8u_C3R	297
7.16.2.48	nppiNormL1GetBufferHostSize_8u_C4R	298
7.17	Norm_L2	299
7.17.1	Detailed Description	302
7.17.2	Function Documentation	303
7.17.2.1	nppiNorm_L2_16s_AC4R	303
7.17.2.2	nppiNorm_L2_16s_C1R	303
7.17.2.3	nppiNorm_L2_16s_C3R	303
7.17.2.4	nppiNorm_L2_16s_C4R	304
7.17.2.5	nppiNorm_L2_16u_AC4R	304
7.17.2.6	nppiNorm_L2_16u_C1MR	304
7.17.2.7	nppiNorm_L2_16u_C1R	305
7.17.2.8	nppiNorm_L2_16u_C3CMR	305
7.17.2.9	nppiNorm_L2_16u_C3R	306
7.17.2.10	nppiNorm_L2_16u_C4R	306
7.17.2.11	nppiNorm_L2_32f_AC4R	306
7.17.2.12	nppiNorm_L2_32f_C1MR	307
7.17.2.13	nppiNorm_L2_32f_C1R	307
7.17.2.14	nppiNorm_L2_32f_C3CMR	308
7.17.2.15	nppiNorm_L2_32f_C3R	308
7.17.2.16	nppiNorm_L2_32f_C4R	308
7.17.2.17	nppiNorm_L2_8s_C1MR	309
7.17.2.18	nppiNorm_L2_8s_C3CMR	309
7.17.2.19	nppiNorm_L2_8u_AC4R	310
7.17.2.20	nppiNorm_L2_8u_C1MR	310
7.17.2.21	nppiNorm_L2_8u_C1R	310
7.17.2.22	nppiNorm_L2_8u_C3CMR	311
7.17.2.23	nppiNorm_L2_8u_C3R	311
7.17.2.24	nppiNorm_L2_8u_C4R	312
7.17.2.25	nppiNormL2GetBufferHostSize_16s_AC4R	312
7.17.2.26	nppiNormL2GetBufferHostSize_16s_C1R	312
7.17.2.27	nppiNormL2GetBufferHostSize_16s_C3R	313

7.17.2.28	nppiNormL2GetBufferHostSize_16s_C4R	313
7.17.2.29	nppiNormL2GetBufferHostSize_16u_AC4R	313
7.17.2.30	nppiNormL2GetBufferHostSize_16u_C1MR	313
7.17.2.31	nppiNormL2GetBufferHostSize_16u_C1R	314
7.17.2.32	nppiNormL2GetBufferHostSize_16u_C3CMR	314
7.17.2.33	nppiNormL2GetBufferHostSize_16u_C3R	314
7.17.2.34	nppiNormL2GetBufferHostSize_16u_C4R	315
7.17.2.35	nppiNormL2GetBufferHostSize_32f_AC4R	315
7.17.2.36	nppiNormL2GetBufferHostSize_32f_C1MR	315
7.17.2.37	nppiNormL2GetBufferHostSize_32f_C1R	315
7.17.2.38	nppiNormL2GetBufferHostSize_32f_C3CMR	316
7.17.2.39	nppiNormL2GetBufferHostSize_32f_C3R	316
7.17.2.40	nppiNormL2GetBufferHostSize_32f_C4R	316
7.17.2.41	nppiNormL2GetBufferHostSize_8s_C1MR	317
7.17.2.42	nppiNormL2GetBufferHostSize_8s_C3CMR	317
7.17.2.43	nppiNormL2GetBufferHostSize_8u_AC4R	317
7.17.2.44	nppiNormL2GetBufferHostSize_8u_C1MR	317
7.17.2.45	nppiNormL2GetBufferHostSize_8u_C1R	318
7.17.2.46	nppiNormL2GetBufferHostSize_8u_C3CMR	318
7.17.2.47	nppiNormL2GetBufferHostSize_8u_C3R	318
7.17.2.48	nppiNormL2GetBufferHostSize_8u_C4R	319
7.18	NormDiff_Inf	320
7.18.1	Detailed Description	324
7.18.2	Function Documentation	324
7.18.2.1	nppiNormDiff_Inf_16s_AC4R	324
7.18.2.2	nppiNormDiff_Inf_16s_C1R	325
7.18.2.3	nppiNormDiff_Inf_16s_C3R	325
7.18.2.4	nppiNormDiff_Inf_16s_C4R	325
7.18.2.5	nppiNormDiff_Inf_16u_AC4R	326
7.18.2.6	nppiNormDiff_Inf_16u_C1MR	326
7.18.2.7	nppiNormDiff_Inf_16u_C1R	327
7.18.2.8	nppiNormDiff_Inf_16u_C3CMR	327
7.18.2.9	nppiNormDiff_Inf_16u_C3R	328
7.18.2.10	nppiNormDiff_Inf_16u_C4R	328
7.18.2.11	nppiNormDiff_Inf_32f_AC4R	329
7.18.2.12	nppiNormDiff_Inf_32f_C1MR	329

7.18.2.13	nppiNormDiff_Inf_32f_C1R	330
7.18.2.14	nppiNormDiff_Inf_32f_C3CMR	330
7.18.2.15	nppiNormDiff_Inf_32f_C3R	331
7.18.2.16	nppiNormDiff_Inf_32f_C4R	331
7.18.2.17	nppiNormDiff_Inf_8s_C1MR	331
7.18.2.18	nppiNormDiff_Inf_8s_C3CMR	332
7.18.2.19	nppiNormDiff_Inf_8u_AC4R	332
7.18.2.20	nppiNormDiff_Inf_8u_C1MR	333
7.18.2.21	nppiNormDiff_Inf_8u_C1R	333
7.18.2.22	nppiNormDiff_Inf_8u_C3CMR	334
7.18.2.23	nppiNormDiff_Inf_8u_C3R	334
7.18.2.24	nppiNormDiff_Inf_8u_C4R	335
7.18.2.25	nppiNormDiffInfGetBufferHostSize_16s_AC4R	335
7.18.2.26	nppiNormDiffInfGetBufferHostSize_16s_C1R	336
7.18.2.27	nppiNormDiffInfGetBufferHostSize_16s_C3R	336
7.18.2.28	nppiNormDiffInfGetBufferHostSize_16s_C4R	336
7.18.2.29	nppiNormDiffInfGetBufferHostSize_16u_AC4R	336
7.18.2.30	nppiNormDiffInfGetBufferHostSize_16u_C1MR	337
7.18.2.31	nppiNormDiffInfGetBufferHostSize_16u_C1R	337
7.18.2.32	nppiNormDiffInfGetBufferHostSize_16u_C3CMR	337
7.18.2.33	nppiNormDiffInfGetBufferHostSize_16u_C3R	338
7.18.2.34	nppiNormDiffInfGetBufferHostSize_16u_C4R	338
7.18.2.35	nppiNormDiffInfGetBufferHostSize_32f_AC4R	338
7.18.2.36	nppiNormDiffInfGetBufferHostSize_32f_C1MR	338
7.18.2.37	nppiNormDiffInfGetBufferHostSize_32f_C1R	339
7.18.2.38	nppiNormDiffInfGetBufferHostSize_32f_C3CMR	339
7.18.2.39	nppiNormDiffInfGetBufferHostSize_32f_C3R	339
7.18.2.40	nppiNormDiffInfGetBufferHostSize_32f_C4R	340
7.18.2.41	nppiNormDiffInfGetBufferHostSize_8s_C1MR	340
7.18.2.42	nppiNormDiffInfGetBufferHostSize_8s_C3CMR	340
7.18.2.43	nppiNormDiffInfGetBufferHostSize_8u_AC4R	340
7.18.2.44	nppiNormDiffInfGetBufferHostSize_8u_C1MR	341
7.18.2.45	nppiNormDiffInfGetBufferHostSize_8u_C1R	341
7.18.2.46	nppiNormDiffInfGetBufferHostSize_8u_C3CMR	341
7.18.2.47	nppiNormDiffInfGetBufferHostSize_8u_C3R	342
7.18.2.48	nppiNormDiffInfGetBufferHostSize_8u_C4R	342

7.19 NormDiff_L1	343
7.19.1 Detailed Description	347
7.19.2 Function Documentation	347
7.19.2.1 nppiNormDiff_L1_16s_AC4R	347
7.19.2.2 nppiNormDiff_L1_16s_C1R	347
7.19.2.3 nppiNormDiff_L1_16s_C3R	348
7.19.2.4 nppiNormDiff_L1_16s_C4R	348
7.19.2.5 nppiNormDiff_L1_16u_AC4R	349
7.19.2.6 nppiNormDiff_L1_16u_C1MR	349
7.19.2.7 nppiNormDiff_L1_16u_C1R	350
7.19.2.8 nppiNormDiff_L1_16u_C3CMR	350
7.19.2.9 nppiNormDiff_L1_16u_C3R	351
7.19.2.10 nppiNormDiff_L1_16u_C4R	351
7.19.2.11 nppiNormDiff_L1_32f_AC4R	351
7.19.2.12 nppiNormDiff_L1_32f_C1MR	352
7.19.2.13 nppiNormDiff_L1_32f_C1R	352
7.19.2.14 nppiNormDiff_L1_32f_C3CMR	353
7.19.2.15 nppiNormDiff_L1_32f_C3R	353
7.19.2.16 nppiNormDiff_L1_32f_C4R	354
7.19.2.17 nppiNormDiff_L1_8s_C1MR	354
7.19.2.18 nppiNormDiff_L1_8s_C3CMR	355
7.19.2.19 nppiNormDiff_L1_8u_AC4R	355
7.19.2.20 nppiNormDiff_L1_8u_C1MR	356
7.19.2.21 nppiNormDiff_L1_8u_C1R	356
7.19.2.22 nppiNormDiff_L1_8u_C3CMR	357
7.19.2.23 nppiNormDiff_L1_8u_C3R	357
7.19.2.24 nppiNormDiff_L1_8u_C4R	358
7.19.2.25 nppiNormDiffL1GetBufferHostSize_16s_AC4R	358
7.19.2.26 nppiNormDiffL1GetBufferHostSize_16s_C1R	358
7.19.2.27 nppiNormDiffL1GetBufferHostSize_16s_C3R	359
7.19.2.28 nppiNormDiffL1GetBufferHostSize_16s_C4R	359
7.19.2.29 nppiNormDiffL1GetBufferHostSize_16u_AC4R	359
7.19.2.30 nppiNormDiffL1GetBufferHostSize_16u_C1MR	359
7.19.2.31 nppiNormDiffL1GetBufferHostSize_16u_C1R	360
7.19.2.32 nppiNormDiffL1GetBufferHostSize_16u_C3CMR	360
7.19.2.33 nppiNormDiffL1GetBufferHostSize_16u_C3R	360

7.19.2.34	nppiNormDiffL1GetBufferHostSize_16u_C4R	361
7.19.2.35	nppiNormDiffL1GetBufferHostSize_32f_AC4R	361
7.19.2.36	nppiNormDiffL1GetBufferHostSize_32f_C1MR	361
7.19.2.37	nppiNormDiffL1GetBufferHostSize_32f_C1R	361
7.19.2.38	nppiNormDiffL1GetBufferHostSize_32f_C3CMR	362
7.19.2.39	nppiNormDiffL1GetBufferHostSize_32f_C3R	362
7.19.2.40	nppiNormDiffL1GetBufferHostSize_32f_C4R	362
7.19.2.41	nppiNormDiffL1GetBufferHostSize_8s_C1MR	363
7.19.2.42	nppiNormDiffL1GetBufferHostSize_8s_C3CMR	363
7.19.2.43	nppiNormDiffL1GetBufferHostSize_8u_AC4R	363
7.19.2.44	nppiNormDiffL1GetBufferHostSize_8u_C1MR	363
7.19.2.45	nppiNormDiffL1GetBufferHostSize_8u_C1R	364
7.19.2.46	nppiNormDiffL1GetBufferHostSize_8u_C3CMR	364
7.19.2.47	nppiNormDiffL1GetBufferHostSize_8u_C3R	364
7.19.2.48	nppiNormDiffL1GetBufferHostSize_8u_C4R	365
7.20	NormDiff_L2	366
7.20.1	Detailed Description	370
7.20.2	Function Documentation	370
7.20.2.1	nppiNormDiff_L2_16s_AC4R	370
7.20.2.2	nppiNormDiff_L2_16s_C1R	370
7.20.2.3	nppiNormDiff_L2_16s_C3R	371
7.20.2.4	nppiNormDiff_L2_16s_C4R	371
7.20.2.5	nppiNormDiff_L2_16u_AC4R	372
7.20.2.6	nppiNormDiff_L2_16u_C1MR	372
7.20.2.7	nppiNormDiff_L2_16u_C1R	373
7.20.2.8	nppiNormDiff_L2_16u_C3CMR	373
7.20.2.9	nppiNormDiff_L2_16u_C3R	374
7.20.2.10	nppiNormDiff_L2_16u_C4R	374
7.20.2.11	nppiNormDiff_L2_32f_AC4R	374
7.20.2.12	nppiNormDiff_L2_32f_C1MR	375
7.20.2.13	nppiNormDiff_L2_32f_C1R	375
7.20.2.14	nppiNormDiff_L2_32f_C3CMR	376
7.20.2.15	nppiNormDiff_L2_32f_C3R	376
7.20.2.16	nppiNormDiff_L2_32f_C4R	377
7.20.2.17	nppiNormDiff_L2_8s_C1MR	377
7.20.2.18	nppiNormDiff_L2_8s_C3CMR	378

7.20.2.19	nppiNormDiff_L2_8u_AC4R	378
7.20.2.20	nppiNormDiff_L2_8u_C1MR	379
7.20.2.21	nppiNormDiff_L2_8u_C1R	379
7.20.2.22	nppiNormDiff_L2_8u_C3CMR	380
7.20.2.23	nppiNormDiff_L2_8u_C3R	380
7.20.2.24	nppiNormDiff_L2_8u_C4R	381
7.20.2.25	nppiNormDiffL2GetBufferHostSize_16s_AC4R	381
7.20.2.26	nppiNormDiffL2GetBufferHostSize_16s_C1R	381
7.20.2.27	nppiNormDiffL2GetBufferHostSize_16s_C3R	382
7.20.2.28	nppiNormDiffL2GetBufferHostSize_16s_C4R	382
7.20.2.29	nppiNormDiffL2GetBufferHostSize_16u_AC4R	382
7.20.2.30	nppiNormDiffL2GetBufferHostSize_16u_C1MR	382
7.20.2.31	nppiNormDiffL2GetBufferHostSize_16u_C1R	383
7.20.2.32	nppiNormDiffL2GetBufferHostSize_16u_C3CMR	383
7.20.2.33	nppiNormDiffL2GetBufferHostSize_16u_C3R	383
7.20.2.34	nppiNormDiffL2GetBufferHostSize_16u_C4R	384
7.20.2.35	nppiNormDiffL2GetBufferHostSize_32f_AC4R	384
7.20.2.36	nppiNormDiffL2GetBufferHostSize_32f_C1MR	384
7.20.2.37	nppiNormDiffL2GetBufferHostSize_32f_C1R	384
7.20.2.38	nppiNormDiffL2GetBufferHostSize_32f_C3CMR	385
7.20.2.39	nppiNormDiffL2GetBufferHostSize_32f_C3R	385
7.20.2.40	nppiNormDiffL2GetBufferHostSize_32f_C4R	385
7.20.2.41	nppiNormDiffL2GetBufferHostSize_8s_C1MR	386
7.20.2.42	nppiNormDiffL2GetBufferHostSize_8s_C3CMR	386
7.20.2.43	nppiNormDiffL2GetBufferHostSize_8u_AC4R	386
7.20.2.44	nppiNormDiffL2GetBufferHostSize_8u_C1MR	386
7.20.2.45	nppiNormDiffL2GetBufferHostSize_8u_C1R	387
7.20.2.46	nppiNormDiffL2GetBufferHostSize_8u_C3CMR	387
7.20.2.47	nppiNormDiffL2GetBufferHostSize_8u_C3R	387
7.20.2.48	nppiNormDiffL2GetBufferHostSize_8u_C4R	388
7.21	NormRel_Inf	389
7.21.1	Detailed Description	393
7.21.2	Function Documentation	393
7.21.2.1	nppiNormRel_Inf_16s_AC4R	393
7.21.2.2	nppiNormRel_Inf_16s_C1R	393
7.21.2.3	nppiNormRel_Inf_16s_C3R	394

7.21.2.4	nppiNormRel_Inf_16s_C4R	394
7.21.2.5	nppiNormRel_Inf_16u_AC4R	395
7.21.2.6	nppiNormRel_Inf_16u_C1MR	395
7.21.2.7	nppiNormRel_Inf_16u_C1R	396
7.21.2.8	nppiNormRel_Inf_16u_C3CMR	396
7.21.2.9	nppiNormRel_Inf_16u_C3R	397
7.21.2.10	nppiNormRel_Inf_16u_C4R	397
7.21.2.11	nppiNormRel_Inf_32f_AC4R	398
7.21.2.12	nppiNormRel_Inf_32f_C1MR	398
7.21.2.13	nppiNormRel_Inf_32f_C1R	399
7.21.2.14	nppiNormRel_Inf_32f_C3CMR	399
7.21.2.15	nppiNormRel_Inf_32f_C3R	400
7.21.2.16	nppiNormRel_Inf_32f_C4R	400
7.21.2.17	nppiNormRel_Inf_8s_C1MR	401
7.21.2.18	nppiNormRel_Inf_8s_C3CMR	401
7.21.2.19	nppiNormRel_Inf_8u_AC4R	402
7.21.2.20	nppiNormRel_Inf_8u_C1MR	402
7.21.2.21	nppiNormRel_Inf_8u_C1R	403
7.21.2.22	nppiNormRel_Inf_8u_C3CMR	403
7.21.2.23	nppiNormRel_Inf_8u_C3R	404
7.21.2.24	nppiNormRel_Inf_8u_C4R	404
7.21.2.25	nppiNormRelInfGetBufferHostSize_16s_AC4R	404
7.21.2.26	nppiNormRelInfGetBufferHostSize_16s_C1R	405
7.21.2.27	nppiNormRelInfGetBufferHostSize_16s_C3R	405
7.21.2.28	nppiNormRelInfGetBufferHostSize_16s_C4R	405
7.21.2.29	nppiNormRelInfGetBufferHostSize_16u_AC4R	406
7.21.2.30	nppiNormRelInfGetBufferHostSize_16u_C1MR	406
7.21.2.31	nppiNormRelInfGetBufferHostSize_16u_C1R	406
7.21.2.32	nppiNormRelInfGetBufferHostSize_16u_C3CMR	406
7.21.2.33	nppiNormRelInfGetBufferHostSize_16u_C3R	407
7.21.2.34	nppiNormRelInfGetBufferHostSize_16u_C4R	407
7.21.2.35	nppiNormRelInfGetBufferHostSize_32f_AC4R	407
7.21.2.36	nppiNormRelInfGetBufferHostSize_32f_C1MR	408
7.21.2.37	nppiNormRelInfGetBufferHostSize_32f_C1R	408
7.21.2.38	nppiNormRelInfGetBufferHostSize_32f_C3CMR	408
7.21.2.39	nppiNormRelInfGetBufferHostSize_32f_C3R	408

7.21.2.40	nppiNormRelInfGetBufferHostSize_32f_C4R	409
7.21.2.41	nppiNormRelInfGetBufferHostSize_32s_C1R	409
7.21.2.42	nppiNormRelInfGetBufferHostSize_8s_C1MR	409
7.21.2.43	nppiNormRelInfGetBufferHostSize_8s_C3CMR	410
7.21.2.44	nppiNormRelInfGetBufferHostSize_8u_AC4R	410
7.21.2.45	nppiNormRelInfGetBufferHostSize_8u_C1MR	410
7.21.2.46	nppiNormRelInfGetBufferHostSize_8u_C1R	410
7.21.2.47	nppiNormRelInfGetBufferHostSize_8u_C3CMR	411
7.21.2.48	nppiNormRelInfGetBufferHostSize_8u_C3R	411
7.21.2.49	nppiNormRelInfGetBufferHostSize_8u_C4R	411
7.22	NormRel_L1	412
7.22.1	Detailed Description	416
7.22.2	Function Documentation	416
7.22.2.1	nppiNormRel_L1_16s_AC4R	416
7.22.2.2	nppiNormRel_L1_16s_C1R	416
7.22.2.3	nppiNormRel_L1_16s_C3R	417
7.22.2.4	nppiNormRel_L1_16s_C4R	417
7.22.2.5	nppiNormRel_L1_16u_AC4R	418
7.22.2.6	nppiNormRel_L1_16u_C1MR	418
7.22.2.7	nppiNormRel_L1_16u_C1R	419
7.22.2.8	nppiNormRel_L1_16u_C3CMR	419
7.22.2.9	nppiNormRel_L1_16u_C3R	420
7.22.2.10	nppiNormRel_L1_16u_C4R	420
7.22.2.11	nppiNormRel_L1_32f_AC4R	420
7.22.2.12	nppiNormRel_L1_32f_C1MR	421
7.22.2.13	nppiNormRel_L1_32f_C1R	421
7.22.2.14	nppiNormRel_L1_32f_C3CMR	422
7.22.2.15	nppiNormRel_L1_32f_C3R	422
7.22.2.16	nppiNormRel_L1_32f_C4R	423
7.22.2.17	nppiNormRel_L1_8s_C1MR	423
7.22.2.18	nppiNormRel_L1_8s_C3CMR	424
7.22.2.19	nppiNormRel_L1_8u_AC4R	424
7.22.2.20	nppiNormRel_L1_8u_C1MR	425
7.22.2.21	nppiNormRel_L1_8u_C1R	425
7.22.2.22	nppiNormRel_L1_8u_C3CMR	426
7.22.2.23	nppiNormRel_L1_8u_C3R	426

7.22.2.24	nppiNormRel_L1_8u_C4R	427
7.22.2.25	nppiNormRelL1GetBufferHostSize_16s_AC4R	427
7.22.2.26	nppiNormRelL1GetBufferHostSize_16s_C1R	428
7.22.2.27	nppiNormRelL1GetBufferHostSize_16s_C3R	428
7.22.2.28	nppiNormRelL1GetBufferHostSize_16s_C4R	428
7.22.2.29	nppiNormRelL1GetBufferHostSize_16u_AC4R	428
7.22.2.30	nppiNormRelL1GetBufferHostSize_16u_C1MR	429
7.22.2.31	nppiNormRelL1GetBufferHostSize_16u_C1R	429
7.22.2.32	nppiNormRelL1GetBufferHostSize_16u_C3CMR	429
7.22.2.33	nppiNormRelL1GetBufferHostSize_16u_C3R	430
7.22.2.34	nppiNormRelL1GetBufferHostSize_16u_C4R	430
7.22.2.35	nppiNormRelL1GetBufferHostSize_32f_AC4R	430
7.22.2.36	nppiNormRelL1GetBufferHostSize_32f_C1MR	430
7.22.2.37	nppiNormRelL1GetBufferHostSize_32f_C1R	431
7.22.2.38	nppiNormRelL1GetBufferHostSize_32f_C3CMR	431
7.22.2.39	nppiNormRelL1GetBufferHostSize_32f_C3R	431
7.22.2.40	nppiNormRelL1GetBufferHostSize_32f_C4R	432
7.22.2.41	nppiNormRelL1GetBufferHostSize_8s_C1MR	432
7.22.2.42	nppiNormRelL1GetBufferHostSize_8s_C3CMR	432
7.22.2.43	nppiNormRelL1GetBufferHostSize_8u_AC4R	432
7.22.2.44	nppiNormRelL1GetBufferHostSize_8u_C1MR	433
7.22.2.45	nppiNormRelL1GetBufferHostSize_8u_C1R	433
7.22.2.46	nppiNormRelL1GetBufferHostSize_8u_C3CMR	433
7.22.2.47	nppiNormRelL1GetBufferHostSize_8u_C3R	434
7.22.2.48	nppiNormRelL1GetBufferHostSize_8u_C4R	434
7.23	NormRel_L2	435
7.23.1	Detailed Description	439
7.23.2	Function Documentation	439
7.23.2.1	nppiNormRel_L2_16s_AC4R	439
7.23.2.2	nppiNormRel_L2_16s_C1R	439
7.23.2.3	nppiNormRel_L2_16s_C3R	440
7.23.2.4	nppiNormRel_L2_16s_C4R	440
7.23.2.5	nppiNormRel_L2_16u_AC4R	441
7.23.2.6	nppiNormRel_L2_16u_C1MR	441
7.23.2.7	nppiNormRel_L2_16u_C1R	442
7.23.2.8	nppiNormRel_L2_16u_C3CMR	442

7.23.2.9	nppiNormRel_L2_16u_C3R	443
7.23.2.10	nppiNormRel_L2_16u_C4R	443
7.23.2.11	nppiNormRel_L2_32f_AC4R	443
7.23.2.12	nppiNormRel_L2_32f_C1MR	444
7.23.2.13	nppiNormRel_L2_32f_C1R	444
7.23.2.14	nppiNormRel_L2_32f_C3CMR	445
7.23.2.15	nppiNormRel_L2_32f_C3R	445
7.23.2.16	nppiNormRel_L2_32f_C4R	446
7.23.2.17	nppiNormRel_L2_8s_C1MR	446
7.23.2.18	nppiNormRel_L2_8s_C3CMR	447
7.23.2.19	nppiNormRel_L2_8u_AC4R	447
7.23.2.20	nppiNormRel_L2_8u_C1MR	448
7.23.2.21	nppiNormRel_L2_8u_C1R	448
7.23.2.22	nppiNormRel_L2_8u_C3CMR	449
7.23.2.23	nppiNormRel_L2_8u_C3R	449
7.23.2.24	nppiNormRel_L2_8u_C4R	450
7.23.2.25	nppiNormRelL2GetBufferHostSize_16s_AC4R	450
7.23.2.26	nppiNormRelL2GetBufferHostSize_16s_C1R	451
7.23.2.27	nppiNormRelL2GetBufferHostSize_16s_C3R	451
7.23.2.28	nppiNormRelL2GetBufferHostSize_16s_C4R	451
7.23.2.29	nppiNormRelL2GetBufferHostSize_16u_AC4R	451
7.23.2.30	nppiNormRelL2GetBufferHostSize_16u_C1MR	452
7.23.2.31	nppiNormRelL2GetBufferHostSize_16u_C1R	452
7.23.2.32	nppiNormRelL2GetBufferHostSize_16u_C3CMR	452
7.23.2.33	nppiNormRelL2GetBufferHostSize_16u_C3R	453
7.23.2.34	nppiNormRelL2GetBufferHostSize_16u_C4R	453
7.23.2.35	nppiNormRelL2GetBufferHostSize_32f_AC4R	453
7.23.2.36	nppiNormRelL2GetBufferHostSize_32f_C1MR	453
7.23.2.37	nppiNormRelL2GetBufferHostSize_32f_C1R	454
7.23.2.38	nppiNormRelL2GetBufferHostSize_32f_C3CMR	454
7.23.2.39	nppiNormRelL2GetBufferHostSize_32f_C3R	454
7.23.2.40	nppiNormRelL2GetBufferHostSize_32f_C4R	455
7.23.2.41	nppiNormRelL2GetBufferHostSize_8s_C1MR	455
7.23.2.42	nppiNormRelL2GetBufferHostSize_8s_C3CMR	455
7.23.2.43	nppiNormRelL2GetBufferHostSize_8u_AC4R	455
7.23.2.44	nppiNormRelL2GetBufferHostSize_8u_C1MR	456

7.23.2.45	nppiNormRelL2GetBufferHostSize_8u_C1R	456
7.23.2.46	nppiNormRelL2GetBufferHostSize_8u_C3CMR	456
7.23.2.47	nppiNormRelL2GetBufferHostSize_8u_C3R	457
7.23.2.48	nppiNormRelL2GetBufferHostSize_8u_C4R	457
7.24	DotProd	458
7.24.1	Detailed Description	462
7.24.2	Function Documentation	462
7.24.2.1	nppiDotProd_16s64f_AC4R	462
7.24.2.2	nppiDotProd_16s64f_C1R	463
7.24.2.3	nppiDotProd_16s64f_C3R	463
7.24.2.4	nppiDotProd_16s64f_C4R	463
7.24.2.5	nppiDotProd_16u64f_AC4R	464
7.24.2.6	nppiDotProd_16u64f_C1R	464
7.24.2.7	nppiDotProd_16u64f_C3R	465
7.24.2.8	nppiDotProd_16u64f_C4R	465
7.24.2.9	nppiDotProd_32f64f_AC4R	466
7.24.2.10	nppiDotProd_32f64f_C1R	466
7.24.2.11	nppiDotProd_32f64f_C3R	466
7.24.2.12	nppiDotProd_32f64f_C4R	467
7.24.2.13	nppiDotProd_32s64f_AC4R	467
7.24.2.14	nppiDotProd_32s64f_C1R	468
7.24.2.15	nppiDotProd_32s64f_C3R	468
7.24.2.16	nppiDotProd_32s64f_C4R	469
7.24.2.17	nppiDotProd_32u64f_AC4R	469
7.24.2.18	nppiDotProd_32u64f_C1R	469
7.24.2.19	nppiDotProd_32u64f_C3R	470
7.24.2.20	nppiDotProd_32u64f_C4R	470
7.24.2.21	nppiDotProd_8s64f_AC4R	471
7.24.2.22	nppiDotProd_8s64f_C1R	471
7.24.2.23	nppiDotProd_8s64f_C3R	472
7.24.2.24	nppiDotProd_8s64f_C4R	472
7.24.2.25	nppiDotProd_8u64f_AC4R	472
7.24.2.26	nppiDotProd_8u64f_C1R	473
7.24.2.27	nppiDotProd_8u64f_C3R	473
7.24.2.28	nppiDotProd_8u64f_C4R	474
7.24.2.29	nppiDotProdGetBufferHostSize_16s64f_AC4R	474

7.24.2.30	nppiDotProdGetBufferHostSize_16s64f_C1R	474
7.24.2.31	nppiDotProdGetBufferHostSize_16s64f_C3R	475
7.24.2.32	nppiDotProdGetBufferHostSize_16s64f_C4R	475
7.24.2.33	nppiDotProdGetBufferHostSize_16u64f_AC4R	475
7.24.2.34	nppiDotProdGetBufferHostSize_16u64f_C1R	475
7.24.2.35	nppiDotProdGetBufferHostSize_16u64f_C3R	476
7.24.2.36	nppiDotProdGetBufferHostSize_16u64f_C4R	476
7.24.2.37	nppiDotProdGetBufferHostSize_32f64f_AC4R	476
7.24.2.38	nppiDotProdGetBufferHostSize_32f64f_C1R	477
7.24.2.39	nppiDotProdGetBufferHostSize_32f64f_C3R	477
7.24.2.40	nppiDotProdGetBufferHostSize_32f64f_C4R	477
7.24.2.41	nppiDotProdGetBufferHostSize_32s64f_AC4R	477
7.24.2.42	nppiDotProdGetBufferHostSize_32s64f_C1R	478
7.24.2.43	nppiDotProdGetBufferHostSize_32s64f_C3R	478
7.24.2.44	nppiDotProdGetBufferHostSize_32s64f_C4R	478
7.24.2.45	nppiDotProdGetBufferHostSize_32u64f_AC4R	479
7.24.2.46	nppiDotProdGetBufferHostSize_32u64f_C1R	479
7.24.2.47	nppiDotProdGetBufferHostSize_32u64f_C3R	479
7.24.2.48	nppiDotProdGetBufferHostSize_32u64f_C4R	479
7.24.2.49	nppiDotProdGetBufferHostSize_8s64f_AC4R	480
7.24.2.50	nppiDotProdGetBufferHostSize_8s64f_C1R	480
7.24.2.51	nppiDotProdGetBufferHostSize_8s64f_C3R	480
7.24.2.52	nppiDotProdGetBufferHostSize_8s64f_C4R	481
7.24.2.53	nppiDotProdGetBufferHostSize_8u64f_AC4R	481
7.24.2.54	nppiDotProdGetBufferHostSize_8u64f_C1R	481
7.24.2.55	nppiDotProdGetBufferHostSize_8u64f_C3R	481
7.24.2.56	nppiDotProdGetBufferHostSize_8u64f_C4R	482
7.25	CountInRange.	483
7.25.1	Detailed Description	484
7.25.2	Function Documentation	484
7.25.2.1	nppiCountInRange_32f_AC4R	484
7.25.2.2	nppiCountInRange_32f_C1R	484
7.25.2.3	nppiCountInRange_32f_C3R	485
7.25.2.4	nppiCountInRange_8u_AC4R	485
7.25.2.5	nppiCountInRange_8u_C1R	486
7.25.2.6	nppiCountInRange_8u_C3R	486

7.25.2.7	nppiCountInRangeGetBufferHostSize_32f_AC4R	487
7.25.2.8	nppiCountInRangeGetBufferHostSize_32f_C1R	487
7.25.2.9	nppiCountInRangeGetBufferHostSize_32f_C3R	487
7.25.2.10	nppiCountInRangeGetBufferHostSize_8u_AC4R	488
7.25.2.11	nppiCountInRangeGetBufferHostSize_8u_C1R	488
7.25.2.12	nppiCountInRangeGetBufferHostSize_8u_C3R	488
7.26	MaxEvery	489
7.26.1	Detailed Description	490
7.26.2	Function Documentation	490
7.26.2.1	nppiMaxEvery_16s_AC4IR	490
7.26.2.2	nppiMaxEvery_16s_C1IR	491
7.26.2.3	nppiMaxEvery_16s_C3IR	491
7.26.2.4	nppiMaxEvery_16s_C4IR	491
7.26.2.5	nppiMaxEvery_16u_AC4IR	492
7.26.2.6	nppiMaxEvery_16u_C1IR	492
7.26.2.7	nppiMaxEvery_16u_C3IR	492
7.26.2.8	nppiMaxEvery_16u_C4IR	493
7.26.2.9	nppiMaxEvery_32f_AC4IR	493
7.26.2.10	nppiMaxEvery_32f_C1IR	493
7.26.2.11	nppiMaxEvery_32f_C3IR	494
7.26.2.12	nppiMaxEvery_32f_C4IR	494
7.26.2.13	nppiMaxEvery_8u_AC4IR	494
7.26.2.14	nppiMaxEvery_8u_C1IR	495
7.26.2.15	nppiMaxEvery_8u_C3IR	495
7.26.2.16	nppiMaxEvery_8u_C4IR	495
7.27	MinEvery	496
7.27.1	Detailed Description	497
7.27.2	Function Documentation	497
7.27.2.1	nppiMinEvery_16s_AC4IR	497
7.27.2.2	nppiMinEvery_16s_C1IR	498
7.27.2.3	nppiMinEvery_16s_C3IR	498
7.27.2.4	nppiMinEvery_16s_C4IR	498
7.27.2.5	nppiMinEvery_16u_AC4IR	499
7.27.2.6	nppiMinEvery_16u_C1IR	499
7.27.2.7	nppiMinEvery_16u_C3IR	499
7.27.2.8	nppiMinEvery_16u_C4IR	500

7.27.2.9	<code>nppiMinEvery_32f_AC4IR</code>	500
7.27.2.10	<code>nppiMinEvery_32f_C1IR</code>	500
7.27.2.11	<code>nppiMinEvery_32f_C3IR</code>	501
7.27.2.12	<code>nppiMinEvery_32f_C4IR</code>	501
7.27.2.13	<code>nppiMinEvery_8u_AC4IR</code>	501
7.27.2.14	<code>nppiMinEvery_8u_C1IR</code>	502
7.27.2.15	<code>nppiMinEvery_8u_C3IR</code>	502
7.27.2.16	<code>nppiMinEvery_8u_C4IR</code>	502
7.28	Integral	503
7.28.1	Detailed Description	503
7.28.2	Function Documentation	503
7.28.2.1	<code>nppiIntegral_8u32f_C1R</code>	503
7.28.2.2	<code>nppiIntegral_8u32s_C1R</code>	504
7.29	SqrIntegral	505
7.29.1	Detailed Description	505
7.29.2	Function Documentation	505
7.29.2.1	<code>nppiSqrIntegral_8u32f64f_C1R</code>	505
7.29.2.2	<code>nppiSqrIntegral_8u32s64f_C1R</code>	506
7.29.2.3	<code>nppiSqrIntegral_8u32s_C1R</code>	506
7.30	RectStdDev	508
7.30.1	Detailed Description	508
7.30.2	Function Documentation	508
7.30.2.1	<code>nppiRectStdDev_32f_C1R</code>	508
7.30.2.2	<code>nppiRectStdDev_32s32f_C1R</code>	509
7.30.2.3	<code>nppiRectStdDev_32s_C1RSfs</code>	509
7.31	HistogramEven	511
7.31.1	Detailed Description	513
7.31.2	Function Documentation	513
7.31.2.1	<code>nppiEvenLevelsHost_32s</code>	513
7.31.2.2	<code>nppiHistogramEven_16s_AC4R</code>	514
7.31.2.3	<code>nppiHistogramEven_16s_C1R</code>	514
7.31.2.4	<code>nppiHistogramEven_16s_C3R</code>	515
7.31.2.5	<code>nppiHistogramEven_16s_C4R</code>	515
7.31.2.6	<code>nppiHistogramEven_16u_AC4R</code>	516
7.31.2.7	<code>nppiHistogramEven_16u_C1R</code>	516
7.31.2.8	<code>nppiHistogramEven_16u_C3R</code>	517

7.31.2.9	<code>nppiHistogramEven_16u_C4R</code>	517
7.31.2.10	<code>nppiHistogramEven_8u_AC4R</code>	518
7.31.2.11	<code>nppiHistogramEven_8u_C1R</code>	518
7.31.2.12	<code>nppiHistogramEven_8u_C3R</code>	518
7.31.2.13	<code>nppiHistogramEven_8u_C4R</code>	519
7.31.2.14	<code>nppiHistogramEvenGetBufferSize_16s_AC4R</code>	519
7.31.2.15	<code>nppiHistogramEvenGetBufferSize_16s_C1R</code>	520
7.31.2.16	<code>nppiHistogramEvenGetBufferSize_16s_C3R</code>	520
7.31.2.17	<code>nppiHistogramEvenGetBufferSize_16s_C4R</code>	520
7.31.2.18	<code>nppiHistogramEvenGetBufferSize_16u_AC4R</code>	521
7.31.2.19	<code>nppiHistogramEvenGetBufferSize_16u_C1R</code>	521
7.31.2.20	<code>nppiHistogramEvenGetBufferSize_16u_C3R</code>	521
7.31.2.21	<code>nppiHistogramEvenGetBufferSize_16u_C4R</code>	522
7.31.2.22	<code>nppiHistogramEvenGetBufferSize_8u_AC4R</code>	522
7.31.2.23	<code>nppiHistogramEvenGetBufferSize_8u_C1R</code>	522
7.31.2.24	<code>nppiHistogramEvenGetBufferSize_8u_C3R</code>	523
7.31.2.25	<code>nppiHistogramEvenGetBufferSize_8u_C4R</code>	523
7.32	<code>HistogramRange</code>	524
7.32.1	Detailed Description	526
7.32.2	Function Documentation	527
7.32.2.1	<code>nppiHistogramRange_16s_AC4R</code>	527
7.32.2.2	<code>nppiHistogramRange_16s_C1R</code>	527
7.32.2.3	<code>nppiHistogramRange_16s_C3R</code>	527
7.32.2.4	<code>nppiHistogramRange_16s_C4R</code>	528
7.32.2.5	<code>nppiHistogramRange_16u_AC4R</code>	528
7.32.2.6	<code>nppiHistogramRange_16u_C1R</code>	529
7.32.2.7	<code>nppiHistogramRange_16u_C3R</code>	529
7.32.2.8	<code>nppiHistogramRange_16u_C4R</code>	530
7.32.2.9	<code>nppiHistogramRange_32f_AC4R</code>	530
7.32.2.10	<code>nppiHistogramRange_32f_C1R</code>	531
7.32.2.11	<code>nppiHistogramRange_32f_C3R</code>	531
7.32.2.12	<code>nppiHistogramRange_32f_C4R</code>	531
7.32.2.13	<code>nppiHistogramRange_8u_AC4R</code>	532
7.32.2.14	<code>nppiHistogramRange_8u_C1R</code>	532
7.32.2.15	<code>nppiHistogramRange_8u_C3R</code>	533
7.32.2.16	<code>nppiHistogramRange_8u_C4R</code>	533

7.32.2.17	nppiHistogramRangeGetBufferSize_16s_AC4R	534
7.32.2.18	nppiHistogramRangeGetBufferSize_16s_C1R	534
7.32.2.19	nppiHistogramRangeGetBufferSize_16s_C3R	534
7.32.2.20	nppiHistogramRangeGetBufferSize_16s_C4R	535
7.32.2.21	nppiHistogramRangeGetBufferSize_16u_AC4R	535
7.32.2.22	nppiHistogramRangeGetBufferSize_16u_C1R	535
7.32.2.23	nppiHistogramRangeGetBufferSize_16u_C3R	536
7.32.2.24	nppiHistogramRangeGetBufferSize_16u_C4R	536
7.32.2.25	nppiHistogramRangeGetBufferSize_32f_AC4R	536
7.32.2.26	nppiHistogramRangeGetBufferSize_32f_C1R	537
7.32.2.27	nppiHistogramRangeGetBufferSize_32f_C3R	537
7.32.2.28	nppiHistogramRangeGetBufferSize_32f_C4R	537
7.32.2.29	nppiHistogramRangeGetBufferSize_8u_AC4R	538
7.32.2.30	nppiHistogramRangeGetBufferSize_8u_C1R	538
7.32.2.31	nppiHistogramRangeGetBufferSize_8u_C3R	538
7.32.2.32	nppiHistogramRangeGetBufferSize_8u_C4R	539
7.33	Image Proximity	540
7.33.1	Detailed Description	540
7.33.2	General Introduction	540
7.33.3	Categorizations	542
7.34	SqrDistanceFull_Norm	543
7.34.1	Detailed Description	544
7.34.2	Function Documentation	545
7.34.2.1	nppiSqrDistanceFull_Norm_16u32f_AC4R	545
7.34.2.2	nppiSqrDistanceFull_Norm_16u32f_C1R	545
7.34.2.3	nppiSqrDistanceFull_Norm_16u32f_C3R	546
7.34.2.4	nppiSqrDistanceFull_Norm_16u32f_C4R	546
7.34.2.5	nppiSqrDistanceFull_Norm_32f_AC4R	546
7.34.2.6	nppiSqrDistanceFull_Norm_32f_C1R	547
7.34.2.7	nppiSqrDistanceFull_Norm_32f_C3R	547
7.34.2.8	nppiSqrDistanceFull_Norm_32f_C4R	548
7.34.2.9	nppiSqrDistanceFull_Norm_8s32f_AC4R	548
7.34.2.10	nppiSqrDistanceFull_Norm_8s32f_C1R	549
7.34.2.11	nppiSqrDistanceFull_Norm_8s32f_C3R	549
7.34.2.12	nppiSqrDistanceFull_Norm_8s32f_C4R	549
7.34.2.13	nppiSqrDistanceFull_Norm_8u32f_AC4R	550

7.34.2.14	nppiSqrDistanceFull_Norm_8u32f_C1R	550
7.34.2.15	nppiSqrDistanceFull_Norm_8u32f_C3R	551
7.34.2.16	nppiSqrDistanceFull_Norm_8u32f_C4R	551
7.34.2.17	nppiSqrDistanceFull_Norm_8u_AC4RSfs	552
7.34.2.18	nppiSqrDistanceFull_Norm_8u_C1RSfs	552
7.34.2.19	nppiSqrDistanceFull_Norm_8u_C3RSfs	553
7.34.2.20	nppiSqrDistanceFull_Norm_8u_C4RSfs	553
7.35	SqrDistanceSame_Norm	554
7.35.1	Detailed Description	556
7.35.2	Function Documentation	556
7.35.2.1	nppiSqrDistanceSame_Norm_16u32f_AC4R	556
7.35.2.2	nppiSqrDistanceSame_Norm_16u32f_C1R	556
7.35.2.3	nppiSqrDistanceSame_Norm_16u32f_C3R	557
7.35.2.4	nppiSqrDistanceSame_Norm_16u32f_C4R	557
7.35.2.5	nppiSqrDistanceSame_Norm_32f_AC4R	558
7.35.2.6	nppiSqrDistanceSame_Norm_32f_C1R	558
7.35.2.7	nppiSqrDistanceSame_Norm_32f_C3R	558
7.35.2.8	nppiSqrDistanceSame_Norm_32f_C4R	559
7.35.2.9	nppiSqrDistanceSame_Norm_8s32f_AC4R	559
7.35.2.10	nppiSqrDistanceSame_Norm_8s32f_C1R	560
7.35.2.11	nppiSqrDistanceSame_Norm_8s32f_C3R	560
7.35.2.12	nppiSqrDistanceSame_Norm_8s32f_C4R	561
7.35.2.13	nppiSqrDistanceSame_Norm_8u32f_AC4R	561
7.35.2.14	nppiSqrDistanceSame_Norm_8u32f_C1R	561
7.35.2.15	nppiSqrDistanceSame_Norm_8u32f_C3R	562
7.35.2.16	nppiSqrDistanceSame_Norm_8u32f_C4R	562
7.35.2.17	nppiSqrDistanceSame_Norm_8u_AC4RSfs	563
7.35.2.18	nppiSqrDistanceSame_Norm_8u_C1RSfs	563
7.35.2.19	nppiSqrDistanceSame_Norm_8u_C3RSfs	564
7.35.2.20	nppiSqrDistanceSame_Norm_8u_C4RSfs	564
7.36	SqrDistanceValid_Norm	565
7.36.1	Detailed Description	567
7.36.2	Function Documentation	567
7.36.2.1	nppiSqrDistanceValid_Norm_16u32f_AC4R	567
7.36.2.2	nppiSqrDistanceValid_Norm_16u32f_C1R	567
7.36.2.3	nppiSqrDistanceValid_Norm_16u32f_C3R	568

7.36.2.4	nppiSqrDistanceValid_Norm_16u32f_C4R	568
7.36.2.5	nppiSqrDistanceValid_Norm_32f_AC4R	569
7.36.2.6	nppiSqrDistanceValid_Norm_32f_C1R	569
7.36.2.7	nppiSqrDistanceValid_Norm_32f_C3R	569
7.36.2.8	nppiSqrDistanceValid_Norm_32f_C4R	570
7.36.2.9	nppiSqrDistanceValid_Norm_8s32f_AC4R	570
7.36.2.10	nppiSqrDistanceValid_Norm_8s32f_C1R	571
7.36.2.11	nppiSqrDistanceValid_Norm_8s32f_C3R	571
7.36.2.12	nppiSqrDistanceValid_Norm_8s32f_C4R	572
7.36.2.13	nppiSqrDistanceValid_Norm_8u32f_AC4R	572
7.36.2.14	nppiSqrDistanceValid_Norm_8u32f_C1R	572
7.36.2.15	nppiSqrDistanceValid_Norm_8u32f_C3R	573
7.36.2.16	nppiSqrDistanceValid_Norm_8u32f_C4R	573
7.36.2.17	nppiSqrDistanceValid_Norm_8u_AC4RSfs	574
7.36.2.18	nppiSqrDistanceValid_Norm_8u_C1RSfs	574
7.36.2.19	nppiSqrDistanceValid_Norm_8u_C3RSfs	575
7.36.2.20	nppiSqrDistanceValid_Norm_8u_C4RSfs	575
7.37	CrossCorrFull_Norm	576
7.37.1	Detailed Description	577
7.37.2	Function Documentation	578
7.37.2.1	nppiCrossCorrFull_Norm_16u32f_AC4R	578
7.37.2.2	nppiCrossCorrFull_Norm_16u32f_C1R	578
7.37.2.3	nppiCrossCorrFull_Norm_16u32f_C3R	579
7.37.2.4	nppiCrossCorrFull_Norm_16u32f_C4R	579
7.37.2.5	nppiCrossCorrFull_Norm_32f_AC4R	579
7.37.2.6	nppiCrossCorrFull_Norm_32f_C1R	580
7.37.2.7	nppiCrossCorrFull_Norm_32f_C3R	580
7.37.2.8	nppiCrossCorrFull_Norm_32f_C4R	581
7.37.2.9	nppiCrossCorrFull_Norm_8s32f_AC4R	581
7.37.2.10	nppiCrossCorrFull_Norm_8s32f_C1R	582
7.37.2.11	nppiCrossCorrFull_Norm_8s32f_C3R	582
7.37.2.12	nppiCrossCorrFull_Norm_8s32f_C4R	582
7.37.2.13	nppiCrossCorrFull_Norm_8u32f_AC4R	583
7.37.2.14	nppiCrossCorrFull_Norm_8u32f_C1R	583
7.37.2.15	nppiCrossCorrFull_Norm_8u32f_C3R	584
7.37.2.16	nppiCrossCorrFull_Norm_8u32f_C4R	584

7.37.2.17	nppiCrossCorrFull_Norm_8u_AC4RSfs	585
7.37.2.18	nppiCrossCorrFull_Norm_8u_C1RSfs	585
7.37.2.19	nppiCrossCorrFull_Norm_8u_C3RSfs	586
7.37.2.20	nppiCrossCorrFull_Norm_8u_C4RSfs	586
7.38	CrossCorrSame_Norm	587
7.38.1	Detailed Description	588
7.38.2	Function Documentation	589
7.38.2.1	nppiCrossCorrSame_Norm_16u32f_AC4R	589
7.38.2.2	nppiCrossCorrSame_Norm_16u32f_C1R	589
7.38.2.3	nppiCrossCorrSame_Norm_16u32f_C3R	590
7.38.2.4	nppiCrossCorrSame_Norm_16u32f_C4R	590
7.38.2.5	nppiCrossCorrSame_Norm_32f_AC4R	590
7.38.2.6	nppiCrossCorrSame_Norm_32f_C1R	591
7.38.2.7	nppiCrossCorrSame_Norm_32f_C3R	591
7.38.2.8	nppiCrossCorrSame_Norm_32f_C4R	592
7.38.2.9	nppiCrossCorrSame_Norm_8s32f_AC4R	592
7.38.2.10	nppiCrossCorrSame_Norm_8s32f_C1R	593
7.38.2.11	nppiCrossCorrSame_Norm_8s32f_C3R	593
7.38.2.12	nppiCrossCorrSame_Norm_8s32f_C4R	593
7.38.2.13	nppiCrossCorrSame_Norm_8u32f_AC4R	594
7.38.2.14	nppiCrossCorrSame_Norm_8u32f_C1R	594
7.38.2.15	nppiCrossCorrSame_Norm_8u32f_C3R	595
7.38.2.16	nppiCrossCorrSame_Norm_8u32f_C4R	595
7.38.2.17	nppiCrossCorrSame_Norm_8u_AC4RSfs	596
7.38.2.18	nppiCrossCorrSame_Norm_8u_C1RSfs	596
7.38.2.19	nppiCrossCorrSame_Norm_8u_C3RSfs	597
7.38.2.20	nppiCrossCorrSame_Norm_8u_C4RSfs	597
7.39	CrossCorrValid_Norm	598
7.39.1	Detailed Description	599
7.39.2	Function Documentation	600
7.39.2.1	nppiCrossCorrValid_Norm_16u32f_AC4R	600
7.39.2.2	nppiCrossCorrValid_Norm_16u32f_C1R	600
7.39.2.3	nppiCrossCorrValid_Norm_16u32f_C3R	601
7.39.2.4	nppiCrossCorrValid_Norm_16u32f_C4R	601
7.39.2.5	nppiCrossCorrValid_Norm_32f_AC4R	601
7.39.2.6	nppiCrossCorrValid_Norm_32f_C1R	602

7.39.2.7	nppiCrossCorrValid_Norm_32f_C3R	602
7.39.2.8	nppiCrossCorrValid_Norm_32f_C4R	603
7.39.2.9	nppiCrossCorrValid_Norm_8s32f_AC4R	603
7.39.2.10	nppiCrossCorrValid_Norm_8s32f_C1R	604
7.39.2.11	nppiCrossCorrValid_Norm_8s32f_C3R	604
7.39.2.12	nppiCrossCorrValid_Norm_8s32f_C4R	604
7.39.2.13	nppiCrossCorrValid_Norm_8u32f_AC4R	605
7.39.2.14	nppiCrossCorrValid_Norm_8u32f_C1R	605
7.39.2.15	nppiCrossCorrValid_Norm_8u32f_C3R	606
7.39.2.16	nppiCrossCorrValid_Norm_8u32f_C4R	606
7.39.2.17	nppiCrossCorrValid_Norm_8u_AC4RSfs	607
7.39.2.18	nppiCrossCorrValid_Norm_8u_C1RSfs	607
7.39.2.19	nppiCrossCorrValid_Norm_8u_C3RSfs	608
7.39.2.20	nppiCrossCorrValid_Norm_8u_C4RSfs	608
7.40	CrossCorrValid	609
7.40.1	Detailed Description	609
7.40.2	Function Documentation	609
7.40.2.1	nppiCrossCorrValid_16u32f_C1R	609
7.40.2.2	nppiCrossCorrValid_32f_C1R	610
7.40.2.3	nppiCrossCorrValid_8s32f_C1R	610
7.40.2.4	nppiCrossCorrValid_8u32f_C1R	611
7.41	CrossCorrFull_NormLevel	612
7.41.1	Detailed Description	615
7.41.2	Function Documentation	616
7.41.2.1	nppiCrossCorrFull_NormLevel_16u32f_AC4R	616
7.41.2.2	nppiCrossCorrFull_NormLevel_16u32f_C1R	616
7.41.2.3	nppiCrossCorrFull_NormLevel_16u32f_C3R	617
7.41.2.4	nppiCrossCorrFull_NormLevel_16u32f_C4R	617
7.41.2.5	nppiCrossCorrFull_NormLevel_32f_AC4R	618
7.41.2.6	nppiCrossCorrFull_NormLevel_32f_C1R	618
7.41.2.7	nppiCrossCorrFull_NormLevel_32f_C3R	619
7.41.2.8	nppiCrossCorrFull_NormLevel_32f_C4R	619
7.41.2.9	nppiCrossCorrFull_NormLevel_8s32f_AC4R	620
7.41.2.10	nppiCrossCorrFull_NormLevel_8s32f_C1R	620
7.41.2.11	nppiCrossCorrFull_NormLevel_8s32f_C3R	621
7.41.2.12	nppiCrossCorrFull_NormLevel_8s32f_C4R	621

7.41.2.13	<code>nppiCrossCorrFull_NormLevel_8u32f_AC4R</code>	622
7.41.2.14	<code>nppiCrossCorrFull_NormLevel_8u32f_C1R</code>	622
7.41.2.15	<code>nppiCrossCorrFull_NormLevel_8u32f_C3R</code>	623
7.41.2.16	<code>nppiCrossCorrFull_NormLevel_8u32f_C4R</code>	623
7.41.2.17	<code>nppiCrossCorrFull_NormLevel_8u_AC4RSfs</code>	624
7.41.2.18	<code>nppiCrossCorrFull_NormLevel_8u_C1RSfs</code>	624
7.41.2.19	<code>nppiCrossCorrFull_NormLevel_8u_C3RSfs</code>	625
7.41.2.20	<code>nppiCrossCorrFull_NormLevel_8u_C4RSfs</code>	625
7.41.2.21	<code>nppiFullNormLevelGetBufferHostSize_16u32f_AC4R</code>	626
7.41.2.22	<code>nppiFullNormLevelGetBufferHostSize_16u32f_C1R</code>	626
7.41.2.23	<code>nppiFullNormLevelGetBufferHostSize_16u32f_C3R</code>	626
7.41.2.24	<code>nppiFullNormLevelGetBufferHostSize_16u32f_C4R</code>	626
7.41.2.25	<code>nppiFullNormLevelGetBufferHostSize_32f_AC4R</code>	627
7.41.2.26	<code>nppiFullNormLevelGetBufferHostSize_32f_C1R</code>	627
7.41.2.27	<code>nppiFullNormLevelGetBufferHostSize_32f_C3R</code>	627
7.41.2.28	<code>nppiFullNormLevelGetBufferHostSize_32f_C4R</code>	628
7.41.2.29	<code>nppiFullNormLevelGetBufferHostSize_8s32f_AC4R</code>	628
7.41.2.30	<code>nppiFullNormLevelGetBufferHostSize_8s32f_C1R</code>	628
7.41.2.31	<code>nppiFullNormLevelGetBufferHostSize_8s32f_C3R</code>	628
7.41.2.32	<code>nppiFullNormLevelGetBufferHostSize_8s32f_C4R</code>	629
7.41.2.33	<code>nppiFullNormLevelGetBufferHostSize_8u32f_AC4R</code>	629
7.41.2.34	<code>nppiFullNormLevelGetBufferHostSize_8u32f_C1R</code>	629
7.41.2.35	<code>nppiFullNormLevelGetBufferHostSize_8u32f_C3R</code>	630
7.41.2.36	<code>nppiFullNormLevelGetBufferHostSize_8u32f_C4R</code>	630
7.41.2.37	<code>nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs</code>	630
7.41.2.38	<code>nppiFullNormLevelGetBufferHostSize_8u_C1RSfs</code>	630
7.41.2.39	<code>nppiFullNormLevelGetBufferHostSize_8u_C3RSfs</code>	631
7.41.2.40	<code>nppiFullNormLevelGetBufferHostSize_8u_C4RSfs</code>	631
7.42	<code>CrossCorrSame_NormLevel</code>	632
7.42.1	Detailed Description	635
7.42.2	Function Documentation	636
7.42.2.1	<code>nppiCrossCorrSame_NormLevel_16u32f_AC4R</code>	636
7.42.2.2	<code>nppiCrossCorrSame_NormLevel_16u32f_C1R</code>	636
7.42.2.3	<code>nppiCrossCorrSame_NormLevel_16u32f_C3R</code>	637
7.42.2.4	<code>nppiCrossCorrSame_NormLevel_16u32f_C4R</code>	637
7.42.2.5	<code>nppiCrossCorrSame_NormLevel_32f_AC4R</code>	638

7.42.2.6	nppiCrossCorrSame_NormLevel_32f_C1R	638
7.42.2.7	nppiCrossCorrSame_NormLevel_32f_C3R	639
7.42.2.8	nppiCrossCorrSame_NormLevel_32f_C4R	639
7.42.2.9	nppiCrossCorrSame_NormLevel_8s32f_AC4R	640
7.42.2.10	nppiCrossCorrSame_NormLevel_8s32f_C1R	640
7.42.2.11	nppiCrossCorrSame_NormLevel_8s32f_C3R	641
7.42.2.12	nppiCrossCorrSame_NormLevel_8s32f_C4R	641
7.42.2.13	nppiCrossCorrSame_NormLevel_8u32f_AC4R	642
7.42.2.14	nppiCrossCorrSame_NormLevel_8u32f_C1R	642
7.42.2.15	nppiCrossCorrSame_NormLevel_8u32f_C3R	643
7.42.2.16	nppiCrossCorrSame_NormLevel_8u32f_C4R	643
7.42.2.17	nppiCrossCorrSame_NormLevel_8u_AC4RSfs	644
7.42.2.18	nppiCrossCorrSame_NormLevel_8u_C1RSfs	644
7.42.2.19	nppiCrossCorrSame_NormLevel_8u_C3RSfs	645
7.42.2.20	nppiCrossCorrSame_NormLevel_8u_C4RSfs	645
7.42.2.21	nppiSameNormLevelGetBufferHostSize_16u32f_AC4R	646
7.42.2.22	nppiSameNormLevelGetBufferHostSize_16u32f_C1R	646
7.42.2.23	nppiSameNormLevelGetBufferHostSize_16u32f_C3R	646
7.42.2.24	nppiSameNormLevelGetBufferHostSize_16u32f_C4R	646
7.42.2.25	nppiSameNormLevelGetBufferHostSize_32f_AC4R	647
7.42.2.26	nppiSameNormLevelGetBufferHostSize_32f_C1R	647
7.42.2.27	nppiSameNormLevelGetBufferHostSize_32f_C3R	647
7.42.2.28	nppiSameNormLevelGetBufferHostSize_32f_C4R	648
7.42.2.29	nppiSameNormLevelGetBufferHostSize_8s32f_AC4R	648
7.42.2.30	nppiSameNormLevelGetBufferHostSize_8s32f_C1R	648
7.42.2.31	nppiSameNormLevelGetBufferHostSize_8s32f_C3R	648
7.42.2.32	nppiSameNormLevelGetBufferHostSize_8s32f_C4R	649
7.42.2.33	nppiSameNormLevelGetBufferHostSize_8u32f_AC4R	649
7.42.2.34	nppiSameNormLevelGetBufferHostSize_8u32f_C1R	649
7.42.2.35	nppiSameNormLevelGetBufferHostSize_8u32f_C3R	650
7.42.2.36	nppiSameNormLevelGetBufferHostSize_8u32f_C4R	650
7.42.2.37	nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs	650
7.42.2.38	nppiSameNormLevelGetBufferHostSize_8u_C1RSfs	650
7.42.2.39	nppiSameNormLevelGetBufferHostSize_8u_C3RSfs	651
7.42.2.40	nppiSameNormLevelGetBufferHostSize_8u_C4RSfs	651
7.43	CrossCorrValid_NormLevel	652

7.43.1	Detailed Description	655
7.43.2	Function Documentation	656
7.43.2.1	nppiCrossCorrValid_NormLevel_16u32f_AC4R	656
7.43.2.2	nppiCrossCorrValid_NormLevel_16u32f_C1R	656
7.43.2.3	nppiCrossCorrValid_NormLevel_16u32f_C3R	657
7.43.2.4	nppiCrossCorrValid_NormLevel_16u32f_C4R	657
7.43.2.5	nppiCrossCorrValid_NormLevel_32f_AC4R	658
7.43.2.6	nppiCrossCorrValid_NormLevel_32f_C1R	658
7.43.2.7	nppiCrossCorrValid_NormLevel_32f_C3R	659
7.43.2.8	nppiCrossCorrValid_NormLevel_32f_C4R	659
7.43.2.9	nppiCrossCorrValid_NormLevel_8s32f_AC4R	660
7.43.2.10	nppiCrossCorrValid_NormLevel_8s32f_C1R	660
7.43.2.11	nppiCrossCorrValid_NormLevel_8s32f_C3R	661
7.43.2.12	nppiCrossCorrValid_NormLevel_8s32f_C4R	661
7.43.2.13	nppiCrossCorrValid_NormLevel_8u32f_AC4R	662
7.43.2.14	nppiCrossCorrValid_NormLevel_8u32f_C1R	662
7.43.2.15	nppiCrossCorrValid_NormLevel_8u32f_C3R	663
7.43.2.16	nppiCrossCorrValid_NormLevel_8u32f_C4R	663
7.43.2.17	nppiCrossCorrValid_NormLevel_8u_AC4RSfs	664
7.43.2.18	nppiCrossCorrValid_NormLevel_8u_C1RSfs	664
7.43.2.19	nppiCrossCorrValid_NormLevel_8u_C3RSfs	665
7.43.2.20	nppiCrossCorrValid_NormLevel_8u_C4RSfs	665
7.43.2.21	nppiValidNormLevelGetBufferHostSize_16u32f_AC4R	666
7.43.2.22	nppiValidNormLevelGetBufferHostSize_16u32f_C1R	666
7.43.2.23	nppiValidNormLevelGetBufferHostSize_16u32f_C3R	666
7.43.2.24	nppiValidNormLevelGetBufferHostSize_16u32f_C4R	666
7.43.2.25	nppiValidNormLevelGetBufferHostSize_32f_AC4R	667
7.43.2.26	nppiValidNormLevelGetBufferHostSize_32f_C1R	667
7.43.2.27	nppiValidNormLevelGetBufferHostSize_32f_C3R	667
7.43.2.28	nppiValidNormLevelGetBufferHostSize_32f_C4R	668
7.43.2.29	nppiValidNormLevelGetBufferHostSize_8s32f_AC4R	668
7.43.2.30	nppiValidNormLevelGetBufferHostSize_8s32f_C1R	668
7.43.2.31	nppiValidNormLevelGetBufferHostSize_8s32f_C3R	668
7.43.2.32	nppiValidNormLevelGetBufferHostSize_8s32f_C4R	669
7.43.2.33	nppiValidNormLevelGetBufferHostSize_8u32f_AC4R	669
7.43.2.34	nppiValidNormLevelGetBufferHostSize_8u32f_C1R	669

7.43.2.35	<code>nppiValidNormLevelGetBufferHostSize_8u32f_C3R</code>	670
7.43.2.36	<code>nppiValidNormLevelGetBufferHostSize_8u32f_C4R</code>	670
7.43.2.37	<code>nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs</code>	670
7.43.2.38	<code>nppiValidNormLevelGetBufferHostSize_8u_C1RSfs</code>	670
7.43.2.39	<code>nppiValidNormLevelGetBufferHostSize_8u_C3RSfs</code>	671
7.43.2.40	<code>nppiValidNormLevelGetBufferHostSize_8u_C4RSfs</code>	671
7.44	Image Quality Index	672
7.44.1	Detailed Description	674
7.44.2	Function Documentation	674
7.44.2.1	<code>nppiQualityIndex_16u32f_AC4R</code>	674
7.44.2.2	<code>nppiQualityIndex_16u32f_C1R</code>	674
7.44.2.3	<code>nppiQualityIndex_16u32f_C3R</code>	675
7.44.2.4	<code>nppiQualityIndex_32f_AC4R</code>	675
7.44.2.5	<code>nppiQualityIndex_32f_C1R</code>	676
7.44.2.6	<code>nppiQualityIndex_32f_C3R</code>	676
7.44.2.7	<code>nppiQualityIndex_8u32f_AC4R</code>	677
7.44.2.8	<code>nppiQualityIndex_8u32f_C1R</code>	677
7.44.2.9	<code>nppiQualityIndex_8u32f_C3R</code>	677
7.44.2.10	<code>nppiQualityIndexGetBufferHostSize_16u32f_AC4R</code>	678
7.44.2.11	<code>nppiQualityIndexGetBufferHostSize_16u32f_C1R</code>	678
7.44.2.12	<code>nppiQualityIndexGetBufferHostSize_16u32f_C3R</code>	679
7.44.2.13	<code>nppiQualityIndexGetBufferHostSize_32f_AC4R</code>	679
7.44.2.14	<code>nppiQualityIndexGetBufferHostSize_32f_C1R</code>	679
7.44.2.15	<code>nppiQualityIndexGetBufferHostSize_32f_C3R</code>	679
7.44.2.16	<code>nppiQualityIndexGetBufferHostSize_8u32f_AC4R</code>	680
7.44.2.17	<code>nppiQualityIndexGetBufferHostSize_8u32f_C1R</code>	680
7.44.2.18	<code>nppiQualityIndexGetBufferHostSize_8u32f_C3R</code>	680
7.45	MaximumError	681
7.45.1	Detailed Description	684
7.45.2	Function Documentation	684
7.45.2.1	<code>nppiMaximumError_16s_C1R</code>	684
7.45.2.2	<code>nppiMaximumError_16s_C2R</code>	685
7.45.2.3	<code>nppiMaximumError_16s_C3R</code>	685
7.45.2.4	<code>nppiMaximumError_16s_C4R</code>	686
7.45.2.5	<code>nppiMaximumError_16sc_C1R</code>	686
7.45.2.6	<code>nppiMaximumError_16sc_C2R</code>	686

7.45.2.7	nppiMaximumError_16sc_C3R	687
7.45.2.8	nppiMaximumError_16sc_C4R	687
7.45.2.9	nppiMaximumError_16u_C1R	688
7.45.2.10	nppiMaximumError_16u_C2R	688
7.45.2.11	nppiMaximumError_16u_C3R	689
7.45.2.12	nppiMaximumError_16u_C4R	689
7.45.2.13	nppiMaximumError_32f_C1R	689
7.45.2.14	nppiMaximumError_32f_C2R	690
7.45.2.15	nppiMaximumError_32f_C3R	690
7.45.2.16	nppiMaximumError_32f_C4R	691
7.45.2.17	nppiMaximumError_32fc_C1R	691
7.45.2.18	nppiMaximumError_32fc_C2R	692
7.45.2.19	nppiMaximumError_32fc_C3R	692
7.45.2.20	nppiMaximumError_32fc_C4R	693
7.45.2.21	nppiMaximumError_32s_C1R	693
7.45.2.22	nppiMaximumError_32s_C2R	693
7.45.2.23	nppiMaximumError_32s_C3R	694
7.45.2.24	nppiMaximumError_32s_C4R	694
7.45.2.25	nppiMaximumError_32sc_C1R	695
7.45.2.26	nppiMaximumError_32sc_C2R	695
7.45.2.27	nppiMaximumError_32sc_C3R	696
7.45.2.28	nppiMaximumError_32sc_C4R	696
7.45.2.29	nppiMaximumError_32u_C1R	696
7.45.2.30	nppiMaximumError_32u_C2R	697
7.45.2.31	nppiMaximumError_32u_C3R	697
7.45.2.32	nppiMaximumError_32u_C4R	698
7.45.2.33	nppiMaximumError_64f_C1R	698
7.45.2.34	nppiMaximumError_64f_C2R	699
7.45.2.35	nppiMaximumError_64f_C3R	699
7.45.2.36	nppiMaximumError_64f_C4R	699
7.45.2.37	nppiMaximumError_8s_C1R	700
7.45.2.38	nppiMaximumError_8s_C2R	700
7.45.2.39	nppiMaximumError_8s_C3R	701
7.45.2.40	nppiMaximumError_8s_C4R	701
7.45.2.41	nppiMaximumError_8u_C1R	702
7.45.2.42	nppiMaximumError_8u_C2R	702

7.45.2.43	<code>nppiMaximumError_8u_C3R</code>	702
7.45.2.44	<code>nppiMaximumError_8u_C4R</code>	703
7.46	<code>AverageError</code>	704
7.46.1	Detailed Description	707
7.46.2	Function Documentation	707
7.46.2.1	<code>nppiAverageError_16s_C1R</code>	707
7.46.2.2	<code>nppiAverageError_16s_C2R</code>	708
7.46.2.3	<code>nppiAverageError_16s_C3R</code>	708
7.46.2.4	<code>nppiAverageError_16s_C4R</code>	709
7.46.2.5	<code>nppiAverageError_16sc_C1R</code>	709
7.46.2.6	<code>nppiAverageError_16sc_C2R</code>	710
7.46.2.7	<code>nppiAverageError_16sc_C3R</code>	710
7.46.2.8	<code>nppiAverageError_16sc_C4R</code>	710
7.46.2.9	<code>nppiAverageError_16u_C1R</code>	711
7.46.2.10	<code>nppiAverageError_16u_C2R</code>	711
7.46.2.11	<code>nppiAverageError_16u_C3R</code>	712
7.46.2.12	<code>nppiAverageError_16u_C4R</code>	712
7.46.2.13	<code>nppiAverageError_32f_C1R</code>	713
7.46.2.14	<code>nppiAverageError_32f_C2R</code>	713
7.46.2.15	<code>nppiAverageError_32f_C3R</code>	713
7.46.2.16	<code>nppiAverageError_32f_C4R</code>	714
7.46.2.17	<code>nppiAverageError_32fc_C1R</code>	714
7.46.2.18	<code>nppiAverageError_32fc_C2R</code>	715
7.46.2.19	<code>nppiAverageError_32fc_C3R</code>	715
7.46.2.20	<code>nppiAverageError_32fc_C4R</code>	716
7.46.2.21	<code>nppiAverageError_32s_C1R</code>	716
7.46.2.22	<code>nppiAverageError_32s_C2R</code>	717
7.46.2.23	<code>nppiAverageError_32s_C3R</code>	717
7.46.2.24	<code>nppiAverageError_32s_C4R</code>	717
7.46.2.25	<code>nppiAverageError_32sc_C1R</code>	718
7.46.2.26	<code>nppiAverageError_32sc_C2R</code>	718
7.46.2.27	<code>nppiAverageError_32sc_C3R</code>	719
7.46.2.28	<code>nppiAverageError_32sc_C4R</code>	719
7.46.2.29	<code>nppiAverageError_32u_C1R</code>	720
7.46.2.30	<code>nppiAverageError_32u_C2R</code>	720
7.46.2.31	<code>nppiAverageError_32u_C3R</code>	720

7.46.2.32	<code>nppiAverageError_32u_C4R</code>	721
7.46.2.33	<code>nppiAverageError_64f_C1R</code>	721
7.46.2.34	<code>nppiAverageError_64f_C2R</code>	722
7.46.2.35	<code>nppiAverageError_64f_C3R</code>	722
7.46.2.36	<code>nppiAverageError_64f_C4R</code>	723
7.46.2.37	<code>nppiAverageError_8s_C1R</code>	723
7.46.2.38	<code>nppiAverageError_8s_C2R</code>	724
7.46.2.39	<code>nppiAverageError_8s_C3R</code>	724
7.46.2.40	<code>nppiAverageError_8s_C4R</code>	724
7.46.2.41	<code>nppiAverageError_8u_C1R</code>	725
7.46.2.42	<code>nppiAverageError_8u_C2R</code>	725
7.46.2.43	<code>nppiAverageError_8u_C3R</code>	726
7.46.2.44	<code>nppiAverageError_8u_C4R</code>	726
7.47	<code>MaximumRelativeError</code>	727
7.47.1	Detailed Description	730
7.47.2	Function Documentation	730
7.47.2.1	<code>nppiMaximumRelativeError_16s_C1R</code>	730
7.47.2.2	<code>nppiMaximumRelativeError_16s_C2R</code>	731
7.47.2.3	<code>nppiMaximumRelativeError_16s_C3R</code>	731
7.47.2.4	<code>nppiMaximumRelativeError_16s_C4R</code>	732
7.47.2.5	<code>nppiMaximumRelativeError_16sc_C1R</code>	732
7.47.2.6	<code>nppiMaximumRelativeError_16sc_C2R</code>	733
7.47.2.7	<code>nppiMaximumRelativeError_16sc_C3R</code>	733
7.47.2.8	<code>nppiMaximumRelativeError_16sc_C4R</code>	734
7.47.2.9	<code>nppiMaximumRelativeError_16u_C1R</code>	734
7.47.2.10	<code>nppiMaximumRelativeError_16u_C2R</code>	734
7.47.2.11	<code>nppiMaximumRelativeError_16u_C3R</code>	735
7.47.2.12	<code>nppiMaximumRelativeError_16u_C4R</code>	735
7.47.2.13	<code>nppiMaximumRelativeError_32f_C1R</code>	736
7.47.2.14	<code>nppiMaximumRelativeError_32f_C2R</code>	736
7.47.2.15	<code>nppiMaximumRelativeError_32f_C3R</code>	737
7.47.2.16	<code>nppiMaximumRelativeError_32f_C4R</code>	737
7.47.2.17	<code>nppiMaximumRelativeError_32fc_C1R</code>	738
7.47.2.18	<code>nppiMaximumRelativeError_32fc_C2R</code>	738
7.47.2.19	<code>nppiMaximumRelativeError_32fc_C3R</code>	739
7.47.2.20	<code>nppiMaximumRelativeError_32fc_C4R</code>	739

7.47.2.21	nppiMaximumRelativeError_32s_C1R	740
7.47.2.22	nppiMaximumRelativeError_32s_C2R	740
7.47.2.23	nppiMaximumRelativeError_32s_C3R	740
7.47.2.24	nppiMaximumRelativeError_32s_C4R	741
7.47.2.25	nppiMaximumRelativeError_32sc_C1R	741
7.47.2.26	nppiMaximumRelativeError_32sc_C2R	742
7.47.2.27	nppiMaximumRelativeError_32sc_C3R	742
7.47.2.28	nppiMaximumRelativeError_32sc_C4R	743
7.47.2.29	nppiMaximumRelativeError_32u_C1R	743
7.47.2.30	nppiMaximumRelativeError_32u_C2R	744
7.47.2.31	nppiMaximumRelativeError_32u_C3R	744
7.47.2.32	nppiMaximumRelativeError_32u_C4R	744
7.47.2.33	nppiMaximumRelativeError_64f_C1R	745
7.47.2.34	nppiMaximumRelativeError_64f_C2R	745
7.47.2.35	nppiMaximumRelativeError_64f_C3R	746
7.47.2.36	nppiMaximumRelativeError_64f_C4R	746
7.47.2.37	nppiMaximumRelativeError_8s_C1R	747
7.47.2.38	nppiMaximumRelativeError_8s_C2R	747
7.47.2.39	nppiMaximumRelativeError_8s_C3R	748
7.47.2.40	nppiMaximumRelativeError_8s_C4R	748
7.47.2.41	nppiMaximumRelativeError_8u_C1R	749
7.47.2.42	nppiMaximumRelativeError_8u_C2R	749
7.47.2.43	nppiMaximumRelativeError_8u_C3R	749
7.47.2.44	nppiMaximumRelativeError_8u_C4R	750
7.48	AverageRelativeError	751
7.48.1	Detailed Description	754
7.48.2	Function Documentation	754
7.48.2.1	nppiAverageRelativeError_16s_C1R	754
7.48.2.2	nppiAverageRelativeError_16s_C2R	755
7.48.2.3	nppiAverageRelativeError_16s_C3R	755
7.48.2.4	nppiAverageRelativeError_16s_C4R	756
7.48.2.5	nppiAverageRelativeError_16sc_C1R	756
7.48.2.6	nppiAverageRelativeError_16sc_C2R	757
7.48.2.7	nppiAverageRelativeError_16sc_C3R	757
7.48.2.8	nppiAverageRelativeError_16sc_C4R	758
7.48.2.9	nppiAverageRelativeError_16u_C1R	758

7.48.2.10	nppiAverageRelativeError_16u_C2R	758
7.48.2.11	nppiAverageRelativeError_16u_C3R	759
7.48.2.12	nppiAverageRelativeError_16u_C4R	759
7.48.2.13	nppiAverageRelativeError_32f_C1R	760
7.48.2.14	nppiAverageRelativeError_32f_C2R	760
7.48.2.15	nppiAverageRelativeError_32f_C3R	761
7.48.2.16	nppiAverageRelativeError_32f_C4R	761
7.48.2.17	nppiAverageRelativeError_32fc_C1R	762
7.48.2.18	nppiAverageRelativeError_32fc_C2R	762
7.48.2.19	nppiAverageRelativeError_32fc_C3R	763
7.48.2.20	nppiAverageRelativeError_32fc_C4R	763
7.48.2.21	nppiAverageRelativeError_32s_C1R	764
7.48.2.22	nppiAverageRelativeError_32s_C2R	764
7.48.2.23	nppiAverageRelativeError_32s_C3R	764
7.48.2.24	nppiAverageRelativeError_32s_C4R	765
7.48.2.25	nppiAverageRelativeError_32sc_C1R	765
7.48.2.26	nppiAverageRelativeError_32sc_C2R	766
7.48.2.27	nppiAverageRelativeError_32sc_C3R	766
7.48.2.28	nppiAverageRelativeError_32sc_C4R	767
7.48.2.29	nppiAverageRelativeError_32u_C1R	767
7.48.2.30	nppiAverageRelativeError_32u_C2R	768
7.48.2.31	nppiAverageRelativeError_32u_C3R	768
7.48.2.32	nppiAverageRelativeError_32u_C4R	768
7.48.2.33	nppiAverageRelativeError_64f_C1R	769
7.48.2.34	nppiAverageRelativeError_64f_C2R	769
7.48.2.35	nppiAverageRelativeError_64f_C3R	770
7.48.2.36	nppiAverageRelativeError_64f_C4R	770
7.48.2.37	nppiAverageRelativeError_8s_C1R	771
7.48.2.38	nppiAverageRelativeError_8s_C2R	771
7.48.2.39	nppiAverageRelativeError_8s_C3R	772
7.48.2.40	nppiAverageRelativeError_8s_C4R	772
7.48.2.41	nppiAverageRelativeError_8u_C1R	773
7.48.2.42	nppiAverageRelativeError_8u_C2R	773
7.48.2.43	nppiAverageRelativeError_8u_C3R	773
7.48.2.44	nppiAverageRelativeError_8u_C4R	774
7.49	Linear Transforms	775

7.49.1	Detailed Description	775
7.50	Fourier Transforms	776
7.50.1	Function Documentation	776
7.50.1.1	nppiMagnitude_32fc32f_C1R	776
7.50.1.2	nppiMagnitudeSqr_32fc32f_C1R	776
8	Data Structure Documentation	779
8.1	NPP_ALIGN_16 Struct Reference	779
8.1.1	Detailed Description	779
8.1.2	Field Documentation	779
8.1.2.1	im	779
8.1.2.2	im	780
8.1.2.3	re	780
8.1.2.4	re	780
8.2	NPP_ALIGN_8 Struct Reference	781
8.2.1	Detailed Description	781
8.2.2	Field Documentation	781
8.2.2.1	im	781
8.2.2.2	im	781
8.2.2.3	im	781
8.2.2.4	re	782
8.2.2.5	re	782
8.2.2.6	re	782
8.3	NppiHaarBuffer Struct Reference	783
8.3.1	Field Documentation	783
8.3.1.1	haarBuffer	783
8.3.1.2	haarBufferSize	783
8.4	NppiHaarClassifier_32f Struct Reference	784
8.4.1	Field Documentation	784
8.4.1.1	classifiers	784
8.4.1.2	classifierSize	784
8.4.1.3	classifierStep	784
8.4.1.4	counterDevice	784
8.4.1.5	numClassifiers	784
8.5	NppiPoint Struct Reference	785
8.5.1	Detailed Description	785
8.5.2	Field Documentation	785

8.5.2.1	x	785
8.5.2.2	y	785
8.6	NppiRect Struct Reference	786
8.6.1	Detailed Description	786
8.6.2	Field Documentation	786
8.6.2.1	height	786
8.6.2.2	width	786
8.6.2.3	x	786
8.6.2.4	y	786
8.7	NppiSize Struct Reference	787
8.7.1	Detailed Description	787
8.7.2	Field Documentation	787
8.7.2.1	height	787
8.7.2.2	width	787
8.8	NppLibraryVersion Struct Reference	788
8.8.1	Field Documentation	788
8.8.1.1	build	788
8.8.1.2	major	788
8.8.1.3	minor	788

Chapter 1

NVIDIA Performance Primitives

Note: Starting with release 6.5, NPP is also provided as a static library (`libnppc_static.a`, `libnppi_static.a`, and `libnpps_static.a`) on Linux, Android, and Mac OSes in addition to being provided as a shared library. The static NPP libraries depend on a common thread abstraction layer library called `cuLIBOS` (`libculibos.a`) that is now distributed as part of the toolkit. Consequently, `cuLIBOS` must be provided to the linker when the static library is being linked against. The `libnppi` library is becoming quite large so to minimize library loading and CUDA runtime startup times it is recommended to use the static library(s) whenever possible. To improve loading and runtime performance when using dynamic libraries NPP 8.0 now includes the full set of `nppi` sub-libraries in addition to the full sized `nppi` library itself. Linking to only the sub-libraries that contain functions that your application uses can significantly improve load time and runtime startup performance. Some `nppi` functions make calls to other `nppi` and/or `npps` functions internally so you may need to link to a few extra libraries depending on what function calls your application makes. The `nppi` sub-libraries are split into sections corresponding to the way that `nppi` header files are split. There are also static versions of each of the new sub-libraries. The full sized `nppi` library will be deprecated in the next CUDA release. This list of sub-libraries is as follows:

```
nppial arithmetic and logical operation functions in nppi_arithmetic_and_logical_operations.h
nppicc color conversion and sampling functions in nppi_color_conversion.h
nppicom JPEG compression and decompression functions in nppi_compression_functions.h
nppidei data exchange and initialization functions in nppi_data_exchange_and_initialization.h
nppif filtering and computer vision functions in nppi_filter_functions.h
nppig geometry transformation functions found in nppi_geometry_transforms.h
nppim morphological operation functions found in nppi_morphological_operations.h
nppist statistics and linear transform in nppi_statistics_functions.h and nppi_linear_transforms.h
nppisu memory support functions in nppi_support_functions.h
nppitc threshold and compare operation functions in nppi_threshold_and_compare_operations.h
```

For example, on Linux, to compile a small application `foo` using NPP against the dynamic library, the following command can be used:

```
nvcc foo.c -lnppi -o foo
```

Whereas to compile against the static NPP library, the following command has to be used:

```
nvcc foo.c -lnppi_static -lculibos -o foo
```

It is also possible to use the native host C++ compiler. Depending on the host operating system, some additional libraries like `pthread` or `dl` might be needed on the linking line. The following command on Linux is suggested:

```
g++ foo.c -lnppi_static -lculibos -lcudart_static -lpthread -ldl
-I <cuda-toolkit-path>/include -L <cuda-toolkit-path>/lib64 -o foo
```

NPP is a stateless API, as of NPP 6.5 the ONLY state that NPP remembers between function calls is the current stream ID, i.e. the stream ID that was set in the most recent `nppSetStream` call. The default stream ID is 0. If an application intends to use NPP with multiple streams then it is the responsibility of the application to call `nppSetStream` whenever it wishes to change stream IDs. Several NPP functions may call other NPP functions internally to complete their functionality. For this reason it is recommended that `cudaDeviceSynchronize` be called before making an `nppSetStream` call to change to a new stream ID. This will insure that any internal function calls that have not yet occurred will be completed using the current stream ID before it changes to a new ID. Calling `cudaDeviceSynchronize` frequently call kill performance so minimizing the frequency of these calls is critical for good performance. It is not necessary to call `cudaDeviceSynchronize` for stream management while the same stream ID is used for multiple NPP calls. All NPP functions should be thread safe except for the following functions:

```
nppiGraphcut_32s8u - this function has been deprecated in NPP 8.0
nppiGraphcut_32f8u - this function has been deprecated in NPP 8.0
nppiGraphcut8_32s8u - this function has been deprecated in NPP 8.0
nppiGraphcut8_32f8u - this function has been deprecated in NPP 8.0
nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R
nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R
```

As of NPP version 5.0 and beyond a few parameters for a few pre-5.0 existing image LUT functions have changed from host memory pointers to device memory pointers. Your application will fail (crash or report an error) if you use these functions with host memory pointers. The functions are the `nppiLUT_Linear_8u_xxx` functions.

Also, pre-5.0 function `nppiMeanStdDev8uC1RGetBufferHostSize` has been renamed `nppiMeanStdDevGetBufferHostSize_8u_C1R`.

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, 8, and 10 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Linux (Centos, Ubuntu, and several others) (64-bit and 32-bit)
- Mac OS X (64-bit)
- Android on Arm (32-bit and 64-bit)

1.4 Files

NPP is comprised of the following files:

1.4.1 Header Files

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

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

```
/include/
```

directory.

1.4.2 Library Files

Starting with Version 5.5 NPP's functionality is now split up into 3 distinct libraries:

- A core library (NPPC) containing basic functionality from the `npp.h` header files as well as functionality shared by the other two libraries.
- The image processing library NPPI. Any functions from the `nppi.h` header file (or the various header files named "nppi_XXX.h") are bundled into the NPPI library.

- The signal processing library NPPS. Any function from the npps.h header file (or the various header files named "npps_xxx.h" are bundled into the NPPS library.

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

```
/lib/nppc.lib
```

```
/lib/nppi.lib
```

```
/lib/npps.lib
```

The matching DLLs are located in the CUDA Toolkit's binary directory. Example

```
/bin/nppi64_55_<build_no>.dll // Dynamic image-processing library for 64-bit Windows.
```

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

```
/lib/libnppc32.so.5.5.<build_no> // NPP 32-bit dynamic core library for Linux
```

```
/lib/libnpps32.5.5.dylib // NPP 32-bit dynamic signal processing 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

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 signal sum primitive and allocate and free the necessary scratch memory:

```
// pSrc, pSum, pDeviceBuffer are all device pointers.
Npp32f * pSrc;
Npp32f * pSum;
Npp8u * pDeviceBuffer;
int nLength = 1024;

// Allocate the device memroy.
cudaMalloc((void **)(&pSrc), sizeof(Npp32f) * nLength);
nppsSet_32f(1.0f, pSrc, nLength);
cudaMalloc((void **)(&pSum), sizeof(Npp32f) * 1);

// Compute the appropriate size of the scratch-memory buffer
int nBufferSize;
nppsSumGetBufferSize_32f(nLength, &nBufferSize);
// Allocate the scratch buffer
cudaMalloc((void **)(&pDeviceBuffer), nBufferSize);

// Call the primitive with the scratch buffer
```

```
nppsSum_32f(pSrc, nLength, pSum, pDeviceBuffer);
Npp32f nSumHost;
cudaMemcpy(&nSumHost, pSum, sizeof(Npp32f) * 1, cudaMemcpyDeviceToHost);
printf("sum = %f\n", nSumHost); // nSumHost = 1024.0f;

// Free the device memory
cudaFree(pSrc);
cudaFree(pDeviceBuffer);
cudaFree(pSum);
```

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 usually a fixed point fractional representation of some physical magnitude (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend to 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

$$128^2 * 2^{-8} = 64$$

2.4 Rounding Modes

Many NPP functions require converting floating-point values to integers. The [NppRoundMode](#) enum lists NPP's supported rounding modes. Not all primitives in NPP that perform rounding as part of their functionality allow the user to specify the round-mode used. Instead they use NPP's default rounding mode, which is [NPP_RND_FINANCIAL](#).

2.4.1 Rounding Mode Parameter

A subset of NPP functions performing rounding as part of their functionality do allow the user to specify which rounding mode is used through a parameter of the [NppRoundMode](#) type.

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-Planar-Image Pointer Array

The planar source image data is generally passed via an array of pointers named

```
pSrc[]
```

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

```
nppiPrimitive_8u_P3R(const Npp8u * const pSrc[3], ...)
```

Each pointer in the array points to a different image plane.

4.2.2.1.3 Source-Planar-Image Pointer

The multiple plane source image data is passed via a set of pointers named

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

The planar source image pointer is generally defined as one of a set of constant pointers with each pointer pointing to a different input image plane.

4.2.2.1.4 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, ...
```

4.2.2.1.5 Source-Planar-Image Line Step Array

The source planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the input image. The source planar image line step array parameter is

```
rSrcStep[]
```

4.2.2.1.6 Source-Planar-Image Line Step

The source planar image line step is the number of bytes between successive rows in a particular plane of the multiplane input image. The source planar image line step parameter is

```
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 generates multiple images as outputs the destination pointers are numbered like this:

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

4.2.2.2.2 Destination-Planar-Image Pointer Array

The planar destination image data pointers are generally passed via an array of pointers named

```
pDst[]
```

Each pointer in the array points to a different image plane.

4.2.2.2.3 Destination-Planar-Image Pointer

The destination planar image data is generally passed via a pointer to each plane of a multiplane output image named

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

4.2.2.2.4 Destination-Image Line Step

The destination image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

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

4.2.2.2.5 Destination-Planar-Image Line Step Array

The destination planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the output image. The destination planar image line step array parameter is

```
rDstStep[]
```

4.2.2.2.6 Destination-Planar-Image Line Step

The destination planar image line step is the number of bytes between successive rows for a particular plane in a multiplane output image. The destination planar image line step parameter is

```
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
```

4.2.2.5 Passing Channel-of-Interest Data

Some image processing primitives support [Channel-of-Interest API](#).

4.2.2.5.1 Channel_of_Interest Number

The channel-of-interest data is generally an integer (either 1, 2, or 3):

```
nCOI
```

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 NPPI'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. In most cases the ROI is passed as a single `NppiSize` struct, which provides the width and height of the ROI. This raises the 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-left corner (lowest memory address), the user simply offsets the image-data pointers to point to the first pixel of the ROI.

In practice this means that for an image (`pSrc`, `nSrcStep`) and the start-pixel of the ROI being at location (`x`, `y`), one would pass

```
pSrcOffset = pSrc + y * nSrcStep + x * 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. Some primitives also explicitly specify the selected channel number and pass it via an integer, e.g. `nppiMean_StdDev_8u_C3CR(...)`.

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 one:

```
nppiCopy_8u_C3CR(pSrc + 1, nSrcStep, pDst, nDstStep, oSizeROI);
```

4.5.2 Select-Channel Source-Image

Some primitives allow the user to select the channel-of-interest by specifying the channel number (`nCOI`). This approach is typically used in the image statistical functions. For example,

```
nppiMean_StdDev_8u_C3CR(pSrc, nSrcStep, oSizeROI, nCOI, pDeviceBuffer, pMean, pStdDev );
```

The channel-of-interest number can be either 1, 2, or 3.

4.5.3 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 one:

```
nppiCopy_8u_C3CR(pSrc, nSrcStep, pDst + 1, nDstStep, oSizeROI);
```

4.6 Source-Image Sampling

A large number of NPP image-processing functions consume at least one source image and produce an output image (e.g. `nppiAddC_8u_C1RSfs()` or `nppiFilterBox_8u_C1R()`). All NPP functions falling into this category also operate on ROIs (see [Region-of-Interest \(ROI\)](#)) which for these functions should be considered to describe the destination ROI. In other words the ROI describes a rectangular region in the destination image and all pixels inside of this region are being written by the function in question.

In order to use such functions successfully it is important to understand how the user defined destination ROI affects which pixels in the input image(s) are being read by the algorithms. To simplify the discussion of ROI propagation (i.e. given a destination ROI, what are the ROIs in the source(s)), it makes sense to distinguish two major cases:

1. Point-Wise Operations: These are primitives like `nppiAddC_8u_C1RSfs()`. Each output pixel requires exactly one input pixel to be read.
2. Neighborhood Operations: These are primitives like `nppiFilterBox_8u_C1R()`, which require a group of pixels from the source image(s) to be read in order to produce a single output.

4.6.1 Point-Wise Operations

As mentioned above, point-wise operations consume a single pixel from the input image (or a single pixel from each input image, if the operation in question has more than one input image) in order to produce a single output pixel.

4.6.2 Neighborhood Operations

In the case of neighborhood operations a number of input pixels (a "neighborhood" of pixels) is read in the input image (or images) in order to compute a single output pixel. All of the functions for `image_filtering_functions` and `image_morphological_operations` are neighborhood operations.

Most of these functions have parameters that affect the size and relative location of the neighborhood: a mask-size structure and an anchor-point structure. Both parameters are described in more detail in the next subsections.

4.6.2.1 Mask-Size Parameter

Many NPP neighborhood operations allow the user to specify the size of the neighborhood via a parameter usually named `oMaskSize` of type `NppiSize`. In those cases the neighborhood of pixels read from the source(s) is exactly the size of the mask. Assuming the mask is anchored at location (0, 0) (see [Anchor-Point Parameter](#) below) and has a size of (w, h), i.e.

```
assert(oMaskSize.w == w);  
assert(oMaskSize.h == h);  
assert(oAnchor.x == 0);  
assert(oAnchor.y == 0);
```

a neighborhood operation would read the following source pixels in order to compute destination pixel $D_{i,j}$:

$$\begin{array}{cccc} S_{i,j} & S_{i,j+1} & \cdots & S_{i,j+w-1} \\ S_{i+1,j} & S_{i+1,j+1} & \cdots & S_{i+1,j+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i+h-1,j} & S_{i+h-1,j+1} & \cdots & S_{i+h-1,j+w-1} \end{array}$$

4.6.2.2 Anchor-Point Parameter

Many NPP primitives performing neighborhood operations allow the user to specify the relative location of the neighborhood via a parameter usually named `oAnchor` of type [NppiPoint](#). Using the anchor a developer can choose the position of the mask (see [Mask-Size Parameter](#)) relative to current pixel index.

Using the same example as in [Mask-Size Parameter](#), but this time with an anchor position of (a, b):

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == a);
assert(oAnchor.y == b);
```

the following pixels from the source image would be read:

$$\begin{array}{cccc} S_{i-a,j-b} & S_{i-a,j-b+1} & \cdots & S_{i-a,j-b+w-1} \\ S_{i-a+1,j-b} & S_{i-a+1,j-b+1} & \cdots & S_{i-a+1,j-b+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i-a+h-1,j-b} & S_{i-a+h-1,j-b+1} & \cdots & S_{i-a+h-1,j-b+w-1} \end{array}$$

4.6.2.3 Sampling Beyond Image Boundaries

NPP primitives in general and NPP neighborhood operations in particular require that all pixel locations read and written are valid and within the boundaries of the respective images. Sampling outside of the defined image data regions results in undefined behavior and may lead to system instability.

This poses a problem in practice: when processing full-size images one cannot choose the destination ROI to be the same size as the source image. Because neighborhood operations read pixels from an enlarged source ROI, the destination ROI must be shrunk so that the expanded source ROI does not exceed the source image's size.

For cases where this "shrinking" of the destination image size is unacceptable, NPP provides a set of border-expanding Copy primitives. E.g. `nppiCopyConstBorder_8u_C1R()`, `nppiCopyReplicateBorder_8u_C1R()` and `nppiCopyWrapBorder_8u_C1R()`. The user can use these primitives to "expand" the source image's size using one of the three expansion modes. The expanded image can then be safely passed to a neighborhood operation producing a full-size result.

Chapter 5

Module Index

5.1 Modules

Here is a list of all modules:

NPP Core	27
NPP Type Definitions and Constants	31
Basic NPP Data Types	46
Statistical Operations	50
Sum	117
Min	132
MinIndx	145
Max	159
MaxIndx	172
MinMax	186
MinMaxIndx	200
Mean	217
Mean_StdDev	238
Image Norms	254
Norm_Inf	256
Norm_L1	278
Norm_L2	299
NormDiff_Inf	320
NormDiff_L1	343
NormDiff_L2	366
NormRel_Inf	389
NormRel_L1	412
NormRel_L2	435
DotProd	458
CountInRange.	483
MaxEvery	489
MinEvery	496
Integral	503
SqrIntegral	505
RectStdDev	508
HistogramEven	511
HistogramRange	524
Image Proximity	540

SqrDistanceFull_Norm	543
SqrDistanceSame_Norm	554
SqrDistanceValid_Norm	565
CrossCorrFull_Norm	576
CrossCorrSame_Norm	587
CrossCorrValid_Norm	598
CrossCorrValid	609
CrossCorrFull_NormLevel	612
CrossCorrSame_NormLevel	632
CrossCorrValid_NormLevel	652
Image Quality Index	672
MaximumError	681
AverageError	704
MaximumRelativeError	727
AverageRelativeError	751
Linear Transforms	775
Fourier Transforms	776

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

NPP_ALIGN_16 (Complex Number This struct represents a long long complex number)	779
NPP_ALIGN_8 (Complex Number This struct represents an unsigned int complex number) . .	781
NppiHaarBuffer	783
NppiHaarClassifier_32f	784
NppiPoint (2D Point)	785
NppiRect (2D Rectangle This struct contains position and size information of a rectangle in two space)	786
NppiSize (2D Size This struct typically represents the size of a a rectangular region in two space)	787
NppLibraryVersion	788

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 active CUDA device?
- int [nppGetGpuNumSMs](#) (void)
Get the number of Streaming Multiprocessors (SM) on the active CUDA device.
- int [nppGetMaxThreadsPerBlock](#) (void)
Get the maximum number of threads per block on the active CUDA device.
- int [nppGetMaxThreadsPerSM](#) (void)
Get the maximum number of threads per SM for the active GPU.
- int [nppGetGpuDeviceProperties](#) (int *pMaxThreadsPerSM, int *pMaxThreadsPerBlock, int *pNumberOfSMs)
Get the maximum number of threads per SM, maximum threads per block, and number of SMs for the active GPU.
- const char * [nppGetGpuName](#) (void)
Get the name of the active CUDA device.
- cudaStream_t [nppGetStream](#) (void)
Get the NPP CUDA stream.
- unsigned int [nppGetStreamNumSMs](#) (void)
Get the number of SMs on the device associated with the current NPP CUDA stream.

- unsigned int `nppGetStreamMaxThreadsPerSM` (void)
Get the maximum number of threads per SM on the device associated with the current 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 active 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 int nppGetGpuDeviceProperties (int * pMaxThreadsPerSM, int * pMaxThreadsPerBlock, int * pNumberOfSMs)

Get the maximum number of threads per SM, maximum threads per block, and number of SMs for the active GPU.

Returns:

cudaSuccess for success, -1 for failure

7.1.2.3 const char* nppGetGpuName (void)

Get the name of the active CUDA device.

Returns:

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

7.1.2.4 int nppGetGpuNumSMs (void)

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

Returns:

Number of SMs of the default CUDA device.

7.1.2.5 const NppLibraryVersion* nppGetLibVersion (void)

Get the NPP library version.

Returns:

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

7.1.2.6 int nppGetMaxThreadsPerBlock (void)

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

Returns:

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

7.1.2.7 int nppGetMaxThreadsPerSM (void)

Get the maximum number of threads per SM for the active GPU.

Returns:

Maximum number of threads per SM for the active GPU

7.1.2.8 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.

7.1.2.9 unsigned int nppGetStreamMaxThreadsPerSM (void)

Get the maximum number of threads per SM on the device associated with the current 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. This call avoids a cudaGetDeviceProperties() call.

7.1.2.10 unsigned int nppGetStreamNumSMs (void)

Get the number of SMs on the device associated with the current 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. This call avoids a cudaGetDeviceProperties() call.

7.1.2.11 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](#) (4294967295U)
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](#) (-127 - 1)
Minimum 8-bit signed integer.

- #define `NPP_MAX_8S` (127)
Maximum 8-bit signed integer.
- #define `NPP_MIN_16S` (-32767 - 1)
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)
Maximum 64-bit signed integer.
- #define `NPP_MIN_64S` (-9223372036854775807LL - 1)
Minimum 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_CUBIC2P_BSPLINE`,
`NPPI_INTER_CUBIC2P_CATMULLROM`,
`NPPI_INTER_CUBIC2P_B05C03`,
`NPPI_INTER_SUPER` = 8,
`NPPI_INTER_LANCZOS` = 16,
`NPPI_INTER_LANCZOS3_ADVANCED` = 17,
`NPPI_SMOOTH_EDGE` = (1 << 31) }

Filtering methods.

- enum `NppiBayerGridPosition` {
 `NPPI_BAYER_BGGR` = 0,
 `NPPI_BAYER_RGGB` = 1,
 `NPPI_BAYER_GBRG` = 2,
 `NPPI_BAYER_GRBG` = 3 }

Bayer Grid Position Registration.

- enum `NppiMaskSize` {
 `NPP_MASK_SIZE_1_X_3`,
 `NPP_MASK_SIZE_1_X_5`,
 `NPP_MASK_SIZE_3_X_1` = 100,
 `NPP_MASK_SIZE_5_X_1`,
 `NPP_MASK_SIZE_3_X_3` = 200,
 `NPP_MASK_SIZE_5_X_5`,
 `NPP_MASK_SIZE_7_X_7` = 400,
 `NPP_MASK_SIZE_9_X_9` = 500,
 `NPP_MASK_SIZE_11_X_11` = 600,
 `NPP_MASK_SIZE_13_X_13` = 700,
 `NPP_MASK_SIZE_15_X_15` = 800 }

Fixed filter-kernel sizes.

- enum `NppiDifferentialKernel` {
 `NPP_FILTER_SOBEL`,
 `NPP_FILTER_SCHARR` }

Differential Filter types.

- enum `NppStatus` {
 `NPP_NOT_SUPPORTED_MODE_ERROR` = -9999,
 `NPP_INVALID_HOST_POINTER_ERROR` = -1032,
 `NPP_INVALID_DEVICE_POINTER_ERROR` = -1031,
 `NPP_LUT_PALETTE_BITSIZE_ERROR` = -1030,
 `NPP_ZC_MODE_NOT_SUPPORTED_ERROR` = -1028,
 `NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY` = -1027,
 `NPP_TEXTURE_BIND_ERROR` = -1024,
 `NPP_WRONG_INTERSECTION_ROI_ERROR` = -1020,
 `NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR` = -1006,
 `NPP_MEMFREE_ERROR` = -1005,
 `NPP_MEMSET_ERROR` = -1004,
 `NPP_MEMCPY_ERROR` = -1003,
 `NPP_ALIGNMENT_ERROR` = -1002,
 `NPP_CUDA_KERNEL_EXECUTION_ERROR` = -1000,

NPP_ROUND_MODE_NOT_SUPPORTED_ERROR = -213,
NPP_QUALITY_INDEX_ERROR = -210,
NPP_RESIZE_NO_OPERATION_ERROR = -201,
NPP_OVERFLOW_ERROR = -109,
NPP_NOT_EVEN_STEP_ERROR = -108,
NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR = -107,
NPP_LUT_NUMBER_OF_LEVELS_ERROR = -106,
NPP_CORRUPTED_DATA_ERROR = -61,
NPP_CHANNEL_ORDER_ERROR = -60,
NPP_ZERO_MASK_VALUE_ERROR = -59,
NPP_QUADRANGLE_ERROR = -58,
NPP_RECTANGLE_ERROR = -57,
NPP_COEFFICIENT_ERROR = -56,
NPP_NUMBER_OF_CHANNELS_ERROR = -53,
NPP_COI_ERROR = -52,
NPP_DIVISOR_ERROR = -51,
NPP_CHANNEL_ERROR = -47,
NPP_STRIDE_ERROR = -37,
NPP_ANCHOR_ERROR = -34,
NPP_MASK_SIZE_ERROR = -33,
NPP_RESIZE_FACTOR_ERROR = -23,
NPP_INTERPOLATION_ERROR = -22,
NPP_MIRROR_FLIP_ERROR = -21,
NPP_MOMENT_00_ZERO_ERROR = -20,
NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR = -19,
NPP_THRESHOLD_ERROR = -18,
NPP_CONTEXT_MATCH_ERROR = -17,
NPP_FFT_FLAG_ERROR = -16,
NPP_FFT_ORDER_ERROR = -15,
NPP_STEP_ERROR = -14,
NPP_SCALE_RANGE_ERROR = -13,
NPP_DATA_TYPE_ERROR = -12,
NPP_OUT_OFF_RANGE_ERROR = -11,
NPP_DIVIDE_BY_ZERO_ERROR = -10,
NPP_MEMORY_ALLOCATION_ERR = -9,
NPP_NULL_POINTER_ERROR = -8,
NPP_RANGE_ERROR = -7,
NPP_SIZE_ERROR = -6,
NPP_BAD_ARGUMENT_ERROR = -5,
NPP_NO_MEMORY_ERROR = -4,
NPP_NOT_IMPLEMENTED_ERROR = -3,


```
NPP_ERROR = -2,  
NPP_ERROR_RESERVED = -1,  
NPP_NO_ERROR = 0,  
NPP_SUCCESS = NPP_NO_ERROR,  
NPP_NO_OPERATION_WARNING = 1,  
NPP_DIVIDE_BY_ZERO_WARNING = 6,  
NPP_AFFINE_QUAD_INCORRECT_WARNING = 28,  
NPP_WRONG_INTERSECTION_ROI_WARNING = 29,  
NPP_WRONG_INTERSECTION_QUAD_WARNING = 30,  
NPP_DOUBLE_SIZE_WARNING = 35,  
NPP_MISALIGNED_DST_ROI_WARNING = 10000 }
```

Error Status Codes.

- enum NppGpuComputeCapability {
NPP_CUDA_UNKNOWN_VERSION = -1,
NPP_CUDA_NOT_CAPABLE = 0,
NPP_CUDA_1_0 = 100,
NPP_CUDA_1_1 = 110,
NPP_CUDA_1_2 = 120,
NPP_CUDA_1_3 = 130,
NPP_CUDA_2_0 = 200,
NPP_CUDA_2_1 = 210,
NPP_CUDA_3_0 = 300,
NPP_CUDA_3_2 = 320,
NPP_CUDA_3_5 = 350,
NPP_CUDA_3_7 = 370,
NPP_CUDA_5_0 = 500,
NPP_CUDA_5_2 = 520,
NPP_CUDA_5_3 = 530,
NPP_CUDA_6_0 = 600 }
- 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_NEAR`,
 - `NPP_ROUND_NEAREST_TIES_TO_EVEN` = `NPP_RND_NEAR`,
 - `NPP_RND_FINANCIAL`,
 - `NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO` = `NPP_RND_FINANCIAL`,
 - `NPP_RND_ZERO`,
 - `NPP_ROUND_TOWARD_ZERO` = `NPP_RND_ZERO` }

Rounding Modes.
- enum `NppiBorderType` {
 - `NPP_BORDER_UNDEFINED` = 0,
 - `NPP_BORDER_NONE` = `NPP_BORDER_UNDEFINED`,
 - `NPP_BORDER_CONSTANT` = 1,
 - `NPP_BORDER_REPLICATE` = 2,
 - `NPP_BORDER_WRAP` = 3,
 - `NPP_BORDER_MIRROR` = 4 }
- enum `NppHintAlgorithm` {
 - `NPP_ALG_HINT_NONE`,
 - `NPP_ALG_HINT_FAST`,
 - `NPP_ALG_HINT_ACCURATE` }
- 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` }
- enum `NppsZCType` {
 - `nppZCR`,
 - `nppZCXor`,
 - `nppZCC` }
- enum `NppiHuffmanTableType` {
 - `nppiDCTable`,
 - `nppiACTable` }

- enum `NppiNorm` {
 `nppiNormInf` = 0,
 `nppiNormL1` = 1,
 `nppiNormL2` = 2 }

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` (4294967295U)

Maximum 32-bit unsigned integer.

7.2.1.5 #define `NPP_MAX_64S` (9223372036854775807LL)

Maximum 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 (-32767 - 1)

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)

Minimum 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 (-127 - 1)

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.
NPP_CUDA_1_0 Indicates that CUDA 1.0 capable device is machine's default device.
NPP_CUDA_1_1 Indicates that CUDA 1.1 capable device is machine's default device.
NPP_CUDA_1_2 Indicates that CUDA 1.2 capable device is machine's default device.
NPP_CUDA_1_3 Indicates that CUDA 1.3 capable device is machine's default device.
NPP_CUDA_2_0 Indicates that CUDA 2.0 capable device is machine's default device.
NPP_CUDA_2_1 Indicates that CUDA 2.1 capable device is machine's default device.
NPP_CUDA_3_0 Indicates that CUDA 3.0 capable device is machine's default device.
NPP_CUDA_3_2 Indicates that CUDA 3.2 capable device is machine's default device.
NPP_CUDA_3_5 Indicates that CUDA 3.5 capable device is machine's default device.
NPP_CUDA_3_7 Indicates that CUDA 3.7 capable device is machine's default device.
NPP_CUDA_5_0 Indicates that CUDA 5.0 capable device is machine's default device.
NPP_CUDA_5_2 Indicates that CUDA 5.2 capable device is machine's default device.
NPP_CUDA_5_3 Indicates that CUDA 5.3 capable device is machine's default device.
NPP_CUDA_6_0 Indicates that CUDA 6.0 or better is machine's default device.

7.2.2.3 enum NppHintAlgorithm

Enumerator:

NPP_ALG_HINT_NONE
NPP_ALG_HINT_FAST
NPP_ALG_HINT_ACCURATE

7.2.2.4 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

7.2.2.5 enum NppiAxis

Enumerator:

NPP_HORIZONTAL_AXIS
NPP_VERTICAL_AXIS
NPP_BOTH_AXIS

7.2.2.6 enum NppiBayerGridPosition

Bayer Grid Position Registration.

Enumerator:

NPPI_BAYER_BGGR Default registration position.
NPPI_BAYER_RGGB
NPPI_BAYER_GBRG
NPPI_BAYER_GRBG

7.2.2.7 enum NppiBorderType

Enumerator:

NPP_BORDER_UNDEFINED
NPP_BORDER_NONE
NPP_BORDER_CONSTANT
NPP_BORDER_REPLICATE
NPP_BORDER_WRAP
NPP_BORDER_MIRROR

7.2.2.8 enum NppiDifferentialKernel

Differential Filter types.

Enumerator:

NPP_FILTER_SOBEL
NPP_FILTER_SCHARR

7.2.2.9 enum NppiHuffmanTableType

Enumerator:

nppiDCTable DC Table.

nppiACTable AC Table.

7.2.2.10 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_CUBIC2P_BSPLINE Two-parameter cubic filter (B=1, C=0).

NPPI_INTER_CUBIC2P_CATMULLROM Two-parameter cubic filter (B=0, C=1/2).

NPPI_INTER_CUBIC2P_B05C03 Two-parameter cubic filter (B=1/2, C=3/10).

NPPI_INTER_SUPER Super sampling.

NPPI_INTER_LANCZOS Lanczos filtering.

NPPI_INTER_LANCZOS3_ADVANCED Generic Lanczos filtering with order 3.

NPPI_SMOOTH_EDGE Smooth edge filtering.

7.2.2.11 enum NppiMaskSize

Fixed filter-kernel sizes.

Enumerator:

NPP_MASK_SIZE_1_X_3

NPP_MASK_SIZE_1_X_5

NPP_MASK_SIZE_3_X_1

NPP_MASK_SIZE_5_X_1

NPP_MASK_SIZE_3_X_3

NPP_MASK_SIZE_5_X_5

NPP_MASK_SIZE_7_X_7

NPP_MASK_SIZE_9_X_9

NPP_MASK_SIZE_11_X_11

NPP_MASK_SIZE_13_X_13

NPP_MASK_SIZE_15_X_15

7.2.2.12 enum NppiNorm

Enumerator:

- nppiNormInf* maximum
- nppiNormL1* sum
- nppiNormL2* square root of sum of squares

7.2.2.13 enum NppRoundMode

Rounding Modes.

The enumerated rounding modes are used by a large number of NPP primitives to allow the user to specify the method by which fractional values are converted to integer values. Also see [Rounding Modes](#).

For NPP release 5.5 new names for the three rounding modes are introduced that are based on the naming conventions for rounding modes set forth in the IEEE-754 floating-point standard. Developers are encouraged to use the new, longer names to be future proof as the legacy names will be deprecated in subsequent NPP releases.

Enumerator:

NPP_RND_NEAR Round to the nearest even integer.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. $\langle \text{integer} \rangle.5$) are rounded to the closest even integer. E.g.

- `roundNear(0.5) = 0`
- `roundNear(0.6) = 1`
- `roundNear(1.5) = 2`
- `roundNear(-1.5) = -2`

NPP_ROUND_NEAREST_TIES_TO_EVEN Alias name for [NPP_RND_NEAR](#).

NPP_RND_FINANCIAL Round according to financial rule.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. $\langle \text{integer} \rangle.5$) are rounded away from zero. E.g.

- `roundFinancial(0.4) = 0`
- `roundFinancial(0.5) = 1`
- `roundFinancial(-1.5) = -2`

NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO Alias name for [NPP_RND_FINANCIAL](#).

NPP_RND_ZERO Round towards zero (truncation).

All fractional numbers of the form $\langle \text{integer} \rangle.\langle \text{decimals} \rangle$ are truncated to $\langle \text{integer} \rangle$.

- `roundZero(1.5) = 1`
- `roundZero(1.9) = 1`
- `roundZero(-2.5) = -2`

NPP_ROUND_TOWARD_ZERO Alias name for [NPP_RND_ZERO](#).

7.2.2.14 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_INVALID_HOST_POINTER_ERROR

NPP_INVALID_DEVICE_POINTER_ERROR

NPP_LUT_PALETTE_BITSIZE_ERROR

NPP_ZC_MODE_NOT_SUPPORTED_ERROR ZeroCrossing mode not supported.

NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY

NPP_TEXTURE_BIND_ERROR

NPP_WRONG_INTERSECTION_ROI_ERROR

NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR

NPP_MEMFREE_ERROR

NPP_MEMSET_ERROR

NPP_MEMCPY_ERROR

NPP_ALIGNMENT_ERROR

NPP_CUDA_KERNEL_EXECUTION_ERROR

NPP_ROUND_MODE_NOT_SUPPORTED_ERROR Unsupported round mode.

NPP_QUALITY_INDEX_ERROR Image pixels are constant for quality index.

NPP_RESIZE_NO_OPERATION_ERROR One of the output image dimensions is less than 1 pixel.

NPP_OVERFLOW_ERROR Number overflows the upper or lower limit of the data type.

NPP_NOT_EVEN_STEP_ERROR Step value is not pixel multiple.

NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR Number of levels for histogram is less than 2.

NPP_LUT_NUMBER_OF_LEVELS_ERROR Number of levels for LUT is less than 2.

NPP_CORRUPTED_DATA_ERROR Processed data is corrupted.

NPP_CHANNEL_ORDER_ERROR Wrong order of the destination channels.

NPP_ZERO_MASK_VALUE_ERROR All values of the mask are zero.

NPP_QUADRANGLE_ERROR The quadrangle is nonconvex or degenerates into triangle, line or point.

NPP_RECTANGLE_ERROR Size of the rectangle region is less than or equal to 1.

NPP_COEFFICIENT_ERROR Unallowable values of the transformation coefficients.

NPP_NUMBER_OF_CHANNELS_ERROR Bad or unsupported number of channels.

NPP_COI_ERROR Channel of interest is not 1, 2, or 3.

NPP_DIVISOR_ERROR Divisor is equal to zero.

NPP_CHANNEL_ERROR Illegal channel index.

NPP_STRIDE_ERROR Stride is less than the row length.

NPP_ANCHOR_ERROR Anchor point is outside mask.

NPP_MASK_SIZE_ERROR Lower bound is larger than upper bound.

NPP_RESIZE_FACTOR_ERROR

NPP_INTERPOLATION_ERROR

NPP_MIRROR_FLIP_ERROR

NPP_MOMENT_00_ZERO_ERROR

NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR

NPP_THRESHOLD_ERROR

NPP_CONTEXT_MATCH_ERROR

NPP_FFT_FLAG_ERROR

NPP_FFT_ORDER_ERROR

NPP_STEP_ERROR Step is less or equal zero.

NPP_SCALE_RANGE_ERROR

NPP_DATA_TYPE_ERROR

NPP_OUT_OFF_RANGE_ERROR

NPP_DIVIDE_BY_ZERO_ERROR

NPP_MEMORY_ALLOCATION_ERR

NPP_NULL_POINTER_ERROR

NPP_RANGE_ERROR

NPP_SIZE_ERROR

NPP_BAD_ARGUMENT_ERROR

NPP_NO_MEMORY_ERROR

NPP_NOT_IMPLEMENTED_ERROR

NPP_ERROR

NPP_ERROR_RESERVED

NPP_NO_ERROR Error free operation.

NPP_SUCCESS Successful operation (same as ***NPP_NO_ERROR***).

NPP_NO_OPERATION_WARNING Indicates that no operation was performed.

NPP_DIVIDE_BY_ZERO_WARNING Divisor is zero however does not terminate the execution.

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_WRONG_INTERSECTION_ROI_WARNING The given ROI has no intersection with either the source or destination ROI.
Thus no operation was performed.

NPP_WRONG_INTERSECTION_QUAD_WARNING The given quadrangle has no intersection with either the source or destination ROI.
Thus no operation was performed.

NPP_DOUBLE_SIZE_WARNING Image size isn't multiple of two.
Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

NPP_MISALIGNED_DST_ROI_WARNING Speed reduction due to uncoalesced memory accesses warning.

7.2.2.15 enum NppsZCType**Enumerator:**

nppZCR sign change
nppZCXor sign change XOR
nppZCC sign change count_0

7.3 Basic NPP Data Types

Data Structures

- struct [NPP_ALIGN_8](#)
Complex Number This struct represents an unsigned int complex number.
- struct [NPP_ALIGN_16](#)
Complex Number This struct represents a long long 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
- typedef struct [NPP_ALIGN_8](#) [Npp32uc](#)
Complex Number This struct represents an unsigned int complex number.
- typedef struct [NPP_ALIGN_8](#) [Npp32sc](#)
Complex Number This struct represents a signed int complex number.

- typedef struct [NPP_ALIGN_8 Npp32fc](#)
Complex Number This struct represents a single floating-point complex number.
- typedef struct [NPP_ALIGN_16 Npp64sc](#)
Complex Number This struct represents a long long complex number.
- typedef struct [NPP_ALIGN_16 Npp64fc](#)
Complex Number This struct represents a double floating-point complex number.

Functions

- struct [__align__](#) (2)
Complex Number This struct represents an unsigned char complex number.
- struct [__align__](#) (4)
Complex Number This struct represents an unsigned short complex number.

Variables

- [Npp8uc](#)
- [Npp16uc](#)
- [Npp16sc](#)

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 struct NPP_ALIGN_8 Npp32fc

Complex Number This struct represents a single floating-point complex number.

7.3.1.5 typedef int Npp32s

32-bit signed integers

7.3.1.6 typedef struct NPP_ALIGN_8 Npp32sc

Complex Number This struct represents a signed int complex number.

7.3.1.7 typedef unsigned int Npp32u

32-bit unsigned integers

7.3.1.8 typedef struct NPP_ALIGN_8 Npp32uc

Complex Number This struct represents an unsigned int complex number.

7.3.1.9 typedef double Npp64f

64-bit floating-point numbers

7.3.1.10 typedef struct NPP_ALIGN_16 Npp64fc

Complex Number This struct represents a double floating-point complex number.

7.3.1.11 typedef long long Npp64s

64-bit signed integers

7.3.1.12 typedef struct NPP_ALIGN_16 Npp64sc

Complex Number This struct represents a long long complex number.

7.3.1.13 typedef unsigned long long Npp64u

64-bit unsigned integers

7.3.1.14 typedef signed char Npp8s

8-bit signed chars

7.3.1.15 typedef unsigned char Npp8u

8-bit unsigned chars

7.3.2 Function Documentation**7.3.2.1 struct __align__ (4) [read]**

Complex Number This struct represents an unsigned short complex number.

Complex Number This struct represents a short complex number.

< Real part

< Imaginary part

< Real part

< Imaginary part

7.3.2.2 `struct __align__(2)` [read]

Complex Number This struct represents an unsigned char complex number.

< Real part

< Imaginary part

7.3.3 Variable Documentation

7.3.3.1 `Npp16sc`

7.3.3.2 `Npp16uc`

7.3.3.3 `Npp8uc`

7.4 Statistical Operations

Primitives for computing the statistical properties of an image.

Modules

- [Sum](#)
Primitives for computing the sum of all the pixel values in an image.
- [Min](#)
Primitives for computing the minimal pixel value of an image.
- [MinIndx](#)
Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.
- [Max](#)
Primitives for computing the maximal pixel value of an image.
- [MaxIndx](#)
Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.
- [MinMax](#)
Primitives for computing both the minimal and the maximal values of an image.
- [MinMaxIndx](#)
Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.
- [Mean](#)
Primitives for computing the arithmetic mean of all the pixel values in an image.
- [Mean_StdDev](#)
Primitives for computing both the arithmetic mean and the standard deviation of an image.
- [Image Norms](#)
Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.
- [DotProd](#)
Primitives for computing the dot product of two images.
- [CountInRange.](#)
Primitives for computing the amount of pixels that fall into the specified intensity range.
- [MaxEvery](#)
Primitives for computing the maximal value of the pixel pair from two images.
- [MinEvery](#)
Primitives for computing the minimal value of the pixel pair from two images.

- [Integral](#)
Primitives for computing the integral image of a given image.
- [SqrIntegral](#)
Primitives for computing both the integral and the squared integral images of a given image.
- [RectStdDev](#)
Primitives for computing the standard deviation of the integral images.
- [HistogramEven](#)
Primitives for computing the histogram of an image with evenly distributed bins.
- [HistogramRange](#)
Primitives for computing the histogram of an image within specified ranges.
- [Image Proximity](#)
Primitives for computing the proximity measure between a source image and a template image.
- [Image Quality Index](#)
Primitives for computing the image quality index of two images.
- [MaximumError](#)
Primitives for computing the maximum error between two images.
- [AverageError](#)
Primitives for computing the average error between two images.
- [MaximumRelativeError](#)
Primitives for computing the maximum relative error between two images.
- [AverageRelativeError](#)
Primitives for computing the average relative error between two images.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumError_8u_C1R`.
- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumError_8s_C1R`.
- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

- Buffer size for nppiMaximumError_16u_C1R.*
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_16s_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_16s_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_16sc_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_16sc_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_32u_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_32u_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_32s_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_32s_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_32sc_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_32sc_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_32f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_32f_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_32fc_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_32fc_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_64f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_64f_C1R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_8u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_8u_C2R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_8s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_8s_C2R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_16u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_16u_C2R.
 - **NppStatus** `nppiMaximumErrorGetBufferHostSize_16s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for nppiMaximumError_16s_C2R.

- **NppStatus** `nppiMaximumErrorGetBufferHostSize_16sc_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_16sc_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32u_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32u_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32s_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32s_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32sc_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32sc_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32f_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32f_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32fc_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32fc_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_64f_C2R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_64f_C2R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_8u_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_8u_C3R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_8s_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_8s_C3R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_16u_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_16u_C3R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_16s_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_16s_C3R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_16sc_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_16sc_C3R`.
- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32u_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumError_32u_C3R`.

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32s_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32s_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32sc_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32sc_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32f_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32fc_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32fc_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_64f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_64f_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_8u_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_8s_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_16u_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_16s_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16sc_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_16sc_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32u_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumError_32s_C4R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32sc_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiMaximumError_32sc_C4R.

- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32f_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiMaximumError_32f_C4R.

- **NppStatus** `nppiMaximumErrorGetBufferHostSize_32fc_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiMaximumError_32fc_C4R.

- **NppStatus** `nppiMaximumErrorGetBufferHostSize_64f_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiMaximumError_64f_C4R.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_8u_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_8s_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_16u_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_16s_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_16sc_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_32u_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_32s_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C1R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Buffer size for nppiAverageError_32sc_C1R.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32f_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32fc_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_64f_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_8u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_8s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16sc_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32sc_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32f_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32fc_C2R`.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C2R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_64f_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16sc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32sc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32f_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32fc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_64f_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8s_C4R`.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_16u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_16s_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_16sc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_32u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_32s_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_32sc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_32f_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_32fc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C4R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiAverageError_64f_C4R`.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `NormDiff_Inf` primitives.

- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8u_C1R`.
- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R` (`NppiSize` `oSizeROI`, `int`
*`hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8s_C1R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16u_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16s_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16sc_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32u_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32s_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32sc_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32f_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32fc_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_64f_C1R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8u_C2R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8s_C2R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16u_C2R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_16s_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_16sc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_32u_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_32s_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_32sc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_32f_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_32fc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_64f_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_8u_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_8s_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_16u_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_16s_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiMaximumRelativeError_16sc_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32u_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32s_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32sc_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32f_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32fc_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_64f_C3R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8u_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_8s_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16u_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16s_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_16sc_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32u_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32s_C4R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32sc_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32f_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_32fc_C4R`.
- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMaximumRelativeError_64f_C4R`.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `NormDiff_Inf` primitives.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_8u_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_8s_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16u_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16s_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16sc_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32u_C1R`.
- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32s_C1R`.

- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32sc_C1R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32f_C1R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32fc_C1R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_64f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_64f_C1R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_8u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_8u_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_8s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_8s_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16u_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16s_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_16sc_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32u_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32s_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiAverageRelativeError_32sc_C2R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32f_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32f_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32fc_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C2R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_64f_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_8u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_8s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_16u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_16s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_16sc_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32sc_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32f_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32fc_C3R`.

- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_64f_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_64f_C3R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_8u_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_8u_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_8s_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_8s_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16u_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_16u_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16s_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_16s_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_16sc_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32u_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_32u_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32s_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_32s_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_32sc_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_32f_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_32fc_C4R`.
- **NppStatus** `nppiAverageRelativeErrorGetBufferHostSize_64f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageRelativeError_64f_C4R`.

7.4.1 Detailed Description

Primitives for computing the statistical properties of an image.

Some statistical primitives also require scratch buffer during the computation. For details, please refer to [Scratch Buffer and Host Pointer](#).

These functions can be found in either the nppi or nppist libraries. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

7.4.2 Function Documentation

7.4.2.1 NppStatus nppiAverageErrorGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.2 NppStatus nppiAverageErrorGetBufferHostSize_16s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.3 NppStatus nppiAverageErrorGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.4 NppStatus nppiAverageErrorGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.5 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_16sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.6 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_16sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.7 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.8 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.9 NppStatus nppiAverageErrorGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.10 NppStatus nppiAverageErrorGetBufferHostSize_16u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.11 NppStatus nppiAverageErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.12 NppStatus nppiAverageErrorGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.13 NppStatus nppiAverageErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.14 `NppStatus nppiAverageErrorGetBufferHostSize_32f_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32f_C2R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.15 `NppStatus nppiAverageErrorGetBufferHostSize_32f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32f_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.16 `NppStatus nppiAverageErrorGetBufferHostSize_32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32f_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.17 `NppStatus nppiAverageErrorGetBufferHostSize_32fc_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32fc_C1R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.18 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32fc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.19 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32fc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.20 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32fc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.21 `NppStatus nppiAverageErrorGetBufferHostSize_32s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32s_C1R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.22 `NppStatus nppiAverageErrorGetBufferHostSize_32s_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32s_C2R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.23 `NppStatus nppiAverageErrorGetBufferHostSize_32s_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32s_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.24 `NppStatus nppiAverageErrorGetBufferHostSize_32s_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32s_C4R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.25 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.26 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.27 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.28 `NppStatus nppiAverageErrorGetBufferHostSize_32sc_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32sc_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.29 `NppStatus nppiAverageErrorGetBufferHostSize_32u_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32u_C1R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.30 `NppStatus nppiAverageErrorGetBufferHostSize_32u_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32u_C2R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.31 `NppStatus nppiAverageErrorGetBufferHostSize_32u_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_32u_C3R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.32 NppStatus nppiAverageErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.33 NppStatus nppiAverageErrorGetBufferHostSize_64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_64f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.34 NppStatus nppiAverageErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_64f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.35 `NppStatus nppiAverageErrorGetBufferHostSize_64f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_64f_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.36 `NppStatus nppiAverageErrorGetBufferHostSize_64f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_64f_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.37 `NppStatus nppiAverageErrorGetBufferHostSize_8s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_8s_C1R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.38 `NppStatus nppiAverageErrorGetBufferHostSize_8s_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_8s_C2R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.39 NppStatus nppiAverageErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.40 NppStatus nppiAverageErrorGetBufferHostSize_8s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.41 NppStatus nppiAverageErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.42 `NppStatus nppiAverageErrorGetBufferHostSize_8u_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_8u_C2R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.43 `NppStatus nppiAverageErrorGetBufferHostSize_8u_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_8u_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.44 `NppStatus nppiAverageErrorGetBufferHostSize_8u_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageError_8u_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.45 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageRelativeError_16s_C1R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.46 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.47 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.48 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.49 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.50 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.51 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.52 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.53 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.54 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.55 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.56 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.57 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.58 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.59 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.60 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.61 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32fc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.62 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32fc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.63 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32fc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.64 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32fc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.65 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.66 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.67 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.68 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.69 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.70 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32sc_C2R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.71 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32sc_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.72 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32sc_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.73 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiAverageRelativeError_32u_C1R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.74 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.75 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.76 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.77 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_64f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.78 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_64f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.79 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_64f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.80 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_64f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.81 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.82 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.83 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.84 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_8s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.85 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.86 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_8u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.87 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.88 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.89 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.90 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.91 `NppStatus nppiMaximumErrorGetBufferHostSize_16s_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiMaximumError_16s_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.92 `NppStatus nppiMaximumErrorGetBufferHostSize_16s_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiMaximumError_16s_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.93 `NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiMaximumError_16sc_C1R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.94 `NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiMaximumError_16sc_C2R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.95 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.96 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.97 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.98 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.99 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.100 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.101 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.102 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.103 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.104 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.105 `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumError_32fc_C1R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.106 `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumError_32fc_C2R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.107 `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumError_32fc_C3R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.108 `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumError_32fc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.109 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.110 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.111 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.112 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.113 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.114 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.115 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.116 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.117 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.118 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.119 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.120 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.121 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.122 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.123 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.124 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.125 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_8s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.126 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.127 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.128 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.129 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.130 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_8u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.131 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.132 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.133 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.134 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.135 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.136 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.137 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.138 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.139 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.140 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.141 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.142 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.143 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.144 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.145 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.146 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.147 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.148 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.149 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.150 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.151 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.152 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.153 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.154 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.155 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.156 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.157 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.158 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.159 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.160 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.161 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.162 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.163 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.164 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.165 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_64f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.166 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_64f_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.167 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_64f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.168 `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumRelativeError_64f_C4R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.169 `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumRelativeError_8s_C1R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.170 `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumRelativeError_8s_C2R](#).

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.171 `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMaximumRelativeError_8s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.172 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.173 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.174 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C2R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.175 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.4.2.176 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5 Sum

Primitives for computing the sum of all the pixel values in an image.

Sum

Given an image $pSrc$ with width W and height H , the sum will be computed as

$$Sum = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

All the results are stored in a 64-bit double precision format, except for two primitives `ippiSum_8u64s_C1R` and `ippiSum_8u64s_C4R`.

The sum functions require additional scratch buffer for computations.

- `NppStatus ippiSum_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 8-bit unsigned image sum.
- `NppStatus ippiSum_8u64s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64s` *pSum)
One-channel 8-bit unsigned image sum.
- `NppStatus ippiSum_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 16-bit unsigned image sum.
- `NppStatus ippiSum_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 16-bit signed image sum.
- `NppStatus ippiSum_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 32-bit floating point image sum.
- `NppStatus ippiSum_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 8-bit unsigned image sum.
- `NppStatus ippiSum_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 16-bit unsigned image sum.
- `NppStatus ippiSum_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 16-bit signed image sum.
- `NppStatus ippiSum_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 32-bit floating point image sum.

- `NppStatus nppiSum_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Four-channel 8-bit unsigned image sum ignoring alpha channel.
- `NppStatus nppiSum_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Four-channel 16-bit unsigned image sum ignoring alpha channel.
- `NppStatus nppiSum_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Four-channel 16-bit signed image sum ignoring alpha channel.
- `NppStatus nppiSum_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Four-channel 32-bit floating point image sum ignoring alpha channel.
- `NppStatus nppiSum_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[4])
Four-channel 8-bit unsigned image sum.
- `NppStatus nppiSum_8u64s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64s` aSum[4])
Four-channel 8-bit unsigned image sum.
- `NppStatus nppiSum_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[4])
Four-channel 16-bit unsigned image sum.
- `NppStatus nppiSum_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[4])
Four-channel 16-bit signed image sum.
- `NppStatus nppiSum_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[4])
Four-channel 32-bit floating point image sum.

SumGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the sum primitives.

- `NppStatus nppiSumGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiSum_8u_C1R`.
- `NppStatus nppiSumGetBufferHostSize_8u64s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiSum_8u64s_C1R`.
- `NppStatus nppiSumGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiSum_16u_C1R`.

- `NppStatus nppiSumGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16s_C1R`.
- `NppStatus nppiSumGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_32f_C1R`.
- `NppStatus nppiSumGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_8u_C3R`.
- `NppStatus nppiSumGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16u_C3R`.
- `NppStatus nppiSumGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16s_C3R`.
- `NppStatus nppiSumGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_32f_C3R`.
- `NppStatus nppiSumGetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_8u_AC4R`.
- `NppStatus nppiSumGetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16u_AC4R`.
- `NppStatus nppiSumGetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16s_AC4R`.
- `NppStatus nppiSumGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_32f_AC4R`.
- `NppStatus nppiSumGetBufferHostSize_8u64s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_8u64s_C4R`.
- `NppStatus nppiSumGetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_8u_C4R`.
- `NppStatus nppiSumGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16u_C4R`.
- `NppStatus nppiSumGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_16s_C4R`.
- `NppStatus nppiSumGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiSum_32f_C4R`.

7.5.1 Detailed Description

Primitives for computing the sum of all the pixel values in an image.

7.5.2 Function Documentation

7.5.2.1 `NppStatus nppiSum_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Four-channel 16-bit signed image sum ignoring alpha channel.

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](#).
Use [nppiSumGetBufferHostSize_16s_AC4R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

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

7.5.2.2 `NppStatus nppiSum_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)`

One-channel 16-bit signed image sum.

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](#).
Use [nppiSumGetBufferHostSize_16s_C1R](#) to determine the minimum number of bytes required.

pSum Pointer to the computed sum.

Returns:

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

7.5.2.3 `NppStatus nppiSum_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Three-channel 16-bit signed image sum.

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](#).
Use [nppiSumGetBufferHostSize_16s_C3R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.4 NppStatus nppiSum_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])

Four-channel 16-bit signed image sum.

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](#) Use [nppiSumGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.5 NppStatus nppiSum_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Four-channel 16-bit unsigned image sum ignoring alpha channel.

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](#). Use [nppiSumGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

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

7.5.2.6 NppStatus nppiSum_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)

One-channel 16-bit unsigned image sum.

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](#).
Use [nppiSumGetBufferHostSize_16u_C1R](#) to determine the minimum number of bytes required.

pSum Pointer to the computed sum.

Returns:

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

7.5.2.7 NppStatus nppiSum_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Three-channel 16-bit unsigned image sum.

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](#).
Use [nppiSumGetBufferHostSize_16u_C3R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.8 NppStatus nppiSum_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])

Four-channel 16-bit unsigned image sum.

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](#) Use
[nppiSumGetBufferHostSize_16u_C4R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.9 NppStatus nppiSum_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Four-channel 32-bit floating point image sum ignoring alpha channel.

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](#) Use [nppiSumGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

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

7.5.2.10 NppStatus nppiSum_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)

One-channel 32-bit floating point image sum.

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](#) Use [nppiSumGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

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

7.5.2.11 NppStatus nppiSum_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Three-channel 32-bit floating point image sum.

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](#). Use [nppiSumGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.12 `NppStatus nppiSum_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])`

Four-channel 32-bit floating point image sum.

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](#) Use [nppiSumGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.13 `NppStatus nppiSum_8u64s_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64s * pSum)`

One-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

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](#). Use [nppiSumGetBufferHostSize_8u64s_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

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

7.5.2.14 `NppStatus nppiSum_8u64s_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64s aSum[4])`

Four-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

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](#) Use [nppiSumGetBufferHostSize_8u64s_C4R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.15 `NppStatus nppiSum_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Four-channel 8-bit unsigned image sum ignoring alpha channel.

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](#). Use [nppiSumGetBufferHostSize_8u_AC4R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

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

7.5.2.16 `NppStatus nppiSum_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)`

One-channel 8-bit unsigned image sum.

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](#). Use [nppiSumGetBufferHostSize_8u_C1R](#) to determine the minimum number of bytes required.

pSum Pointer to the computed sum.

Returns:

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

7.5.2.17 NppStatus nppiSum_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Three-channel 8-bit unsigned image sum.

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](#).

Use [nppiSumGetBufferHostSize_8u_C3R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.18 NppStatus nppiSum_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])

Four-channel 8-bit unsigned image sum.

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](#) Use

[nppiSumGetBufferHostSize_8u_C4R](#) to determine the minimum number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

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

7.5.2.19 NppStatus nppiSumGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.20 NppStatus nppiSumGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.21 NppStatus nppiSumGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.22 NppStatus nppiSumGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.23 NppStatus nppiSumGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.24 NppStatus nppiSumGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiSum_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.25 NppStatus nppiSumGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiSum_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.26 NppStatus nppiSumGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiSum_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.27 NppStatus nppiSumGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiSum_32f_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.28 NppStatus nppiSumGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.29 NppStatus nppiSumGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.30 NppStatus nppiSumGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.31 `NppStatus nppiSumGetBufferHostSize_8u64s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiSum_8u64s_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.32 `NppStatus nppiSumGetBufferHostSize_8u64s_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiSum_8u64s_C4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.33 `NppStatus nppiSumGetBufferHostSize_8u_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiSum_8u_AC4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.34 `NppStatus nppiSumGetBufferHostSize_8u_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for `nppiSum_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.35 NppStatus nppiSumGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.5.2.36 NppStatus nppiSumGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6 Min

Primitives for computing the minimal pixel value of an image.

Min

The scratch buffer is required by the min functions.

- `NppStatus nppiMin_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u *pMin`)
One-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u *pMin`)
One-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s *pMin`)
One-channel 16-bit signed image min.
- `NppStatus nppiMin_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f *pMin`)
One-channel 32-bit floating point image min.
- `NppStatus nppiMin_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMin[3]`)
Three-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMin[3]`)
Three-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMin[3]`)
Three-channel 16-bit signed image min.
- `NppStatus nppiMin_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMin[3]`)
Three-channel 32-bit floating point image min.
- `NppStatus nppiMin_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMin[4]`)
Four-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMin[4]`)
Four-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMin[4]`)

Four-channel 16-bit signed image min.

- `NppStatus nppiMin_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMin[4]`)

Four-channel 32-bit floating point image min.

- `NppStatus nppiMin_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMin[3]`)

Four-channel 8-bit unsigned image min ignoring alpha channel.

- `NppStatus nppiMin_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMin[3]`)

Four-channel 16-bit unsigned image min ignoring alpha channel.

- `NppStatus nppiMin_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMin[3]`)

Four-channel 16-bit signed image min ignoring alpha channel.

- `NppStatus nppiMin_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMin[3]`)

Four-channel 32-bit floating point image min ignoring alpha channel.

MinGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the min primitives.

- `NppStatus nppiMinGetBufferHostSize_8u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_8u_C1R`.
- `NppStatus nppiMinGetBufferHostSize_16u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_16u_C1R`.
- `NppStatus nppiMinGetBufferHostSize_16s_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_16s_C1R`.
- `NppStatus nppiMinGetBufferHostSize_32f_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_32f_C1R`.
- `NppStatus nppiMinGetBufferHostSize_8u_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_8u_C3R`.
- `NppStatus nppiMinGetBufferHostSize_16u_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_16u_C3R`.
- `NppStatus nppiMinGetBufferHostSize_16s_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_16s_C3R`.
- `NppStatus nppiMinGetBufferHostSize_32f_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMin_32f_C3R`.

- `NppStatus nppiMinGetBufferHostSize_8u_C4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_8u_C4R`.
- `NppStatus nppiMinGetBufferHostSize_16u_C4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_16u_C4R`.
- `NppStatus nppiMinGetBufferHostSize_16s_C4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_16s_C4R`.
- `NppStatus nppiMinGetBufferHostSize_32f_C4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_32f_C4R`.
- `NppStatus nppiMinGetBufferHostSize_8u_AC4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_8u_AC4R`.
- `NppStatus nppiMinGetBufferHostSize_16u_AC4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_16u_AC4R`.
- `NppStatus nppiMinGetBufferHostSize_16s_AC4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_16s_AC4R`.
- `NppStatus nppiMinGetBufferHostSize_32f_AC4R` (`NppiSize oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMin_32f_AC4R`.

7.6.1 Detailed Description

Primitives for computing the minimal pixel value of an image.

7.6.2 Function Documentation

7.6.2.1 `NppStatus nppiMin_16s_AC4R` (`const Npp16s * pSrc`, `int nSrcStep`, `NppiSize oSizeROI`, `Npp8u * pDeviceBuffer`, `Npp16s aMin[3]`)

Four-channel 16-bit signed image min ignoring alpha channel.

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](#) Use `nppiMinGetBufferHostSize_16s_AC4R` to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

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

7.6.2.2 NppStatus nppiMin_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMin)

One-channel 16-bit signed image min.

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](#) Use [nppiMinGetBufferHostSize_16s_C1R](#) to determine the minimum number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

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

7.6.2.3 NppStatus nppiMin_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3])

Three-channel 16-bit signed image min.

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](#) Use [nppiMinGetBufferHostSize_16s_C3R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.4 NppStatus nppiMin_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[4])

Four-channel 16-bit signed image min.

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](#) Use [nppiMinGetBufferHostSize_16s_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.5 `NppStatus nppiMin_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3])`

Four-channel 16-bit unsigned image min ignoring alpha channel.

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](#) Use [nppiMinGetBufferHostSize_16u_AC4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

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

7.6.2.6 `NppStatus nppiMin_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMin)`

One-channel 16-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_16u_C1R](#) to determine the minimum number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

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

7.6.2.7 `NppStatus nppiMin_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3])`

Three-channel 16-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_16u_C3R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.8 NppStatus nppiMin_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[4])

Four-channel 16-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_16u_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.9 NppStatus nppiMin_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3])

Four-channel 32-bit floating point image min ignoring alpha channel.

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](#) Use [nppiMinGetBufferHostSize_32f_AC4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

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

7.6.2.10 NppStatus nppiMin_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMin)

One-channel 32-bit floating point image min.

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](#) Use [nppiMinGetBufferHostSize_32f_C1R](#) to determine the minimum number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

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

7.6.2.11 NppStatus nppiMin_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3])

Three-channel 32-bit floating point image min.

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](#) Use [nppiMinGetBufferHostSize_32f_C3R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.12 NppStatus nppiMin_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4])

Four-channel 32-bit floating point image min.

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](#) Use [nppiMinGetBufferHostSize_32f_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.13 NppStatus nppiMin_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3])

Four-channel 8-bit unsigned image min ignoring alpha channel.

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](#) Use [nppiMinGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

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

7.6.2.14 NppStatus nppiMin_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMin)

One-channel 8-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

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

7.6.2.15 NppStatus nppiMin_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3])

Three-channel 8-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.16 NppStatus nppiMin_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4])

Four-channel 8-bit unsigned image min.

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](#) Use [nppiMinGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

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

7.6.2.17 NppStatus nppiMinGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.18 NppStatus nppiMinGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16s_C1R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.19 NppStatus nppiMinGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.20 NppStatus nppiMinGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.21 NppStatus nppiMinGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.22 NppStatus nppiMinGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.23 NppStatus nppiMinGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.24 NppStatus nppiMinGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.25 NppStatus nppiMinGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_32f_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.26 NppStatus nppiMinGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.27 NppStatus nppiMinGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.28 NppStatus nppiMinGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.29 NppStatus nppiMinGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.30 NppStatus nppiMinGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.31 NppStatus nppiMinGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.6.2.32 NppStatus nppiMinGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7 MinIndx

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

MinIndx

If there are several minima in the selected ROI, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- **NppStatus nppiMinIndx_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** *pMin, int *pIndexX, int *pIndexY)
One-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** *pMin, int *pIndexX, int *pIndexY)
One-channel 16-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** *pMin, int *pIndexX, int *pIndexY)
One-channel 16-bit signed image MinIndx.
- **NppStatus nppiMinIndx_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** *pMin, int *pIndexX, int *pIndexY)
One-channel 32-bit floating point image MinIndx.
- **NppStatus nppiMinIndx_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit signed image MinIndx.
- **NppStatus nppiMinIndx_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 32-bit floating point image MinIndx.
- **NppStatus nppiMinIndx_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit unsigned image MinIndx.

- `NppStatus nppiMinIndx_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit signed image MinIndx.
- `NppStatus nppiMinIndx_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 32-bit floating point image MinIndx.
- `NppStatus nppiMinIndx_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 8-bit unsigned image MinIndx ignoring alpha channel.
- `NppStatus nppiMinIndx_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.
- `NppStatus nppiMinIndx_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit signed image MinIndx ignoring alpha channel.
- `NppStatus nppiMinIndx_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 32-bit floating point image MinIndx ignoring alpha channel.

MinIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinIndx primitives.

- `NppStatus nppiMinIndxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C1R.
- `NppStatus nppiMinIndxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C1R.
- `NppStatus nppiMinIndxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16s_C1R.
- `NppStatus nppiMinIndxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_C1R.
- `NppStatus nppiMinIndxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C3R.
- `NppStatus nppiMinIndxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C3R.
- `NppStatus nppiMinIndxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16s_C3R.

- `NppStatus nppiMinIndxGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_32f_C3R`.
- `NppStatus nppiMinIndxGetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_8u_C4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_16u_C4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_16s_C4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_32f_C4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_8u_AC4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_16u_AC4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_16s_AC4R`.
- `NppStatus nppiMinIndxGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiMinIndx_32f_AC4R`.

7.7.1 Detailed Description

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

7.7.2 Function Documentation

7.7.2.1 `NppStatus nppiMinIndx_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMin[3]`, `int aIndexX[3]`, `int aIndexY[3]`)

Four-channel 16-bit signed image MinIndx ignoring alpha channel.

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](#) Use `nppiMinIndxGetBufferHostSize_16s_AC4R` to determine the minimum number of bytes required.

`aMin` Array that contains the min values.

`aIndexX` Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.2 NppStatus nppiMinIndx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMin, int * pIndexX, int * pIndexY)

One-channel 16-bit signed image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

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

7.7.2.3 NppStatus nppiMinIndx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit signed image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.4 NppStatus nppiMinIndx_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit signed image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.5 NppStatus nppiMinIndx_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.

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](#) Use [nppiMinIndxGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.6 NppStatus nppiMinIndx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMin, int * pIndexX, int * pIndexY)

One-channel 16-bit unsigned image MinIndx.

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](#) Use [nppiMinIdxGetBufferHostSize_16u_C1R](#) to determine the minimum number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

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

7.7.2.7 NppStatus nppiMinIdx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit unsigned image MinIdx.

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](#) Use [nppiMinIdxGetBufferHostSize_16u_C3R](#) to determine the minimum number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.8 NppStatus nppiMinIdx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit unsigned image MinIdx.

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](#) Use [nppiMinIdxGetBufferHostSize_16u_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.9 NppStatus nppiMinIndx_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 32-bit floating point image MinIndx ignoring alpha channel.

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](#) Use [nppiMinIndxGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.10 NppStatus nppiMinIndx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMin, int * pIndexX, int * pIndexY)

One-channel 32-bit floating point image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

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

7.7.2.11 NppStatus nppiMinIndx_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 32-bit floating point image MinIndx.

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](#) Use [nppiMinIdxGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.12 NppStatus nppiMinIdx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 32-bit floating point image MinIdx.

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](#) Use [nppiMinIdxGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.13 NppStatus nppiMinIdx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 8-bit unsigned image MinIdx ignoring alpha channel.

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](#) Use [nppiMinIdxGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.14 NppStatus nppiMinIndx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMin, int * pIndexX, int * pIndexY)

One-channel 8-bit unsigned image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Ppointer to the Y coordinate of the image min value.

Returns:

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

7.7.2.15 NppStatus nppiMinIndx_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 8-bit unsigned image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.16 NppStatus nppiMinIndx_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 8-bit unsigned image MinIndx.

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](#) Use [nppiMinIndxGetBufferHostSize_8u_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

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

7.7.2.17 NppStatus nppiMinIndxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMinIndx_16u_AC4R](#).

Parameters:

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

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

Returns:

[NPP_NULL_POINTER_ERROR](#) if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.18 NppStatus nppiMinIndxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMinIndx_16s_C1R](#).

Parameters:

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

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

Returns:

[NPP_NULL_POINTER_ERROR](#) if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.19 NppStatus nppiMinIndxGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMinIndx_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.20 NppStatus nppiMinIndxGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_16s_C4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.21 NppStatus nppiMinIndxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_8u_AC4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.22 NppStatus nppiMinIndxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_16u_C1R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.23 `NppStatus nppiMinIdxGetBufferHostSize_16u_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiMinIdx_16u_C3R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.24 `NppStatus nppiMinIdxGetBufferHostSize_16u_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiMinIdx_16u_C4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.25 `NppStatus nppiMinIdxGetBufferHostSize_32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiMinIdx_32f_AC4R`.

Parameters:

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

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

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.26 `NppStatus nppiMinIdxGetBufferHostSize_32f_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiMinIdx_32f_C1R`.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.27 NppStatus nppiMinIndxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_32f_C3R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.28 NppStatus nppiMinIndxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_32f_C4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.29 NppStatus nppiMinIndxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIndx_8u_AC4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.30 NppStatus nppiMinIdxGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.31 NppStatus nppiMinIdxGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_C3R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.7.2.32 NppStatus nppiMinIdxGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_C4R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8 Max

Primitives for computing the maximal pixel value of an image.

Max

The scratch buffer is required by the functions.

- `NppStatus nppiMax_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u *pMax`)
One-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u *pMax`)
One-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s *pMax`)
One-channel 16-bit signed image Max.
- `NppStatus nppiMax_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f *pMax`)
One-channel 32-bit floating point image Max.
- `NppStatus nppiMax_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMax[3]`)
Three-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMax[3]`)
Three-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMax[3]`)
Three-channel 16-bit signed image Max.
- `NppStatus nppiMax_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMax[3]`)
Three-channel 32-bit floating point image Max.
- `NppStatus nppiMax_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMax[4]`)
Four-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMax[4]`)
Four-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMax[4]`)

Four-channel 16-bit signed image Max.

- `NppStatus nppiMax_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[4])

Four-channel 32-bit floating point image Max.

- `NppStatus nppiMax_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[3])

Four-channel 8-bit unsigned image Max ignoring alpha channel.

- `NppStatus nppiMax_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[3])

Four-channel 16-bit unsigned image Max ignoring alpha channel.

- `NppStatus nppiMax_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMax[3])

Four-channel 16-bit signed image Max ignoring alpha channel.

- `NppStatus nppiMax_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[3])

Four-channel 32-bit floating point image Max ignoring alpha channel.

MaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Max primitives.

- `NppStatus nppiMaxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_8u_C1R`.
- `NppStatus nppiMaxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_16u_C1R`.
- `NppStatus nppiMaxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_16s_C1R`.
- `NppStatus nppiMaxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_32f_C1R`.
- `NppStatus nppiMaxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_8u_C3R`.
- `NppStatus nppiMaxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_16u_C3R`.
- `NppStatus nppiMaxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_16s_C3R`.
- `NppStatus nppiMaxGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMax_32f_C3R`.

- `NppStatus nppiMaxGetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_8u_C4R`.
- `NppStatus nppiMaxGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_16u_C4R`.
- `NppStatus nppiMaxGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_16s_C4R`.
- `NppStatus nppiMaxGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_32f_C4R`.
- `NppStatus nppiMaxGetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_8u_AC4R`.
- `NppStatus nppiMaxGetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_16u_AC4R`.
- `NppStatus nppiMaxGetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_16s_AC4R`.
- `NppStatus nppiMaxGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMax_32f_AC4R`.

7.8.1 Detailed Description

Primitives for computing the maximal pixel value of an image.

7.8.2 Function Documentation

7.8.2.1 `NppStatus nppiMax_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMax[3]`)

Four-channel 16-bit signed image Max ignoring alpha channel.

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](#) Use `nppiMaxGetBufferHostSize_16s_AC4R` to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

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

7.8.2.2 `NppStatus nppiMax_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMax)`

One-channel 16-bit signed image Max.

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](#) Use [nppiMaxGetBufferHostSize_16s_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

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

7.8.2.3 `NppStatus nppiMax_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3])`

Three-channel 16-bit signed image Max.

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](#) Use [nppiMaxGetBufferHostSize_16s_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.4 `NppStatus nppiMax_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[4])`

Four-channel 16-bit signed image Max.

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](#) Use [nppiMaxGetBufferHostSize_16s_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.5 NppStatus nppiMax_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3])

Four-channel 16-bit unsigned image Max ignoring alpha channel.

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](#) Use [nppiMaxGetBufferHostSize_16u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

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

7.8.2.6 NppStatus nppiMax_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMax)

One-channel 16-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_16u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

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

7.8.2.7 NppStatus nppiMax_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3])

Three-channel 16-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_16u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.8 NppStatus nppiMax_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4])

Four-channel 16-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_16u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.9 NppStatus nppiMax_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])

Four-channel 32-bit floating point image Max ignoring alpha channel.

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](#) Use [nppiMaxGetBufferHostSize_32f_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

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

7.8.2.10 NppStatus nppiMax_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMax)

One-channel 32-bit floating point image Max.

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](#) Use [nppiMaxGetBufferHostSize_32f_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

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

7.8.2.11 NppStatus nppiMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])

Three-channel 32-bit floating point image Max.

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](#) Use [nppiMaxGetBufferHostSize_32f_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.12 NppStatus nppiMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4])

Four-channel 32-bit floating point image Max.

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](#) Use [nppiMaxGetBufferHostSize_32f_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.13 NppStatus nppiMax_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3])

Four-channel 8-bit unsigned image Max ignoring alpha channel.

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](#) Use [nppiMaxGetBufferHostSize_8u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

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

7.8.2.14 NppStatus nppiMax_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMax)

One-channel 8-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_8u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

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

7.8.2.15 NppStatus nppiMax_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3])

Three-channel 8-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_8u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.16 NppStatus nppiMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4])

Four-channel 8-bit unsigned image Max.

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](#) Use [nppiMaxGetBufferHostSize_8u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

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

7.8.2.17 NppStatus nppiMaxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.18 NppStatus nppiMaxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_C1R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.19 NppStatus nppiMaxGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_C3R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.20 NppStatus nppiMaxGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_C4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.21 NppStatus nppiMaxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.22 NppStatus nppiMaxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C1R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.23 NppStatus nppiMaxGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.24 NppStatus nppiMaxGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.25 NppStatus nppiMaxGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.26 NppStatus nppiMaxGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.27 NppStatus nppiMaxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.28 NppStatus nppiMaxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.29 NppStatus nppiMaxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.30 NppStatus nppiMaxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.31 NppStatus nppiMaxGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.8.2.32 NppStatus nppiMaxGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9 MaxIndx

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

MaxIndx

If there are several maxima in the selected region of interest, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- `NppStatus nppiMaxIndx_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` *pMax, int *pIndexX, int *pIndexY)
One-channel 8-bit unsigned image MaxIndx.
- `NppStatus nppiMaxIndx_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` *pMax, int *pIndexX, int *pIndexY)
One-channel 16-bit unsigned image MaxIndx.
- `NppStatus nppiMaxIndx_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` *pMax, int *pIndexX, int *pIndexY)
One-channel 16-bit signed image MaxIndx.
- `NppStatus nppiMaxIndx_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` *pMax, int *pIndexX, int *pIndexY)
One-channel 32-bit floating point image MaxIndx.
- `NppStatus nppiMaxIndx_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 8-bit unsigned image MaxIndx.
- `NppStatus nppiMaxIndx_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit unsigned image MaxIndx.
- `NppStatus nppiMaxIndx_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit signed image MaxIndx.
- `NppStatus nppiMaxIndx_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 32-bit floating point image MaxIndx.
- `NppStatus nppiMaxIndx_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 8-bit unsigned image MaxIndx.
- `NppStatus nppiMaxIndx_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit unsigned image MaxIndx.

- **NppStatus nppiMaxIndx_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit signed image MaxIndx.
- **NppStatus nppiMaxIndx_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 32-bit floating point image MaxIndx.
- **NppStatus nppiMaxIndx_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 8-bit unsigned image MaxIndx ignoring alpha channel.
- **NppStatus nppiMaxIndx_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit unsigned image MaxIndx ignoring alpha channel.
- **NppStatus nppiMaxIndx_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit signed image MaxIndx ignoring alpha channel.
- **NppStatus nppiMaxIndx_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 32-bit floating point image MaxIndx ignoring alpha channel.

MaxIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MaxIndx primitives.

- **NppStatus nppiMaxIndxGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C1R.
- **NppStatus nppiMaxIndxGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C1R.
- **NppStatus nppiMaxIndxGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C1R.
- **NppStatus nppiMaxIndxGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C1R.
- **NppStatus nppiMaxIndxGetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C3R.
- **NppStatus nppiMaxIndxGetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C3R.
- **NppStatus nppiMaxIndxGetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C3R.

- [NppStatus nppiMaxIdxGetBufferHostSize_32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_C3R.
- [NppStatus nppiMaxIdxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_C4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_C4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_C4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_C4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_AC4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_AC4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_AC4R.
- [NppStatus nppiMaxIdxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int *hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_AC4R.

7.9.1 Detailed Description

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

7.9.2 Function Documentation

7.9.2.1 [NppStatus nppiMaxIdx_16s_AC4R](#) ([const Npp16s *pSrc](#), [int nSrcStep](#), [NppiSize oSizeROI](#), [Npp8u *pDeviceBuffer](#), [Npp16s aMax\[3\]](#), [int aIndexX\[3\]](#), [int aIndexY\[3\]](#))

Four-channel 16-bit signed image MaxIdx ignoring alpha channel.

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](#)
Use [nppiMaxIdxGetBufferHostSize_16s_AC4R](#) to determine the maximum number of bytes required.

aMax [Array that contains the max values](#).

aIndexX [Array that contains the X coordinates of the image max values](#).

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.2 NppStatus nppiMaxIndx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMax, int * pIndexX, int * pIndexY)

One-channel 16-bit signed image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16s_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

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

7.9.2.3 NppStatus nppiMaxIndx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit signed image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16s_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.4 NppStatus nppiMaxIndx_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit signed image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16s_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.5 NppStatus nppiMaxIndx_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit unsigned image MaxIndx ignoring alpha channel.

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](#) Use [nppiMaxIndxGetBufferHostSize_16u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.6 NppStatus nppiMaxIndx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMax, int * pIndexX, int * pIndexY)

One-channel 16-bit unsigned image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

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

7.9.2.7 NppStatus nppiMaxIndx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit unsigned image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.8 NppStatus nppiMaxIndx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit unsigned image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_16u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.9 `NppStatus nppiMaxIdx_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3], int aIndexX[3], int aIndexY[3])`

Four-channel 32-bit floating point image MaxIdx ignoring alpha channel.

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](#) Use [nppiMaxIdxGetBufferHostSize_32f_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.10 `NppStatus nppiMaxIdx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMax, int * pIndexX, int * pIndexY)`

One-channel 32-bit floating point image MaxIdx.

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](#) Use [nppiMaxIdxGetBufferHostSize_32f_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

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

7.9.2.11 `NppStatus nppiMaxIdx_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 32-bit floating point image MaxIdx.

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](#) Use [nppiMaxIndxGetBufferHostSize_32f_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.12 NppStatus nppiMaxIndx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4], int aIndexX[4], int aIndexY[4])

Four-channel 32-bit floating point image MaxIndx.

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](#) Use [nppiMaxIndxGetBufferHostSize_32f_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.13 NppStatus nppiMaxIndx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 8-bit unsigned image MaxIndx ignoring alpha channel.

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](#) Use [nppiMaxIndxGetBufferHostSize_8u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.14 `NppStatus nppiMaxIdx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMax, int * pIndexX, int * pIndexY)`

One-channel 8-bit unsigned image MaxIdx.

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](#) Use [nppiMaxIdxGetBufferHostSize_8u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

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

7.9.2.15 `NppStatus nppiMaxIdx_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 8-bit unsigned image MaxIdx.

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](#) Use [nppiMaxIdxGetBufferHostSize_8u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.16 `NppStatus nppiMaxIdx_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4], int aIndexX[4], int aIndexY[4])`

Four-channel 8-bit unsigned image MaxIdx.

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](#) Use [nppiMaxIndxGetBufferHostSize_8u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

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

7.9.2.17 NppStatus nppiMaxIndxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMaxIndx_16u_AC4R](#).

Parameters:

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

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

Returns:

[NPP_NULL_POINTER_ERROR](#) if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.18 NppStatus nppiMaxIndxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMaxIndx_16s_C1R](#).

Parameters:

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

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

Returns:

[NPP_NULL_POINTER_ERROR](#) if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.19 NppStatus nppiMaxIndxGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for [nppiMaxIndx_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.20 NppStatus nppiMaxIndxGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_16s_C4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.21 NppStatus nppiMaxIndxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_8u_AC4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.22 NppStatus nppiMaxIndxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_16u_C1R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.23 NppStatus nppiMaxIndxGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C3R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.24 NppStatus nppiMaxIndxGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C4R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.25 NppStatus nppiMaxIndxGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_AC4R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.26 NppStatus nppiMaxIndxGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C1R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.27 NppStatus nppiMaxIndxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_32f_C3R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.28 NppStatus nppiMaxIndxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_32f_C4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.29 NppStatus nppiMaxIndxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_8u_AC4R*.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.30 NppStatus nppiMaxIndxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.31 NppStatus nppiMaxIndxGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C3R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.9.2.32 NppStatus nppiMaxIndxGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C4R.

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10 MinMax

Primitives for computing both the minimal and the maximal values of an image.

MinMax

The functions require the device scratch buffer.

- `NppStatus nppiMinMax_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pMin`, `Npp8u *pMax`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16u *pMin`, `Npp16u *pMax`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16s *pMin`, `Npp16s *pMax`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit signed image MinMax.
- `NppStatus nppiMinMax_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32f *pMin`, `Npp32f *pMax`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image MinMax.
- `NppStatus nppiMinMax_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u aMin[3]`, `Npp8u aMax[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 8-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16u aMin[3]`, `Npp16u aMax[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16s aMin[3]`, `Npp16s aMax[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit signed image MinMax.
- `NppStatus nppiMinMax_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32f aMin[3]`, `Npp32f aMax[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 32-bit floating point image MinMax.
- `NppStatus nppiMinMax_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u aMin[3]`, `Npp8u aMax[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image MinMax ignoring alpha channel.
- `NppStatus nppiMinMax_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16u aMin[3]`, `Npp16u aMax[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image MinMax ignoring alpha channel.
- `NppStatus nppiMinMax_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp16s aMin[3]`, `Npp16s aMax[3]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed image MinMax ignoring alpha channel.

- `NppStatus nppiMinMax_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp32f aMin[3]`, `Npp32f aMax[3]`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

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

Four-channel 8-bit unsigned image MinMax.

- `NppStatus nppiMinMax_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp16u aMin[4]`, `Npp16u aMax[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit unsigned image MinMax.

- `NppStatus nppiMinMax_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp16s aMin[4]`, `Npp16s aMax[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed image MinMax.

- `NppStatus nppiMinMax_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp32f aMin[4]`, `Npp32f aMax[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image MinMax.

MinMaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMax primitives.

- `NppStatus nppiMinMaxGetBufferHostSize_8u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_8u_C1R`.

- `NppStatus nppiMinMaxGetBufferHostSize_16u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_16u_C1R`.

- `NppStatus nppiMinMaxGetBufferHostSize_16s_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_16s_C1R`.

- `NppStatus nppiMinMaxGetBufferHostSize_32f_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_32f_C1R`.

- `NppStatus nppiMinMaxGetBufferHostSize_8u_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_8u_C3R`.

- `NppStatus nppiMinMaxGetBufferHostSize_16u_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_16u_C3R`.

- `NppStatus nppiMinMaxGetBufferHostSize_16s_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_16s_C3R`.

- `NppStatus nppiMinMaxGetBufferHostSize_32f_C3R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMax_32f_C3R`.

- [NppStatus nppiMinMaxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_8u_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_16u_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_16s_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_32f_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_8u_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_16u_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_16s_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) [oSizeROI](#), int *hpBufferSize)
Buffer size for [nppiMinMax_32f_C4R](#).

7.10.1 Detailed Description

Primitives for computing both the minimal and the maximal values of an image.

7.10.2 Function Documentation

7.10.2.1 [NppStatus nppiMinMax_16s_AC4R](#) (const [Npp16s](#) * pSrc, int nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp16s](#) aMin[3], [Npp16s](#) aMax[3], [Npp8u](#) * pDeviceBuffer)

Four-channel 16-bit signed image MinMax ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

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

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

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_AC4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.2 NppStatus nppiMinMax_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s * pMin, Npp16s * pMax, Npp8u * pDeviceBuffer)

One-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

Returns:

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

7.10.2.3 NppStatus nppiMinMax_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[3], Npp16s aMax[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

Returns:

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

7.10.2.4 NppStatus nppiMinMax_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[4], Npp16s aMax[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.5 NppStatus nppiMinMax_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image MinMax ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_AC4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.6 NppStatus nppiMinMax_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u * pMin, Npp16u * pMax, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C1R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.7 NppStatus nppiMinMax_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C3R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.8 NppStatus nppiMinMax_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[4], Npp16u aMax[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.9 NppStatus nppiMinMax_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_AC4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.10 NppStatus nppiMinMax_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMin, Npp32f * pMax, Npp8u * pDeviceBuffer)

One-channel 32-bit floating point image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C1R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.11 NppStatus nppiMinMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u * pDeviceBuffer)

Three-channel 32-bit floating point image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C3R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.12 NppStatus nppiMinMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[4], Npp32f aMax[4], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.13 NppStatus nppiMinMax_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image MinMax ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_AC4R](#) to determine the minimum number of bytes required.

Returns:

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

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

One-channel 8-bit unsigned image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C1R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.15 NppStatus nppiMinMax_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)

Three-channel 8-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

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

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

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C3R](#) to determine the minimum number of bytes required.

Returns:

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

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

Four-channel 8-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

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

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

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C4R](#) to determine the minimum number of bytes required.

Returns:

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

7.10.2.17 NppStatus nppiMinMaxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16s_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.18 NppStatus nppiMinMaxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.19 NppStatus nppiMinMaxGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.20 NppStatus nppiMinMaxGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.21 NppStatus nppiMinMaxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.22 NppStatus nppiMinMaxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.23 NppStatus nppiMinMaxGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.24 NppStatus nppiMinMaxGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.25 NppStatus nppiMinMaxGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_32f_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.26 NppStatus nppiMinMaxGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.27 NppStatus nppiMinMaxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.28 NppStatus nppiMinMaxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.29 NppStatus nppiMinMaxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_8u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.10.2.30 NppStatus nppiMinMaxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.10.2.31 NppStatus nppiMinMaxGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.10.2.32 NppStatus nppiMinMaxGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11 MinMaxIndx

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

MinMaxIndx

If there are several minima and maxima in the selected region of interest, the function returns ones on the top leftmost position.

The scratch buffer is required by the functions.

- `NppStatus nppiMinMaxIndx_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.
- `NppStatus nppiMinMaxIndx_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.
- `NppStatus nppiMinMaxIndx_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` *pMinValue, `Npp16u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.
- `NppStatus nppiMinMaxIndx_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` *pMinValue, `Npp32f` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.

Masked MinMaxIndx

See [Masked Operation](#).

- `NppStatus nppiMinMaxIndx_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit unsigned image MinMaxIndx.
- `NppStatus nppiMinMaxIndx_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit signed image MinMaxIndx.

- `NppStatus nppiMinMaxIndx_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp16u` *pMinValue, `Npp16u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked one-channel 16-bit unsigned image MinMaxIndx.
- `NppStatus nppiMinMaxIndx_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp32f` *pMinValue, `Npp32f` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked one-channel 32-bit floating point image MinMaxIndx.

Channel MinMaxIndx

See [Channel-of-Interest API](#).

- `NppStatus nppiMinMaxIndx_8u_C3CR` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.
- `NppStatus nppiMinMaxIndx_8s_C3CR` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit signed image MinMaxIndx affecting only single channel.
- `NppStatus nppiMinMaxIndx_16u_C3CR` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp16u` *pMinValue, `Npp16u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.
- `NppStatus nppiMinMaxIndx_32f_C3CR` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp32f` *pMinValue, `Npp32f` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

Masked Channel MinMaxIndx

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiMinMaxIndx_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.
- `NppStatus nppiMinMaxIndx_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image MinMaxIndx affecting only single channel.

- `NppStatus nppiMinMaxIndx_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp16u *pMinValue`, `Npp16u *pMaxValue`, `NppiPoint *pMinIndex`, `NppiPoint *pMaxIndex`, `Npp8u *pDeviceBuffer`)
Masked three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.
- `NppStatus nppiMinMaxIndx_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp32f *pMinValue`, `Npp32f *pMaxValue`, `NppiPoint *pMinIndex`, `NppiPoint *pMaxIndex`, `Npp8u *pDeviceBuffer`)
Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

MinMaxIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `MinMaxIndx` primitives.

- `NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8u_C1R`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8s_C1R`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_16u_C1R`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1R` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_32f_C1R`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1MR` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8u_C1MR`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1MR` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8s_C1MR`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1MR` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_16u_C1MR`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1MR` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_32f_C1MR`.
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C3CR` (`NppiSize oSizeROI`, int `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8u_C3CR`.

- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_8s_C3CR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8s_C3CR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_16u_C3CR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_16u_C3CR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_32f_C3CR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_32f_C3CR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_8u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8u_C3CMR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_8s_C3CMR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_16u_C3CMR`.
- **NppStatus** `nppiMinMaxIndxGetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMinMaxIndx_32f_C3CMR`.

7.11.1 Detailed Description

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

7.11.2 Function Documentation

7.11.2.1 **NppStatus** `nppiMinMaxIndx_16u_C1MR` (**const** **Npp16u** `*pSrc`, **int** `nSrcStep`, **const** **Npp8u** `*pMask`, **int** `nMaskStep`, **NppiSize** `oSizeROI`, **Npp16u** `*pMinValue`, **Npp16u** `*pMaxValue`, **NppiPoint** `*pMinIndex`, **NppiPoint** `*pMaxIndex`, **Npp8u** `*pDeviceBuffer`)

Masked one-channel 16-bit unsigned image MinMaxIndx.

Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pMask` Mask-Image Pointer.
- `nMaskStep` Mask-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize_16u_C1MR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., $pMinIndex = \{0, 0\}$, $pMaxIndex = \{0, 0\}$, $pMinValue = 0$, $pMaxValue = 0$. If any of $pMinValue$, $pMaxValue$, $pMinIndex$, or $pMaxIndex$ is not needed, zero pointer must be passed correspondingly.

7.11.2.2 NppStatus nppiMinMaxIdx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize_16u_C1R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of $pMinValue$, $pMaxValue$, $pMinIndex$, or $pMaxIndex$ is not needed, zero pointer must be passed correspondingly.

7.11.2.3 NppStatus nppiMinMaxIdx_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image MinMaxIdx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.11.2.4 NppStatus nppiMinMaxIndx_16u_C3CR (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.11.2.5 NppStatus nppiMinMaxIdx_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Masked one-channel 32-bit floating point image MinMaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize_32f_C1MR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.11.2.6 NppStatus nppiMinMaxIdx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.11.2.7 NppStatus nppiMinMaxIndx_32f_C3CMR (**const Npp32f * pSrc**, **int nSrcStep**, **const Npp8u * pMask**, **int nMaskStep**, **NppiSize oSizeROI**, **int nCOI**, **Npp32f * pMinValue**, **Npp32f * pMaxValue**, **NppiPoint * pMinIndex**, **NppiPoint * pMaxIndex**, **Npp8u * pDeviceBuffer**)

Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.11.2.8 NppStatus nppiMinMaxIndx_32f_C3CR (**const Npp32f * pSrc**, **int nSrcStep**, **NppiSize oSizeROI**, **int nCOI**, **Npp32f * pMinValue**, **Npp32f * pMaxValue**, **NppiPoint * pMinIndex**, **NppiPoint * pMaxIndex**, **Npp8u * pDeviceBuffer**)

Three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_32f_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

7.11.2.9 NppStatus nppiMinMaxIdx_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Masked one-channel 8-bit signed image MinMaxIdx.

Parameters:

pSrc [Source-Image Pointer](#).

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

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

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

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., pMinIndex = {0, 0}, pMaxIndex = {0, 0}, pMinValue = 0, pMaxValue = 0. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

7.11.2.10 NppStatus nppiMinMaxIdx_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.

Parameters:

pSrc [Source-Image Pointer](#).

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

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

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C1R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.11 `NppStatus nppiMinMaxIndx_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C3CMR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., $pMinIndex = \{0, 0\}$, $pMaxIndex = \{0, 0\}$, $pMinValue = 0$, $pMaxValue = 0$. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.12 `NppStatus nppiMinMaxIndx_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

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

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

nCOI [Channel_of_Interest Number](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.13 `NppStatus nppiMinMaxIndx_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image MinMaxIndx.

Parameters:

pSrc [Source-Image Pointer](#).

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

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

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

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C1MR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.14 NppStatus nppiMinMaxIndx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pMinValue* Pointer to the minimum value.
- pMaxValue* Pointer to the maximum value.
- pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.
- pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.
- pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.15 NppStatus nppiMinMaxIndx_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)

Masked three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel_of_Interest Number.
- pMinValue* Pointer to the minimum value.
- pMaxValue* Pointer to the maximum value.
- pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.
- pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.
- pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C3CMR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.16 `NppStatus nppiMinMaxIndx_8u_C3CR (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use `nppiMinMaxIndxGetBufferHostSize_8u_C3CR` to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.11.2.17 `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMinMaxIndx_16u_C1MR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.18 `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMinMaxIndx_16u_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.19 NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_16u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.20 NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_16u_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.21 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.22 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.23 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.24 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.25 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8s_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.26 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.27 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.28 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C3CR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.29 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8u_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.30 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.31 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.11.2.32 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8u_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12 Mean

Primitives for computing the arithmetic mean of all the pixel values in an image.

Mean

Given an image $pSrc$ with width W and height H , the arithmetic mean will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

The mean functions require additional scratch buffer for computations.

- `NppStatus nppiMean_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 8-bit unsigned image Mean.
- `NppStatus nppiMean_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 16-bit signed image Mean.
- `NppStatus nppiMean_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 32-bit floating point image Mean.
- `NppStatus nppiMean_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 8-bit unsigned image Mean.
- `NppStatus nppiMean_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 16-bit signed image Mean.
- `NppStatus nppiMean_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 32-bit floating point image Mean.
- `NppStatus nppiMean_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[4])
Four-channel 8-bit unsigned image Mean.

- `NppStatus nppiMean_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[4])
Four-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[4])
Four-channel 16-bit signed image Mean.
- `NppStatus nppiMean_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[4])
Four-channel 32-bit floating point image Mean.
- `NppStatus nppiMean_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Four-channel 8-bit unsigned image Mean ignoring alpha channel.
- `NppStatus nppiMean_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Four-channel 16-bit unsigned image Mean ignoring alpha channel.
- `NppStatus nppiMean_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Four-channel 16-bit signed image Mean ignoring alpha channel.
- `NppStatus nppiMean_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Four-channel 32-bit floating point image Mean ignoring alpha channel.

Masked Mean

See [Masked Operation](#).

- `NppStatus nppiMean_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked one-channel 8-bit unsigned image Mean.
- `NppStatus nppiMean_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked one-channel 8-bit signed image Mean.
- `NppStatus nppiMean_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked one-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked one-channel 32-bit floating point image Mean.

Masked Channel Mean

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiMean_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 8-bit unsigned image Mean affecting only single channel.
- `NppStatus nppiMean_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 8-bit signed image Mean affecting only single channel.
- `NppStatus nppiMean_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 16-bit unsigned image Mean affecting only single channel.
- `NppStatus nppiMean_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 32-bit floating point image Mean affecting only single channel.

MeanGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean primitives.

- `NppStatus nppiMeanGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for nppiMean_8u_AC4R.

- `NppStatus nppiMeanGetBufferHostSize_16u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16u_AC4R.
- `NppStatus nppiMeanGetBufferHostSize_16s_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16s_AC4R.
- `NppStatus nppiMeanGetBufferHostSize_32f_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_32f_AC4R.
- `NppStatus nppiMeanGetBufferHostSize_8u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_8u_C4R.
- `NppStatus nppiMeanGetBufferHostSize_16u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16u_C4R.
- `NppStatus nppiMeanGetBufferHostSize_16s_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16s_C4R.
- `NppStatus nppiMeanGetBufferHostSize_32f_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_32f_C4R.
- `NppStatus nppiMeanGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_8u_C1MR.
- `NppStatus nppiMeanGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_8s_C1MR.
- `NppStatus nppiMeanGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16u_C1MR.
- `NppStatus nppiMeanGetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_32f_C1MR.
- `NppStatus nppiMeanGetBufferHostSize_8u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_8u_C3CMR.
- `NppStatus nppiMeanGetBufferHostSize_8s_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_8s_C3CMR.
- `NppStatus nppiMeanGetBufferHostSize_16u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_16u_C3CMR.
- `NppStatus nppiMeanGetBufferHostSize_32f_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiMean_32f_C3CMR.

7.12.1 Detailed Description

Primitives for computing the arithmetic mean of all the pixel values in an image.

7.12.2 Function Documentation

7.12.2.1 `NppStatus nppiMean_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Four-channel 16-bit signed image Mean ignoring alpha channel.

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](#) Use [nppiMeanGetBufferHostSize_16s_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.2 `NppStatus nppiMean_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 16-bit signed image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

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

7.12.2.3 `NppStatus nppiMean_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 16-bit signed image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.4 NppStatus nppiMean_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])

Four-channel 16-bit signed image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.5 NppStatus nppiMean_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])

Four-channel 16-bit unsigned image Mean ignoring alpha channel.

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](#) Use [nppiMeanGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.6 NppStatus nppiMean_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)

Masked one-channel 16-bit unsigned image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C1MR](#) to determine the minium number of bytes required.
pMean Pointer to the computed mean result.

Returns:

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

7.12.2.7 NppStatus nppiMean_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)

One-channel 16-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.
pMean Pointer to the computed mean result.

Returns:

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

7.12.2.8 NppStatus nppiMean_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)

Masked three-channel 16-bit unsigned image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.
pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or [NPP_COI_ERROR](#) if an invalid channel of interest is specified.

7.12.2.9 NppStatus nppiMean_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])

Three-channel 16-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.10 NppStatus nppiMean_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])

Four-channel 16-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.11 NppStatus nppiMean_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])

Four-channel 32-bit floating point image Mean ignoring alpha channel.

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](#) Use [nppiMeanGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.12.2.12 `NppStatus nppiMean_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 32-bit floating point image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.12.2.13 `NppStatus nppiMean_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 32-bit floating point image Mean.

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](#) Use [nppiMeanGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.12.2.14 `NppStatus nppiMean_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 32-bit floating point image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [NPP_NOT_EVEN_STEP_ERROR](#) if an invalid floating-point image is specified, or [NPP_COI_ERROR](#) if an invalid channel of interest is specified.

7.12.2.15 `NppStatus nppiMean_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 32-bit floating point image Mean.

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](#) Use [nppiMeanGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [NPP_NOT_EVEN_STEP_ERROR](#) if an invalid floating-point image is specified.

7.12.2.16 `NppStatus nppiMean_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 32-bit floating point image Mean.

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](#) Use [nppiMeanGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.12.2.17 `NppStatus nppiMean_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 8-bit signed image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

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

7.12.2.18 `NppStatus nppiMean_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 8-bit signed image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8s_C3CMR](#) to determine the minimum number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.12.2.19 NppStatus nppiMean_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])

Four-channel 8-bit unsigned image Mean ignoring alpha channel.

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](#) Use [nppiMeanGetBufferHostSize_8u_AC4R](#) to determine the minimum number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.20 NppStatus nppiMean_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)

Masked one-channel 8-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C1MR](#) to determine the minimum number of bytes required.

pMean Pointer to the computed mean result.

Returns:

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

7.12.2.21 `NppStatus nppiMean_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 8-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

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

7.12.2.22 `NppStatus nppiMean_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 8-bit unsigned image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.12.2.23 `NppStatus nppiMean_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 8-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.24 `NppStatus nppiMean_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 8-bit unsigned image Mean.

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](#) Use [nppiMeanGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

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

7.12.2.25 `NppStatus nppiMeanGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.26 `NppStatus nppiMeanGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.27 NppStatus nppiMeanGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.28 NppStatus nppiMeanGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.29 NppStatus nppiMeanGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.30 NppStatus nppiMeanGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.31 NppStatus nppiMeanGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.32 NppStatus nppiMeanGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.33 NppStatus nppiMeanGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.34 NppStatus nppiMeanGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.35 NppStatus nppiMeanGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.36 NppStatus nppiMeanGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.37 NppStatus nppiMeanGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.38 NppStatus nppiMeanGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.39 NppStatus nppiMeanGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.40 NppStatus nppiMeanGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_32f_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.41 NppStatus nppiMeanGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8s_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.42 NppStatus nppiMeanGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8s_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.43 NppStatus nppiMeanGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.44 NppStatus nppiMeanGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8u_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.45 NppStatus nppiMeanGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.46 NppStatus nppiMeanGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.47 NppStatus nppiMeanGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8u_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.12.2.48 NppStatus nppiMeanGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_8u_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13 Mean_StdDev

Primitives for computing both the arithmetic mean and the standard deviation of an image.

Mean_StdDev

Given an image $pSrc$ with width W and height H , the mean and the standard deviation will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

$$StdDev = \sqrt{\frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (pSrc(j, i) - Mean)^2}$$

The Mean_StdDev primitives require additional scratch buffer for computations.

- `NppStatus nppiMean_StdDev_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 8-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 8-bit signed image Mean_StdDev.
- `NppStatus nppiMean_StdDev_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 16-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 32-bit floating point image Mean_StdDev.

Masked Mean_StdDev

See [Masked Operation](#).

- `NppStatus nppiMean_StdDev_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 8-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 8-bit signed image Mean_StdDev.
- `NppStatus nppiMean_StdDev_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 16-bit unsigned image Mean_StdDev.

- `NppStatus nppiMean_StdDev_32f_C1MR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Masked one-channel 32-bit floating point image Mean_StdDev.

Channel Mean_StdDev

See [Channel-of-Interest API](#).

- `NppStatus nppiMean_StdDev_8u_C3CR` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Three-channel 8-bit unsigned image Mean_StdDev affecting only single channel.
- `NppStatus nppiMean_StdDev_8s_C3CR` (const `Npp8s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Three-channel 8-bit signed image Mean_StdDev affecting only single channel.
- `NppStatus nppiMean_StdDev_16u_C3CR` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Three-channel 16-bit unsigned image Mean_StdDev affecting only single channel.
- `NppStatus nppiMean_StdDev_32f_C3CR` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Three-channel 32-bit floating point image Mean_StdDev affecting only single channel.

Masked Channel Mean_StdDev

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiMean_StdDev_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Masked three-channel 8-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_8s_C3CMR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Masked three-channel 8-bit signed image Mean_StdDev.
- `NppStatus nppiMean_StdDev_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Masked three-channel 16-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)
Masked three-channel 32-bit floating point image Mean_StdDev.

MeanStdDevGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8u_C1R`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8s_C1R`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_16u_C1R`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_32f_C1R`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8u_C1MR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8s_C1MR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_16u_C1MR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_32f_C1MR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8u_C3CR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_8s_C3CR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_16u_C3CR`.
- `NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiMean_StdDev_32f_C3CR`.

- **NppStatus** `nppiMeanStdDevGetBufferHostSize_8u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMean_StdDev_8u_C3CMR`.
- **NppStatus** `nppiMeanStdDevGetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMean_StdDev_8s_C3CMR`.
- **NppStatus** `nppiMeanStdDevGetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMean_StdDev_16u_C3CMR`.
- **NppStatus** `nppiMeanStdDevGetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiMean_StdDev_32f_C3CMR`.

7.13.1 Detailed Description

Primitives for computing both the arithmetic mean and the standard deviation of an image.

7.13.2 Function Documentation

7.13.2.1 **NppStatus** `nppiMean_StdDev_16u_C1MR` (**const** **Npp16u** `*pSrc`, **int** `nSrcStep`, **const** **Npp8u** `*pMask`, **int** `nMaskStep`, **NppiSize** `oSizeROI`, **Npp8u** `*pDeviceBuffer`, **Npp64f** `*pMean`, **Npp64f** `*pStdDev`)

Masked one-channel 16-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
 Use `nppiMeanStdDevGetBufferHostSize_16u_C1MR` to determine the minimum number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.2 NppStatus nppiMean_StdDev_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

One-channel 16-bit unsigned image Mean_StdDev.

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](#) Use [nppiMeanStdDevGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.3 NppStatus nppiMean_StdDev_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked three-channel 16-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.4 NppStatus nppiMean_StdDev_16u_C3CR (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Three-channel 16-bit unsigned image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_16u_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.5 NppStatus nppiMean_StdDev_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked one-channel 32-bit floating point image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_32f_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.13.2.6 NppStatus nppiMean_StdDev_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

One-channel 32-bit floating point image Mean_StdDev.

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](#) Use [nppiMeanStdDevGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.13.2.7 NppStatus nppiMean_StdDev_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked three-channel 32-bit floating point image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.8 NppStatus nppiMean_StdDev_32f_C3CR (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Three-channel 32-bit floating point image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_32f_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.13.2.9 NppStatus nppiMean_StdDev_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked one-channel 8-bit signed image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.10 `NppStatus nppiMean_StdDev_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

One-channel 8-bit signed image Mean_StdDev.

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](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.11 `NppStatus nppiMean_StdDev_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

Masked three-channel 8-bit signed image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.12 NppStatus nppiMean_StdDev_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Three-channel 8-bit signed image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.13 NppStatus nppiMean_StdDev_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked one-channel 8-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.14 NppStatus nppiMean_StdDev_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

One-channel 8-bit unsigned image Mean_StdDev.

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](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

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

7.13.2.15 NppStatus nppiMean_StdDev_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)

Masked three-channel 8-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.16 `NppStatus nppiMean_StdDev_8u_C3CR (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

Three-channel 8-bit unsigned image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use `nppiMeanStdDevGetBufferHostSize_8u_C3CR` to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.13.2.17 `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_16u_C1MR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.18 `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_16u_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.19 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_16u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.20 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_16u_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.21 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.22 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.23 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.24 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.25 NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8s_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.26 `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_8s_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.27 `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_8s_C3CMR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.28 `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_8s_C3CR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.29 `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiMean_StdDev_8u_C1MR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.30 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.31 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8u_C3CMR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.13.2.32 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8u_C3CR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.14 Image Norms

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

Modules

- [Norm_Inf](#)
Primitives for computing the infinity norm of an image.
- [Norm_L1](#)
Primitives for computing the L1 norm of an image.
- [Norm_L2](#)
Primitives for computing the L2 norm of an image.
- [NormDiff_Inf](#)
Primitives for computing the infinity norm of difference of pixels between two images.
- [NormDiff_L1](#)
Primitives for computing the L1 norm of difference of pixels between two images.
- [NormDiff_L2](#)
Primitives for computing the L2 norm of difference of pixels between two images.
- [NormRel_Inf](#)
Primitives for computing the relative error of infinity norm between two images.
- [NormRel_L1](#)
Primitives for computing the relative error of L1 norm between two images.
- [NormRel_L2](#)
Primitives for computing the relative error of L2 norm between two images.

7.14.1 Detailed Description

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

Given an image $pSrc$ with width W and height H ,

1. The infinity norm (`Norm_Inf`) is defined as the largest absolute pixel value of the image.
2. The L1 norm (`Norm_L1`) is defined as the sum of the absolute pixel value of the image, i.e.,

$$Norm_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|$$

3. The L2 norm (Norm_L2) is defined as the square root of the sum of the squared absolute pixel value of the image, i.e.,

$$Norm_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|^2}$$

Given two images $pSrc1$ and $pSrc2$ both with width W and height H ,

1. The infinity norm of difference (NormDiff_Inf) is defined as the largest absolute difference between pixels of two images.
2. The L1 norm of difference (NormDiff_L1) is defined as the sum of the absolute difference between pixels of two images, i.e.,

$$NormDiff_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

3. The L2 norm of difference (NormDiff_L2) is defined as the squared root of the sum of the squared absolute difference between pixels of two images, i.e.,

$$NormDiff_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|^2}$$

Given two images $pSrc1$ and $pSrc2$ both with width W and height H ,

1. The relative error for the infinity norm of difference (NormRel_Inf) is defined as NormDiff_Inf divided by the infinity norm of the second image, i.e.,

$$NormRel_Inf = \frac{NormDiff_Inf}{Norm_Inf_{src2}}$$

2. The relative error for the L1 norm of difference (NormRel_L1) is defined as NormDiff_L1 divided by the L1 norm of the second image, i.e.,

$$NormRel_L1 = \frac{NormDiff_L1}{Norm_L1_{src2}}$$

3. The relative error for the L2 norm of difference (NormRel_L2) is defined as NormDiff_L2 divided by the L2 norm of the second image, i.e.,

$$NormRel_L2 = \frac{NormDiff_L2}{Norm_L2_{src2}}$$

The norm functions require the addition device scratch buffer for the computations.

7.15 Norm_Inf

Primitives for computing the infinity norm of an image.

Basic Norm_Inf

- **NppStatus nppiNorm_Inf_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image Norm_Inf.
- **NppStatus nppiNorm_Inf_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image Norm_Inf.
- **NppStatus nppiNorm_Inf_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_Inf ignoring alpha channel.
- **NppStatus nppiNorm_Inf_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_Inf ignoring alpha channel.

- `NppStatus nppiNorm_Inf_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_Inf ignoring alpha channel.
- `NppStatus nppiNorm_Inf_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_Inf ignoring alpha channel.
- `NppStatus nppiNorm_Inf_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_Inf.
- `NppStatus nppiNorm_Inf_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_Inf.

Masked Norm_Inf

See [Masked Operation](#).

- `NppStatus nppiNorm_Inf_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit signed image Norm_Inf.
- `NppStatus nppiNorm_Inf_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 16-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 32-bit floating point image Norm_Inf.

Masked Channel Norm_Inf

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_Inf_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image Norm_Inf affecting only single channel.
- `NppStatus nppiNorm_Inf_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image Norm_Inf affecting only single channel.
- `NppStatus nppiNorm_Inf_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image Norm_Inf affecting only single channel.
- `NppStatus nppiNorm_Inf_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image Norm_Inf affecting only single channel.

NormInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_Inf primitives.

- `NppStatus nppiNormInfGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_8u_C1R`.
- `NppStatus nppiNormInfGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_16u_C1R`.
- `NppStatus nppiNormInfGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_16s_C1R`.
- `NppStatus nppiNormInfGetBufferHostSize_32s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_32s_C1R`.
- `NppStatus nppiNormInfGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_32f_C1R`.
- `NppStatus nppiNormInfGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_8u_C1MR`.
- `NppStatus nppiNormInfGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_8s_C1MR`.
- `NppStatus nppiNormInfGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_16u_C1MR`.
- `NppStatus nppiNormInfGetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_Inf_32f_C1MR`.
- `NppStatus nppiNormInfGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for nppiNorm_Inf_8u_C3R.

- **NppStatus nppiNormInfGetBufferHostSize_16u_C3R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_C3R.
- **NppStatus nppiNormInfGetBufferHostSize_16s_C3R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16s_C3R.
- **NppStatus nppiNormInfGetBufferHostSize_32f_C3R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_C3R.
- **NppStatus nppiNormInfGetBufferHostSize_8u_AC4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8u_AC4R.
- **NppStatus nppiNormInfGetBufferHostSize_16u_AC4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_AC4R.
- **NppStatus nppiNormInfGetBufferHostSize_16s_AC4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16s_AC4R.
- **NppStatus nppiNormInfGetBufferHostSize_32f_AC4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_AC4R.
- **NppStatus nppiNormInfGetBufferHostSize_8u_C4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8u_C4R.
- **NppStatus nppiNormInfGetBufferHostSize_16u_C4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_C4R.
- **NppStatus nppiNormInfGetBufferHostSize_16s_C4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16s_C4R.
- **NppStatus nppiNormInfGetBufferHostSize_32f_C4R** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_C4R.
- **NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8u_C3CMR.
- **NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8s_C3CMR.
- **NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_C3CMR.
- **NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR** (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_C3CMR.

7.15.1 Detailed Description

Primitives for computing the infinity norm of an image.

7.15.2 Function Documentation

7.15.2.1 `NppStatus nppiNorm_Inf_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Norm_Inf ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.2 `NppStatus nppiNorm_Inf_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.3 `NppStatus nppiNorm_Inf_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.4 NppStatus nppiNorm_Inf_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.5 NppStatus nppiNorm_Inf_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_Inf ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.6 NppStatus nppiNorm_Inf_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

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

7.15.2.7 NppStatus nppiNorm_Inf_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.8 NppStatus nppiNorm_Inf_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_Inf affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.15.2.9 NppStatus nppiNorm_Inf_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.10 NppStatus nppiNorm_Inf_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.11 `NppStatus nppiNorm_Inf_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm_Inf ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.12 `NppStatus nppiNorm_Inf_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

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

7.15.2.13 `NppStatus nppiNorm_Inf_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.14 `NppStatus nppiNorm_Inf_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm_Inf affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.15.2.15 `NppStatus nppiNorm_Inf_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.16 `NppStatus nppiNorm_Inf_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.17 `NppStatus nppiNorm_Inf_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32s_C1R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.18 `NppStatus nppiNorm_Inf_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

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

7.15.2.19 `NppStatus nppiNorm_Inf_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_Inf affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

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

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

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

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.15.2.20 `NppStatus nppiNorm_Inf_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm_Inf ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

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

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

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.21 `NppStatus nppiNorm_Inf_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

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

7.15.2.22 `NppStatus nppiNorm_Inf_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.23 `NppStatus nppiNorm_Inf_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_Inf affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.15.2.24 NppStatus nppiNorm_Inf_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.25 NppStatus nppiNorm_Inf_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

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

7.15.2.26 NppStatus nppiNormInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16s_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.27 NppStatus nppiNormInfGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16s_C1R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.28 NppStatus nppiNormInfGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16s_C3R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.29 NppStatus nppiNormInfGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16s_C4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.30 NppStatus nppiNormInfGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_AC4R](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.31 NppStatus nppiNormInfGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C1MR](#).

Parameters:

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

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

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.32 NppStatus nppiNormInfGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.33 NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.34 NppStatus nppiNormInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.35 NppStatus nppiNormInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.36 NppStatus nppiNormInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.37 NppStatus nppiNormInfGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.38 NppStatus nppiNormInfGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.39 NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.40 NppStatus nppiNormInfGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.41 NppStatus nppiNormInfGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.42 NppStatus nppiNormInfGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.43 NppStatus nppiNormInfGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.44 NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.45 NppStatus nppiNormInfGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.46 NppStatus nppiNormInfGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.47 `NppStatus nppiNormInfGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_Inf_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_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.48 `NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_Inf_8u_C3CMR`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.49 `NppStatus nppiNormInfGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_Inf_8u_C3R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.15.2.50 `NppStatus nppiNormInfGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_Inf_8u_C4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16 Norm_L1

Primitives for computing the L1 norm of an image.

Basic Norm_L1

- `NppStatus nppiNorm_L1_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image Norm_L1.
- `NppStatus nppiNorm_L1_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 32-bit floating point image Norm_L1.
- `NppStatus nppiNorm_L1_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit signed image Norm_L1 ignoring alpha channel.

- `NppStatus nppiNorm_L1_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image Norm_L1.

Masked Norm_L1

See [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C1MR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
Masked one-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_8s_C1MR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
Masked one-channel 8-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C1MR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
Masked one-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C1MR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
Masked one-channel 32-bit floating point image Norm_L1.

Masked Channel Norm_L1

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
Masked three-channel 8-bit unsigned image Norm_L1 affecting only single channel.

- `NppStatus nppiNorm_L1_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image Norm_L1 affecting only single channel.
- `NppStatus nppiNorm_L1_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image Norm_L1 affecting only single channel.
- `NppStatus nppiNorm_L1_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image Norm_L1 affecting only single channel.

NormL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L1 primitives.

- `NppStatus nppiNormL1GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_8u_C1R.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_16u_C1R.
- `NppStatus nppiNormL1GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_16s_C1R.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_32f_C1R.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_8u_C1MR.
- `NppStatus nppiNormL1GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_8s_C1MR.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_16u_C1MR.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_32f_C1MR.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_8u_C3R.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_16u_C3R.
- `NppStatus nppiNormL1GetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L1_16s_C3R.

- `NppStatus nppiNormL1GetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_32f_C3R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_8u_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_16u_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_16s_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_32f_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_8u_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_16u_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_16s_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_32f_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_8u_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_8s_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_8s_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_16u_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L1_32f_C3CMR`.

7.16.1 Detailed Description

Primitives for computing the L1 norm of an image.

7.16.2 Function Documentation

7.16.2.1 `NppStatus nppiNorm_L1_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormL1GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.2 `NppStatus nppiNorm_L1_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormL1GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.3 `NppStatus nppiNorm_L1_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.4 NppStatus nppiNorm_L1_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.5 NppStatus nppiNorm_L1_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.6 NppStatus nppiNorm_L1_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.7 NppStatus nppiNorm_L1_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.8 NppStatus nppiNorm_L1_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.16.2.9 NppStatus nppiNorm_L1_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.10 NppStatus nppiNorm_L1_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.11 NppStatus nppiNorm_L1_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image Norm_L1 ignoring alpha channel.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNorm* Array that contains the norm values of Three-channels.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.12 NppStatus nppiNorm_L1_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 32-bit floating point image Norm_L1.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNorm* Pointer to the norm value.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.13 NppStatus nppiNorm_L1_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

One-channel 32-bit floating point image Norm_L1.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNorm* Pointer to the norm value.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.14 `NppStatus nppiNorm_L1_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.16.2.15 `NppStatus nppiNorm_L1_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.16 `NppStatus nppiNorm_L1_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.17 `NppStatus nppiNorm_L1_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.18 `NppStatus nppiNorm_L1_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.16.2.19 `NppStatus nppiNorm_L1_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.20 `NppStatus nppiNorm_L1_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.21 `NppStatus nppiNorm_L1_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.22 `NppStatus nppiNorm_L1_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_L1 affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.16.2.23 `NppStatus nppiNorm_L1_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.24 NppStatus nppiNorm_L1_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.16.2.25 NppStatus nppiNormL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.26 NppStatus nppiNormL1GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16s_C1R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.27 NppStatus nppiNormL1GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.28 NppStatus nppiNormL1GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.29 NppStatus nppiNormL1GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.30 NppStatus nppiNormL1GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.31 NppStatus nppiNormL1GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.32 NppStatus nppiNormL1GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.33 NppStatus nppiNormL1GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.34 NppStatus nppiNormL1GetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.35 NppStatus nppiNormL1GetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.36 NppStatus nppiNormL1GetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.37 NppStatus nppiNormL1GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.38 NppStatus nppiNormL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.39 NppStatus nppiNormL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.40 NppStatus nppiNormL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.41 NppStatus nppiNormL1GetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.42 NppStatus nppiNormL1GetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.43 NppStatus nppiNormL1GetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.44 NppStatus nppiNormL1GetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.45 NppStatus nppiNormL1GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.46 NppStatus nppiNormL1GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.16.2.47 NppStatus nppiNormL1GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.16.2.48 NppStatus nppiNormL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiNorm_L1_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17 Norm_L2

Primitives for computing the L2 norm of an image.

Basic Norm_L2

Computes the L2 norm of an image.

- `NppStatus nppiNorm_L2_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit signed image Norm_L2.
- `NppStatus nppiNorm_L2_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image Norm_L2.
- `NppStatus nppiNorm_L2_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 8-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 16-bit signed image Norm_L2.
- `NppStatus nppiNorm_L2_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Three-channel 32-bit floating point image Norm_L2.
- `NppStatus nppiNorm_L2_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image Norm_L2 ignoring alpha channel.
- `NppStatus nppiNorm_L2_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image Norm_L2 ignoring alpha channel.
- `NppStatus nppiNorm_L2_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed image Norm_L2 ignoring alpha channel.

- `NppStatus nppiNorm_L2_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f` `aNorm[3]`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image Norm_L2 ignoring alpha channel.

- `NppStatus nppiNorm_L2_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f` `aNorm[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f` `aNorm[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f` `aNorm[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp64f` `aNorm[4]`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image Norm_L2.

Masked Norm_L2

See [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C1MR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)

Masked one-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_8s_C1MR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)

Masked one-channel 8-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C1MR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)

Masked one-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C1MR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)

Masked one-channel 32-bit floating point image Norm_L2.

Masked Channel Norm_L2

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)

Masked three-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 32-bit floating point image Norm_L2.

NormL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L2 primitives.

- `NppStatus nppiNormL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_8u_C1R.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_16u_C1R.
- `NppStatus nppiNormL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_16s_C1R.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_32f_C1R.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_8u_C1MR.
- `NppStatus nppiNormL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_8s_C1MR.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_16u_C1MR.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_32f_C1MR.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_8u_C3R.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_L2_16u_C3R.
- `NppStatus nppiNormL2GetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16s_C3R`.

- `NppStatus nppiNormL2GetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C3R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_16s_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_16s_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_8s_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_8s_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C3CMR`.

7.17.1 Detailed Description

Primitives for computing the L2 norm of an image.

7.17.2 Function Documentation

7.17.2.1 `NppStatus nppiNorm_L2_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Norm_L2 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.2 `NppStatus nppiNorm_L2_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.3 `NppStatus nppiNorm_L2_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.4 NppStatus nppiNorm_L2_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.5 NppStatus nppiNorm_L2_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L2 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.6 NppStatus nppiNorm_L2_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.7 NppStatus nppiNorm_L2_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.8 NppStatus nppiNorm_L2_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.17.2.9 `NppStatus nppiNorm_L2_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.10 `NppStatus nppiNorm_L2_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.11 `NppStatus nppiNorm_L2_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm_L2 ignoring alpha channel.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNorm* Array that contains the norm values of Three-channels.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.12 `NppStatus nppiNorm_L2_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm_L2.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNorm* Pointer to the norm value.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4.

7.17.2.13 `NppStatus nppiNorm_L2_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm_L2.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNorm* Pointer to the norm value.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.14 `NppStatus nppiNorm_L2_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.17.2.15 `NppStatus nppiNorm_L2_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.16 `NppStatus nppiNorm_L2_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.17 `NppStatus nppiNorm_L2_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.18 `NppStatus nppiNorm_L2_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.17.2.19 `NppStatus nppiNorm_L2_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm_L2 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.20 `NppStatus nppiNorm_L2_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.21 `NppStatus nppiNorm_L2_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.22 `NppStatus nppiNorm_L2_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.17.2.23 `NppStatus nppiNorm_L2_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.24 `NppStatus nppiNorm_L2_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormL2GetBufferHostSize_8u_C4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.17.2.25 `NppStatus nppiNormL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_L2_16s_AC4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.26 `NppStatus nppiNormL2GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for `nppiNorm_L2_16s_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.27 NppStatus nppiNormL2GetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.28 NppStatus nppiNormL2GetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.29 NppStatus nppiNormL2GetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.30 NppStatus nppiNormL2GetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.31 NppStatus nppiNormL2GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.32 NppStatus nppiNormL2GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.33 NppStatus nppiNormL2GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.34 NppStatus nppiNormL2GetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.35 NppStatus nppiNormL2GetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.36 NppStatus nppiNormL2GetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.37 NppStatus nppiNormL2GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.38 NppStatus nppiNormL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.39 NppStatus nppiNormL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.40 NppStatus nppiNormL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.41 NppStatus nppiNormL2GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.42 NppStatus nppiNormL2GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.43 NppStatus nppiNormL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.44 NppStatus nppiNormL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.45 NppStatus nppiNormL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.46 NppStatus nppiNormL2GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.17.2.47 NppStatus nppiNormL2GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.17.2.48 NppStatus nppiNormL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiNorm_L2_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18 NormDiff_Inf

Primitives for computing the infinity norm of difference of pixels between two images.

Basic NormDiff_Inf

- `NppStatus nppiNormDiff_Inf_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16s_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_8u_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_Inf ignoring alpha channel.
- `NppStatus nppiNormDiff_Inf_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_Inf ignoring alpha channel.
- `NppStatus nppiNormDiff_Inf_16s_AC4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_Inf ignoring alpha channel.

- `NppStatus nppiNormDiff_Inf_32f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_Inf ignoring alpha channel.
- `NppStatus nppiNormDiff_Inf_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[4], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16s_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aNormDiff[4], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_Inf.

Masked NormDiff_Inf

See [Masked Operation](#).

- `NppStatus nppiNormDiff_Inf_8u_C1MR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit unsigned images NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_8s_C1MR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit signed images NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_16u_C1MR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked one-channel 16-bit unsigned images NormDiff_Inf.
- `NppStatus nppiNormDiff_Inf_32f_C1MR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked one-channel 32-bit floating point images NormDiff_Inf.

Masked Channel Mean

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormDiff_Inf_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormDiff_Inf affecting only single channel.
- `NppStatus nppiNormDiff_Inf_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image NormDiff_Inf affecting only single channel.
- `NppStatus nppiNormDiff_Inf_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormDiff_Inf affecting only single channel.
- `NppStatus nppiNormDiff_Inf_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormDiff_Inf affecting only single channel.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- `NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_8u_C1R`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_16u_C1R`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_16s_C1R`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_32f_C1R`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_8u_C1MR`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNormDiff_Inf_8s_C1MR`.
- `NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16u_C1MR.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_32f_C1MR.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_8u_C3R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16u_C3R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16s_C3R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_32f_C3R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_8u_C4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16u_C4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16s_C4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_32f_C4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_8u_AC4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16u_AC4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_16s_AC4R.

- **NppStatus** nppiNormDiffInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int *hpBufferSize)

Buffer size for nppiNormDiff_Inf_32f_AC4R.

- **NppStatus** `nppiNormDiffInfGetBufferHostSize_8u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiNormDiff_Inf_8u_C3CMR`.
- **NppStatus** `nppiNormDiffInfGetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiNormDiff_Inf_8s_C3CMR`.
- **NppStatus** `nppiNormDiffInfGetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiNormDiff_Inf_16u_C3CMR`.
- **NppStatus** `nppiNormDiffInfGetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiNormDiff_Inf_32f_C3CMR`.

7.18.1 Detailed Description

Primitives for computing the infinity norm of difference of pixels between two images.

7.18.2 Function Documentation

7.18.2.1 **NppStatus** `nppiNormDiff_Inf_16s_AC4R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `aNormDiff[3]`, **Npp8u** `*pDeviceBuffer`)

Four-channel 16-bit signed image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use `nppiNormDiffInfGetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.2 NppStatus nppiNormDiff_Inf_16s_C1R (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pNormDiff*, Npp8u * *pDeviceBuffer*)

One-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.3 NppStatus nppiNormDiff_Inf_16s_C3R (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormDiff*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.4 NppStatus nppiNormDiff_Inf_16s_C4R (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormDiff*[4], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.5 `NppStatus nppiNormDiff_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.6 `NppStatus nppiNormDiff_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned images NormDiff_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.7 `NppStatus nppiNormDiff_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.8 `NppStatus nppiNormDiff_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.18.2.9 NppStatus nppiNormDiff_Inf_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.10 NppStatus nppiNormDiff_Inf_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.11 `NppStatus nppiNormDiff_Inf_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.12 `NppStatus nppiNormDiff_Inf_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point images NormDiff_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.13 `NppStatus nppiNormDiff_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.14 `NppStatus nppiNormDiff_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.15 `NppStatus nppiNormDiff_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormDiffInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.16 `NppStatus nppiNormDiff_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormDiffInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.18.2.17 `NppStatus nppiNormDiff_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed images NormDiff_Inf.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.18 NppStatus nppiNormDiff_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)

Masked three-channel 8-bit signed image NormDiff_Inf affecting only single channel.

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.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel_of_Interest Number.
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.18.2.19 NppStatus nppiNormDiff_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image NormDiff_Inf ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormDiff* Array that contains computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.20 `NppStatus nppiNormDiff_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned images NormDiff_Inf.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.21 `NppStatus nppiNormDiff_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.22 `NppStatus nppiNormDiff_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.18.2.23 `NppStatus nppiNormDiff_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.24 `NppStatus nppiNormDiff_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.18.2.25 `NppStatus nppiNormDiffInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiNormDiff_Inf_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.26 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.27 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.28 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.29 NppStatus nppiNormDiffInfGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.30 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.31 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.32 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.33 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.34 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.35 NppStatus nppiNormDiffInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.36 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.37 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.38 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.39 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.40 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.41 NppStatus nppiNormDiffInfGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.42 NppStatus nppiNormDiffInfGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.43 NppStatus nppiNormDiffInfGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.44 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.45 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.46 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.47 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.18.2.48 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19 NormDiff_L1

Primitives for computing the L1 norm of difference of pixels between two images.

Basic NormDiff_L1

- **NppStatus nppiNormDiff_L1_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L1 ignoring alpha channel.
- **NppStatus nppiNormDiff_L1_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L1 ignoring alpha channel.
- **NppStatus nppiNormDiff_L1_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L1 ignoring alpha channel.

- [NppStatus nppiNormDiff_L1_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L1 ignoring alpha channel.
- [NppStatus nppiNormDiff_L1_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L1.

Masked NormDiff_L1

See [Masked Operation](#).

- [NppStatus nppiNormDiff_L1_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit unsigned image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit signed image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 16-bit unsigned image NormDiff_L1.
- [NppStatus nppiNormDiff_L1_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 32-bit floating point image NormDiff_L1.

Masked Channel NormDiff_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** `nppiNormDiff_L1_8u_C3CMR` (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormDiff_L1 affecting only single channel.
- **NppStatus** `nppiNormDiff_L1_8s_C3CMR` (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit signed image NormDiff_L1 affecting only single channel.
- **NppStatus** `nppiNormDiff_L1_16u_C3CMR` (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormDiff_L1 affecting only single channel.
- **NppStatus** `nppiNormDiff_L1_32f_C3CMR` (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormDiff_L1 affecting only single channel.

NormDiffL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L1 primitives.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1R.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1R.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16s_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C1R.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1R.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1MR.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8s_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C1MR.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1MR.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_C1MR` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1MR.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C3R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C3R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C3R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C3R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16s_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_AC4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_AC4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16s_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_AC4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_AC4R.

- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_8u_C3CMR`.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_8s_C3CMR`.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_16u_C3CMR`.
- **NppStatus** `nppiNormDiffL1GetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_32f_C3CMR`.

7.19.1 Detailed Description

Primitives for computing the L1 norm of difference of pixels between two images.

7.19.2 Function Documentation

7.19.2.1 **NppStatus** `nppiNormDiff_L1_16s_AC4R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `aNormDiff[3]`, **Npp8u** `*pDeviceBuffer`)

Four-channel 16-bit signed image NormDiff_L1 ignoring alpha channel.

Parameters:

`pSrc1` [Source-Image Pointer](#).

`nSrc1Step` [Source-Image Line Step](#).

`pSrc2` [Source-Image Pointer](#).

`nSrc2Step` [Source-Image Line Step](#).

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`aNormDiff` Array that contains computed Inf-norm of differences.

`pDeviceBuffer` [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
 Use `nppiNormDiffL1GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.2 **NppStatus** `nppiNormDiff_L1_16s_C1R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `*pNormDiff`, **Npp8u** `*pDeviceBuffer`)

One-channel 16-bit signed image NormDiff_L1.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.3 `NppStatus nppiNormDiff_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff_L1.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormDiff* Array that contains computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.4 `NppStatus nppiNormDiff_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff_L1.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.5 NppStatus nppiNormDiff_L1_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.6 NppStatus nppiNormDiff_L1_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image NormDiff_L1.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.7 NppStatus nppiNormDiff_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.8 NppStatus nppiNormDiff_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormDiff_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.19.2.9 `NppStatus nppiNormDiff_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.10 `NppStatus nppiNormDiff_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.11 `NppStatus nppiNormDiff_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L1 ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormDiff* Array that contains computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.12 `NppStatus nppiNormDiff_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff_L1.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.13 `NppStatus nppiNormDiff_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_L1.

Parameters:

- pSrc1* Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.14 `NppStatus nppiNormDiff_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.15 `NppStatus nppiNormDiff_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.16 `NppStatus nppiNormDiff_L1_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.19.2.17 `NppStatus nppiNormDiff_L1_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff_L1.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.18 `NppStatus nppiNormDiff_L1_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.19.2.19 `NppStatus nppiNormDiff_L1_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.20 `NppStatus nppiNormDiff_L1_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff_L1.

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](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.21 `NppStatus nppiNormDiff_L1_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.22 `NppStatus nppiNormDiff_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormDiffL1GetBufferHostSize_8u_C3CMR` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.19.2.23 `NppStatus nppiNormDiff_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormDiffL1GetBufferHostSize_8u_C3R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.24 `NppStatus nppiNormDiff_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.19.2.25 `NppStatus nppiNormDiffL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_AC4R.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.26 `NppStatus nppiNormDiffL1GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C1R.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.27 NppStatus nppiNormDiffL1GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.28 NppStatus nppiNormDiffL1GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.29 NppStatus nppiNormDiffL1GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.30 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.31 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.32 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.33 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.34 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.35 NppStatus nppiNormDiffL1GetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.36 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.37 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.38 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.39 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.40 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.41 NppStatus nppiNormDiffL1GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.42 NppStatus nppiNormDiffL1GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.43 NppStatus nppiNormDiffL1GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.44 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.45 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.46 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.47 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.19.2.48 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20 NormDiff_L2

Primitives for computing the L2 norm of difference of pixels between two images.

Basic NormDiff_L2

- **NppStatus nppiNormDiff_L2_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L2 ignoring alpha channel.
- **NppStatus nppiNormDiff_L2_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L2 ignoring alpha channel.
- **NppStatus nppiNormDiff_L2_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L2 ignoring alpha channel.

- **NppStatus** [nppiNormDiff_L2_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L2 ignoring alpha channel.
- **NppStatus** [nppiNormDiff_L2_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L2.

Masked NormDiff_L2

See [Masked Operation](#).

- **NppStatus** [nppiNormDiff_L2_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit signed image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus** [nppiNormDiff_L2_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 32-bit floating point image NormDiff_L2.

Masked Channel NormDiff_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormDiff_L2_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormDiff_L2 affecting only single channel.
- `NppStatus nppiNormDiff_L2_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image NormDiff_L2 affecting only single channel.
- `NppStatus nppiNormDiff_L2_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormDiff_L2 affecting only single channel.
- `NppStatus nppiNormDiff_L2_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormDiff, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormDiff_L2 affecting only single channel.

NormDiffL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L2 primitives.

- `NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1R.
- `NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1R.
- `NppStatus nppiNormDiffL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C1R.
- `NppStatus nppiNormDiffL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1R.
- `NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1MR.
- `NppStatus nppiNormDiffL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C1MR.
- `NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1MR.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_32f_C1MR` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1MR.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_8u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C3R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C3R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C3R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_32f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C3R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_8u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16s_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_32f_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_8u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_AC4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_AC4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16s_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_AC4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_32f_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_AC4R.

- **NppStatus** `nppiNormDiffL2GetBufferHostSize_8u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8u_C3CMR`.
- **NppStatus** `nppiNormDiffL2GetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8s_C3CMR`.
- **NppStatus** `nppiNormDiffL2GetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_16u_C3CMR`.
- **NppStatus** `nppiNormDiffL2GetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_32f_C3CMR`.

7.20.1 Detailed Description

Primitives for computing the L2 norm of difference of pixels between two images.

7.20.2 Function Documentation

7.20.2.1 **NppStatus** `nppiNormDiff_L2_16s_AC4R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `aNormDiff[3]`, **Npp8u** `*pDeviceBuffer`)

Four-channel 16-bit signed image NormDiff_L2 ignoring alpha channel.

Parameters:

`pSrc1` Source-Image Pointer.

`nSrc1Step` Source-Image Line Step.

`pSrc2` Source-Image Pointer.

`nSrc2Step` Source-Image Line Step.

`oSizeROI` Region-of-Interest (ROI).

`aNormDiff` Array that contains computed Inf-norm of differences.

`pDeviceBuffer` Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use `nppiNormDiffL2GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.2 **NppStatus** `nppiNormDiff_L2_16s_C1R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `*pNormDiff`, **Npp8u** `*pDeviceBuffer`)

One-channel 16-bit signed image NormDiff_L2.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.3 `NppStatus nppiNormDiff_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff_L2.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormDiff* Array that contains computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.4 `NppStatus nppiNormDiff_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff_L2.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.5 NppStatus nppiNormDiff_L2_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.6 NppStatus nppiNormDiff_L2_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image NormDiff_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.7 `NppStatus nppiNormDiff_L2_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.8 `NppStatus nppiNormDiff_L2_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.20.2.9 NppStatus nppiNormDiff_L2_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.10 NppStatus nppiNormDiff_L2_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.11 NppStatus nppiNormDiff_L2_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image NormDiff_L2 ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormDiff* Array that contains computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.12 `NppStatus nppiNormDiff_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff_L2.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormDiff* Pointer to the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.13 `NppStatus nppiNormDiff_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_L2.

Parameters:

- pSrc1* Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.14 `NppStatus nppiNormDiff_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.15 `NppStatus nppiNormDiff_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.16 `NppStatus nppiNormDiff_L2_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.20.2.17 `NppStatus nppiNormDiff_L2_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.18 `NppStatus nppiNormDiff_L2_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.20.2.19 `NppStatus nppiNormDiff_L2_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.20 `NppStatus nppiNormDiff_L2_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.21 `NppStatus nppiNormDiff_L2_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.22 `NppStatus nppiNormDiff_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.20.2.23 `NppStatus nppiNormDiff_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.24 `NppStatus nppiNormDiff_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.20.2.25 `NppStatus nppiNormDiffL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_AC4R.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.26 `NppStatus nppiNormDiffL2GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C1R.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.27 NppStatus nppiNormDiffL2GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.28 NppStatus nppiNormDiffL2GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.29 NppStatus nppiNormDiffL2GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.30 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.31 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.32 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.33 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.34 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.35 NppStatus nppiNormDiffL2GetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.36 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.37 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.38 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.39 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.40 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.41 NppStatus nppiNormDiffL2GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.42 NppStatus nppiNormDiffL2GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.43 NppStatus nppiNormDiffL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.44 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.45 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.46 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.47 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.20.2.48 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21 NormRel_Inf

Primitives for computing the relative error of infinity norm between two images.

Basic NormRel_Inf

- **NppStatus nppiNormRel_Inf_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_Inf ignoring alpha channel.
- **NppStatus nppiNormRel_Inf_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_Inf ignoring alpha channel.
- **NppStatus nppiNormRel_Inf_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_Inf ignoring alpha channel.

- `NppStatus nppiNormRel_Inf_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image NormRel_Inf ignoring alpha channel.
- `NppStatus nppiNormRel_Inf_8u_C4R` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_16u_C4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_16s_C4R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit signed image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image NormRel_Inf.

Masked NormRel_Inf

See [Masked Operation](#).

- `NppStatus nppiNormRel_Inf_8u_C1MR` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
Masked one-channel 8-bit unsigned image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_8s_C1MR` (const `Npp8s *pSrc1`, int `nSrc1Step`, const `Npp8s *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
Masked one-channel 8-bit signed image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_16u_C1MR` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
Masked one-channel 16-bit unsigned image NormRel_Inf.
- `NppStatus nppiNormRel_Inf_32f_C1MR` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
Masked one-channel 32-bit floating point image NormRel_Inf.

Masked Channel NormRel_Inf

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** `nppiNormRel_Inf_8u_C3CMR` (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormRel_Inf affecting only single channel.
- **NppStatus** `nppiNormRel_Inf_8s_C3CMR` (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit signed image NormRel_Inf affecting only single channel.
- **NppStatus** `nppiNormRel_Inf_16u_C3CMR` (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormRel_Inf affecting only single channel.
- **NppStatus** `nppiNormRel_Inf_32f_C3CMR` (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormRel_Inf affecting only single channel.

NormRelInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_Inf primitives.

- **NppStatus** `nppiNormRelInfGetBufferHostSize_8u_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1R.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_16u_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1R.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_16s_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C1R.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_32s_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32s_C1R.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_32f_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1R.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_8u_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1MR.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_8s_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8s_C1MR.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_16u_C1MR` (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1MR.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_32f_C1MR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1MR.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_8u_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C3R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16u_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C3R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16s_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C3R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_32f_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_8u_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16u_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16s_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_32f_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_8u_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_AC4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16u_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_AC4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_16s_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_AC4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_32f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_AC4R.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_8u_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C3CMR.

- **NppStatus** `nppiNormRelInfGetBufferHostSize_8s_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8s_C3CMR`.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16u_C3CMR`.
- **NppStatus** `nppiNormRelInfGetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_C3CMR`.

7.21.1 Detailed Description

Primitives for computing the relative error of infinity norm between two images.

7.21.2 Function Documentation

7.21.2.1 **NppStatus** `nppiNormRel_Inf_16s_AC4R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `aNormRel[3]`, **Npp8u** `*pDeviceBuffer`)

Four-channel 16-bit signed image NormRel_Inf ignoring alpha channel.

Parameters:

`pSrc1` Source-Image Pointer.

`nSrc1Step` Source-Image Line Step.

`pSrc2` Source-Image Pointer.

`nSrc2Step` Source-Image Line Step.

`oSizeROI` Region-of-Interest (ROI).

`aNormRel` Array that contains the computed relative error for the infinity norm of two images.

`pDeviceBuffer` Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelInfGetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.2 **NppStatus** `nppiNormRel_Inf_16s_C1R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `*pNormRel`, **Npp8u** `*pDeviceBuffer`)

One-channel 16-bit signed image NormRel_Inf.

Parameters:

`pSrc1` Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.3 `NppStatus nppiNormRel_Inf_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.4 `NppStatus nppiNormRel_Inf_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.21.2.5 `NppStatus nppiNormRel_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.21.2.6 `NppStatus nppiNormRel_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.7 `NppStatus nppiNormRel_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.8 `NppStatus nppiNormRel_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.21.2.9 NppStatus nppiNormRel_Inf_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.21.2.10 NppStatus nppiNormRel_Inf_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.21.2.11 `NppStatus nppiNormRel_Inf_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.12 `NppStatus nppiNormRel_Inf_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormRel_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.13 `NppStatus nppiNormRel_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.14 `NppStatus nppiNormRel_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.15 `NppStatus nppiNormRel_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.16 `NppStatus nppiNormRel_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.17 `NppStatus nppiNormRel_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.18 `NppStatus nppiNormRel_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.19 `NppStatus nppiNormRel_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.20 `NppStatus nppiNormRel_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel_Inf.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.21 `NppStatus nppiNormRel_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.22 `NppStatus nppiNormRel_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_Inf affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.23 `NppStatus nppiNormRel_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelInfGetBufferHostSize_8u_C3R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.24 `NppStatus nppiNormRel_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelInfGetBufferHostSize_8u_C4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.21.2.25 `NppStatus nppiNormRelInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_AC4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.26 NppStatus nppiNormRelInfGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_C1R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.27 NppStatus nppiNormRelInfGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_C3R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.28 NppStatus nppiNormRelInfGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.29 NppStatus nppiNormRelInfGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.30 NppStatus nppiNormRelInfGetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.31 NppStatus nppiNormRelInfGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.32 NppStatus nppiNormRelInfGetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.33 NppStatus nppiNormRelInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.34 NppStatus nppiNormRelInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16u_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.35 NppStatus nppiNormRelInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_32f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.36 NppStatus nppiNormRelInfGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.37 NppStatus nppiNormRelInfGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.38 NppStatus nppiNormRelInfGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.39 NppStatus nppiNormRelInfGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.40 NppStatus nppiNormRelInfGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.41 NppStatus nppiNormRelInfGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32s_C1R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.42 NppStatus nppiNormRelInfGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8s_C1MR`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.43 NppStatus nppiNormRelInfGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.44 NppStatus nppiNormRelInfGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.45 NppStatus nppiNormRelInfGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.46 NppStatus nppiNormRelInfGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.47 NppStatus nppiNormRelInfGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.48 NppStatus nppiNormRelInfGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.21.2.49 NppStatus nppiNormRelInfGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22 NormRel_L1

Primitives for computing the relative error of L1 norm between two images.

Basic NormRel_L1

- **NppStatus nppiNormRel_L1_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_L1.
- **NppStatus nppiNormRel_L1_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_L1.
- **NppStatus nppiNormRel_L1_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_L1.
- **NppStatus nppiNormRel_L1_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_L1.
- **NppStatus nppiNormRel_L1_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image NormRel_L1 ignoring alpha channel.
- **NppStatus nppiNormRel_L1_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L1 ignoring alpha channel.
- **NppStatus nppiNormRel_L1_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L1 ignoring alpha channel.

- `NppStatus nppiNormRel_L1_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[3]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image NormRel_L1 ignoring alpha channel.
- `NppStatus nppiNormRel_L1_8u_C4R` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image NormRel_L1.
- `NppStatus nppiNormRel_L1_16u_C4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image NormRel_L1.
- `NppStatus nppiNormRel_L1_16s_C4R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit signed image NormRel_L1.
- `NppStatus nppiNormRel_L1_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize` `oSizeROI`, `Npp64f` `aNormRel[4]`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image NormRel_L1.

Masked NormRel_L1

See [Masked Operation](#).

- `NppStatus nppiNormRel_L1_8u_C1MR` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image NormRel_L1.
- `NppStatus nppiNormRel_L1_8s_C1MR` (const `Npp8s *pSrc1`, int `nSrc1Step`, const `Npp8s *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit signed image NormRel_L1.
- `NppStatus nppiNormRel_L1_16u_C1MR` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
One-channel 16-bit unsigned image NormRel_L1.
- `NppStatus nppiNormRel_L1_32f_C1MR` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize` `oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image NormRel_L1.

Masked Channel NormRel_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_L1_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormRel_L1 affecting only single channel.
- `NppStatus nppiNormRel_L1_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image NormRel_L1 affecting only single channel.
- `NppStatus nppiNormRel_L1_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormRel_L1 affecting only single channel.
- `NppStatus nppiNormRel_L1_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormRel_L1 affecting only single channel.

NormRelL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L1 primitives.

- `NppStatus nppiNormRelL1GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1R.
- `NppStatus nppiNormRelL1GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1R.
- `NppStatus nppiNormRelL1GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C1R.
- `NppStatus nppiNormRelL1GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1R.
- `NppStatus nppiNormRelL1GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1MR.
- `NppStatus nppiNormRelL1GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C1MR.
- `NppStatus nppiNormRelL1GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1MR.

- **NppStatus** `nppiNormRelL1GetBufferHostSize_32f_C1MR` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1MR.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_8u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C3R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C3R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_32f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C3R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_8u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16s_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_32f_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_8u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_AC4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16u_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_AC4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_16s_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_AC4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_32f_AC4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_AC4R.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_8u_C3CMR` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3CMR.
- **NppStatus** `nppiNormRelL1GetBufferHostSize_8s_C3CMR` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_8s_C3CMR`.

- `NppStatus` `nppiNormRelL1GetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16u_C3CMR`.

- `NppStatus` `nppiNormRelL1GetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C3CMR`.

7.22.1 Detailed Description

Primitives for computing the relative error of L1 norm between two images.

7.22.2 Function Documentation

7.22.2.1 `NppStatus nppiNormRel_L1_16s_AC4R` (`const Npp16s *pSrc1`, `int nSrc1Step`, `const Npp16s *pSrc2`, `int nSrc2Step`, `NppiSize oSizeROI`, `Npp64f aNormRel[3]`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed image NormRel_L1 ignoring alpha channel.

Parameters:

`pSrc1` Source-Image Pointer.

`nSrc1Step` Source-Image Line Step.

`pSrc2` Source-Image Pointer.

`nSrc2Step` Source-Image Line Step.

`oSizeROI` Region-of-Interest (ROI).

`aNormRel` Array that contains the computed relative error for the L1 norm of two images.

`pDeviceBuffer` Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelL1GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.2 `NppStatus nppiNormRel_L1_16s_C1R` (`const Npp16s *pSrc1`, `int nSrc1Step`, `const Npp16s *pSrc2`, `int nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pNormRel`, `Npp8u *pDeviceBuffer`)

One-channel 16-bit signed image NormRel_L1.

Parameters:

`pSrc1` Source-Image Pointer.

`nSrc1Step` Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelL1GetBufferHostSize_16s_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.3 NppStatus nppiNormRel_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit signed image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelL1GetBufferHostSize_16s_C3R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.4 NppStatus nppiNormRel_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.5 `NppStatus nppiNormRel_L1_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.6 `NppStatus nppiNormRel_L1_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_L1.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.7 `NppStatus nppiNormRel_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.8 `NppStatus nppiNormRel_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.9 `NppStatus nppiNormRel_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.10 `NppStatus nppiNormRel_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.11 `NppStatus nppiNormRel_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.12 `NppStatus nppiNormRel_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L1.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.13 `NppStatus nppiNormRel_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L1.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNormRel* Pointer to the computed relative error for the L1 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.14 `NppStatus nppiNormRel_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_L1 affecting only single channel.

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.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel_of_Interest Number.
- pNormRel* Pointer to the computed relative error for the L1 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.15 `NppStatus nppiNormRel_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.16 NppStatus nppiNormRel_L1_32f_C4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[4], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.22.2.17 NppStatus nppiNormRel_L1_8s_C1MR (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp8s * *pSrc2*, int *nSrc2Step*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNormRel*, Npp8u * *pDeviceBuffer*)

One-channel 8-bit signed image NormRel_L1.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.18 `NppStatus nppiNormRel_L1_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.19 `NppStatus nppiNormRel_L1_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image NormRel_L1 ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormRel* Array that contains the computed relative error for the L1 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.20 `NppStatus nppiNormRel_L1_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L1.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormRel* Pointer to the computed relative error for the L1 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.21 `NppStatus nppiNormRel_L1_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L1.

Parameters:

- pSrc1* Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.22 `NppStatus nppiNormRel_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_L1 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.23 `NppStatus nppiNormRel_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.24 `NppStatus nppiNormRel_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.22.2.25 `NppStatus nppiNormRelL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16s_AC4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.26 NppStatus nppiNormRelL1GetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.27 NppStatus nppiNormRelL1GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.28 NppStatus nppiNormRelL1GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.29 NppStatus nppiNormRelL1GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.30 NppStatus nppiNormRelL1GetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.31 NppStatus nppiNormRelL1GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.32 NppStatus nppiNormRelL1GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.33 `NppStatus nppiNormRelL1GetBufferHostSize_16u_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16u_C3R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.34 `NppStatus nppiNormRelL1GetBufferHostSize_16u_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16u_C4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.35 `NppStatus nppiNormRelL1GetBufferHostSize_32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_AC4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.36 `NppStatus nppiNormRelL1GetBufferHostSize_32f_C1MR` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C1MR`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.37 NppStatus nppiNormRelL1GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C1R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.38 NppStatus nppiNormRelL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C3CMR`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.39 NppStatus nppiNormRelL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C3R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.40 NppStatus nppiNormRelL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.41 NppStatus nppiNormRelL1GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.42 NppStatus nppiNormRelL1GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.43 NppStatus nppiNormRelL1GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.44 NppStatus nppiNormRelL1GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_8u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.45 NppStatus nppiNormRelL1GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.46 NppStatus nppiNormRelL1GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.47 NppStatus nppiNormRelL1GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.22.2.48 NppStatus nppiNormRelL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23 NormRel_L2

Primitives for computing the relative error of L2 norm between two images.

Basic NormRel_L2

- **NppStatus nppiNormRel_L2_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_L2.
- **NppStatus nppiNormRel_L2_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_L2.
- **NppStatus nppiNormRel_L2_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_L2.
- **NppStatus nppiNormRel_L2_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_L2.
- **NppStatus nppiNormRel_L2_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_L2 ignoring alpha channel.
- **NppStatus nppiNormRel_L2_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L2 ignoring alpha channel.
- **NppStatus nppiNormRel_L2_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L2 ignoring alpha channel.

- [NppStatus nppiNormRel_L2_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormRel_L2 ignoring alpha channel.
- [NppStatus nppiNormRel_L2_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_L2.
- [NppStatus nppiNormRel_L2_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L2.
- [NppStatus nppiNormRel_L2_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L2.
- [NppStatus nppiNormRel_L2_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormRel_L2.

Masked NormRel_L2

See [Masked Operation](#).

- [NppStatus nppiNormRel_L2_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit unsigned image NormRel_L2.
- [NppStatus nppiNormRel_L2_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit signed image NormRel_L2.
- [NppStatus nppiNormRel_L2_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 16-bit unsigned image NormRel_L2.
- [NppStatus nppiNormRel_L2_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 32-bit floating point image NormRel_L2.

Masked Channel NormRel_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_L2_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image NormRel_L2 affecting only single channel.
- `NppStatus nppiNormRel_L2_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit signed image NormRel_L2 affecting only single channel.
- `NppStatus nppiNormRel_L2_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 16-bit unsigned image NormRel_L2 affecting only single channel.
- `NppStatus nppiNormRel_L2_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)
Masked three-channel 32-bit floating point image NormRel_L2 affecting only single channel.

NormRelL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L2 primitives.

- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1R.
- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1R.
- `NppStatus nppiNormRelL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C1R.
- `NppStatus nppiNormRelL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1R.
- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1MR.
- `NppStatus nppiNormRelL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C1MR.
- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1MR.

- **NppStatus** [nppiNormRelL2GetBufferHostSize_32f_C1MR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1MR.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_8u_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16u_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16s_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C3R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_32f_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C3R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_8u_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16u_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16s_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_32f_C4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_8u_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_AC4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16u_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_AC4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_16s_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_AC4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_32f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_AC4R.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_8u_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3CMR.
- **NppStatus** [nppiNormRelL2GetBufferHostSize_8s_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C3CMR.

- **NppStatus** `nppiNormRelL2GetBufferHostSize_16u_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3CMR.

- **NppStatus** `nppiNormRelL2GetBufferHostSize_32f_C3CMR` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C3CMR.

7.23.1 Detailed Description

Primitives for computing the relative error of L2 norm between two images.

7.23.2 Function Documentation

7.23.2.1 **NppStatus** `nppiNormRel_L2_16s_AC4R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `aNormRel[3]`, **Npp8u** `*pDeviceBuffer`)

Four-channel 16-bit signed image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use `nppiNormRelL2GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.2 **NppStatus** `nppiNormRel_L2_16s_C1R` (**const** **Npp16s** `*pSrc1`, **int** `nSrc1Step`, **const** **Npp16s** `*pSrc2`, **int** `nSrc2Step`, **NppiSize** `oSizeROI`, **Npp64f** `*pNormRel`, **Npp8u** `*pDeviceBuffer`)

One-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.3 `NppStatus nppiNormRel_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.4 `NppStatus nppiNormRel_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.5 `NppStatus nppiNormRel_L2_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.6 `NppStatus nppiNormRel_L2_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.7 `NppStatus nppiNormRel_L2_16u_C1R` (`const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer`)

One-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel [Pointer to the computed relative error for the L2 norm of two images](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.8 `NppStatus nppiNormRel_L2_16u_C3CMR` (`const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer`)

Masked three-channel 16-bit unsigned image NormRel_L2 affecting only single channel.

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](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel [Pointer to the computed relative error for the L2 norm of two images](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.9 `NppStatus nppiNormRel_L2_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.10 `NppStatus nppiNormRel_L2_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.11 `NppStatus nppiNormRel_L2_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.23.2.12 `NppStatus nppiNormRel_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormRel_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.23.2.13 `NppStatus nppiNormRel_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.23.2.14 `NppStatus nppiNormRel_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.23.2.15 `NppStatus nppiNormRel_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.16 `NppStatus nppiNormRel_L2_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.17 `NppStatus nppiNormRel_L2_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel_L2.

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.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.18 `NppStatus nppiNormRel_L2_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.19 `NppStatus nppiNormRel_L2_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L2 ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNormRel* Array that contains the computed relative error for the L2 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.20 `NppStatus nppiNormRel_L2_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel_L2.

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.
- oSizeROI* Region-of-Interest (ROI).
- pNormRel* Pointer to the computed relative error for the L2 norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.21 `NppStatus nppiNormRel_L2_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L2.

Parameters:

- pSrc1* Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.22 `NppStatus nppiNormRel_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_L2 affecting only single channel.

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.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.23 `NppStatus nppiNormRel_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.24 `NppStatus nppiNormRel_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.23.2.25 `NppStatus nppiNormRelL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_16s_AC4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.26 NppStatus nppiNormRelL2GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.27 NppStatus nppiNormRelL2GetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.28 NppStatus nppiNormRelL2GetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.29 NppStatus nppiNormRelL2GetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.30 NppStatus nppiNormRelL2GetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.31 NppStatus nppiNormRelL2GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.32 NppStatus nppiNormRelL2GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.33 NppStatus nppiNormRelL2GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.34 NppStatus nppiNormRelL2GetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.35 NppStatus nppiNormRelL2GetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.36 NppStatus nppiNormRelL2GetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.37 NppStatus nppiNormRelL2GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_32f_C1R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.38 NppStatus nppiNormRelL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_32f_C3CMR`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.39 NppStatus nppiNormRelL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_32f_C3R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.40 NppStatus nppiNormRelL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.41 NppStatus nppiNormRelL2GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.42 NppStatus nppiNormRelL2GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.43 NppStatus nppiNormRelL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.44 NppStatus nppiNormRelL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_8u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.45 NppStatus nppiNormRelL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.46 NppStatus nppiNormRelL2GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.47 NppStatus nppiNormRelL2GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.23.2.48 NppStatus nppiNormRelL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24 DotProd

Primitives for computing the dot product of two images.

DotProd

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the dot product will be computed as

$$DotProd = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [pSrc1(j, i) \cdot pSrc2(j, i)]$$

The functions require additional scratch buffer for computations.

- **NppStatus nppiDotProd_8u64f_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image DotProd.
- **NppStatus nppiDotProd_8s64f_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image DotProd.
- **NppStatus nppiDotProd_16u64f_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image DotProd.
- **NppStatus nppiDotProd_16s64f_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image DotProd.
- **NppStatus nppiDotProd_32u64f_C1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit unsigned image DotProd.
- **NppStatus nppiDotProd_32s64f_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image DotProd.
- **NppStatus nppiDotProd_32f64f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image DotProd.
- **NppStatus nppiDotProd_8u64f_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image DotProd.
- **NppStatus nppiDotProd_8s64f_C3R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit signed image DotProd.
- **NppStatus nppiDotProd_16u64f_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)

Three-channel 16-bit unsigned image DotProd.

- `NppStatus nppiDotProd_16s64f_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 16-bit signed image DotProd.

- `NppStatus nppiDotProd_32u64f_C3R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit unsigned image DotProd.

- `NppStatus nppiDotProd_32s64f_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit signed image DotProd.

- `NppStatus nppiDotProd_32f64f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image DotProd.

- `NppStatus nppiDotProd_8u64f_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image DotProd.

- `NppStatus nppiDotProd_8s64f_C4R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image DotProd.

- `NppStatus nppiDotProd_16u64f_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit unsigned image DotProd.

- `NppStatus nppiDotProd_16s64f_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image DotProd.

- `NppStatus nppiDotProd_32u64f_C4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit unsigned image DotProd.

- `NppStatus nppiDotProd_32s64f_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed image DotProd.

- `NppStatus nppiDotProd_32f64f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image DotProd.

- `NppStatus nppiDotProd_8u64f_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

- `NppStatus nppiDotProd_8s64f_AC4R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit signed image DotProd ignoring alpha channel.
- `NppStatus nppiDotProd_16u64f_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image DotProd ignoring alpha channel.
- `NppStatus nppiDotProd_16s64f_AC4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image DotProd ignoring alpha channel.
- `NppStatus nppiDotProd_32u64f_AC4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit unsigned image DotProd ignoring alpha channel.
- `NppStatus nppiDotProd_32s64f_AC4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit signed image DotProd ignoring alpha channel.
- `NppStatus nppiDotProd_32f64f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image DotProd ignoring alpha channel.

DotProdGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

- `NppStatus nppiDotProdGetBufferHostSize_8u64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_8s64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_16u64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_16s64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_32u64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_32s64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C1R.
- `NppStatus nppiDotProdGetBufferHostSize_32f64f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C1R.

- [NppStatus nppiDotProdGetBufferHostSize_8u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_16u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_16s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32f64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_8u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_16u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_16s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32f64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_8u64f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_AC4R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_AC4R.

- **NppStatus** [nppiDotProdGetBufferHostSize_16u64f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_AC4R.
- **NppStatus** [nppiDotProdGetBufferHostSize_16s64f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_AC4R.
- **NppStatus** [nppiDotProdGetBufferHostSize_32u64f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_AC4R.
- **NppStatus** [nppiDotProdGetBufferHostSize_32s64f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_AC4R.
- **NppStatus** [nppiDotProdGetBufferHostSize_32f64f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_AC4R.

7.24.1 Detailed Description

Primitives for computing the dot product of two images.

7.24.2 Function Documentation

7.24.2.1 **NppStatus** [nppiDotProd_16s64f_AC4R](#) (**const Npp16s** * *pSrc1*, **int** *nSrc1Step*, **const Npp16s** * *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** *aDp*[3], **Npp8u** * *pDeviceBuffer*)

Four-channel 16-bit signed image DotProd ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_16s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.2 `NppStatus nppiDotProd_16s64f_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.3 `NppStatus nppiDotProd_16s64f_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.4 `NppStatus nppiDotProd_16s64f_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image DotProd.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aDp* Array that contains the computed dot product of the two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.5 NppStatus nppiDotProd_16u64f_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image DotProd ignoring alpha channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aDp* Array that contains the computed Inf-norm of differences.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.6 NppStatus nppiDotProd_16u64f_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image DotProd.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.7 NppStatus nppiDotProd_16u64f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.8 NppStatus nppiDotProd_16u64f_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.9 NppStatus nppiDotProd_32f64f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aDp*[3], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.10 NppStatus nppiDotProd_32f64f_C1R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pDp*, Npp8u * *pDeviceBuffer*)

One-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.11 NppStatus nppiDotProd_32f64f_C3R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aDp*[3], Npp8u * *pDeviceBuffer*)

Three-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.12 `NppStatus nppiDotProd_32f64f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.13 `NppStatus nppiDotProd_32s64f_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.14 `NppStatus nppiDotProd_32s64f_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.15 `NppStatus nppiDotProd_32s64f_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.16 `NppStatus nppiDotProd_32s64f_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.17 `NppStatus nppiDotProd_32u64f_AC4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.18 `NppStatus nppiDotProd_32u64f_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image DotProd.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pDp* Pointer to the computed dot product of the two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.19 `NppStatus nppiDotProd_32u64f_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image DotProd.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aDp* Array that contains the computed dot product of the two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.20 `NppStatus nppiDotProd_32u64f_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.21 NppStatus nppiDotProd_8s64f_AC4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)

Four-channel 8-bit signed image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.22 NppStatus nppiDotProd_8s64f_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)

One-channel 8-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.23 NppStatus nppiDotProd_8s64f_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)

Three-channel 8-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.24 NppStatus nppiDotProd_8s64f_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)

Four-channel 8-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.25 NppStatus nppiDotProd_8u64f_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_8u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.26 `NppStatus nppiDotProd_8u64f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_8u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.27 `NppStatus nppiDotProd_8u64f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_8u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.28 `NppStatus nppiDotProd_8u64f_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.24.2.29 `NppStatus nppiDotProdGetBufferHostSize_16s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiDotProd_16s64f_AC4R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.30 `NppStatus nppiDotProdGetBufferHostSize_16s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiDotProd_16s64f_C1R`.

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.31 NppStatus nppiDotProdGetBufferHostSize_16s64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.32 NppStatus nppiDotProdGetBufferHostSize_16s64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.33 NppStatus nppiDotProdGetBufferHostSize_16u64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.34 NppStatus nppiDotProdGetBufferHostSize_16u64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.35 NppStatus nppiDotProdGetBufferHostSize_16u64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for `nppiDotProd_16u64f_C3R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.36 NppStatus nppiDotProdGetBufferHostSize_16u64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for `nppiDotProd_16u64f_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.37 NppStatus nppiDotProdGetBufferHostSize_32f64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for `nppiDotProd_32f64f_AC4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.38 NppStatus nppiDotProdGetBufferHostSize_32f64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.39 NppStatus nppiDotProdGetBufferHostSize_32f64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.40 NppStatus nppiDotProdGetBufferHostSize_32f64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.41 NppStatus nppiDotProdGetBufferHostSize_32s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.42 NppStatus nppiDotProdGetBufferHostSize_32s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32s64f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.43 NppStatus nppiDotProdGetBufferHostSize_32s64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32s64f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.44 NppStatus nppiDotProdGetBufferHostSize_32s64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32s64f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.45 NppStatus nppiDotProdGetBufferHostSize_32u64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.46 NppStatus nppiDotProdGetBufferHostSize_32u64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.47 NppStatus nppiDotProdGetBufferHostSize_32u64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.48 NppStatus nppiDotProdGetBufferHostSize_32u64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.49 NppStatus nppiDotProdGetBufferHostSize_8s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_8s64f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.50 NppStatus nppiDotProdGetBufferHostSize_8s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_8s64f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.51 NppStatus nppiDotProdGetBufferHostSize_8s64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_8s64f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.52 NppStatus nppiDotProdGetBufferHostSize_8s64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.53 NppStatus nppiDotProdGetBufferHostSize_8u64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.54 NppStatus nppiDotProdGetBufferHostSize_8u64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.55 NppStatus nppiDotProdGetBufferHostSize_8u64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.24.2.56 NppStatus nppiDotProdGetBufferHostSize_8u64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for `nppiDotProd_8u64f_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.25 CountInRange.

Primitives for computing the amount of pixels that fall into the specified intensity range.

CountInRange

The lower bound and the upper bound are inclusive.

The functions require additional scratch buffer for computations.

- `NppStatus nppiCountInRange_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int *pCounts, `Npp8u` nLowerBound, `Npp8u` nUpperBound, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image CountInRange.
- `NppStatus nppiCountInRange_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int *pCounts, `Npp32f` nLowerBound, `Npp32f` nUpperBound, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image CountInRange.
- `NppStatus nppiCountInRange_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image CountInRange.
- `NppStatus nppiCountInRange_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image CountInRange.
- `NppStatus nppiCountInRange_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.
- `NppStatus nppiCountInRange_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

CountInRangeGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CountInRange primitives.

- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiCountInRange_8u_C1R.
- `NppStatus nppiCountInRangeGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiCountInRange_32f_C1R.
- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C3R.

- **NppStatus** [nppiCountInRangeGetBufferHostSize_32f_C3R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C3R.

- **NppStatus** [nppiCountInRangeGetBufferHostSize_8u_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_AC4R.

- **NppStatus** [nppiCountInRangeGetBufferHostSize_32f_AC4R](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_AC4R.

7.25.1 Detailed Description

Primitives for computing the amount of pixels that fall into the specified intensity range.

7.25.2 Function Documentation

7.25.2.1 **NppStatus nppiCountInRange_32f_AC4R** (**const Npp32f *pSrc**, **int nSrcStep**, **NppiSize oSizeROI**, **int aCounts[3]**, **Npp32f aLowerBound[3]**, **Npp32f aUpperBound[3]**, **Npp8u *pDeviceBuffer**)

Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.

aLowerBound Fixed size array of the lower bound of the specified range, one per channel.

aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_RANGE_ERROR` if the lower bound is larger than the upper bound.

7.25.2.2 **NppStatus nppiCountInRange_32f_C1R** (**const Npp32f *pSrc**, **int nSrcStep**, **NppiSize oSizeROI**, **int *pCounts**, **Npp32f nLowerBound**, **Npp32f nUpperBound**, **Npp8u *pDeviceBuffer**)

One-channel 32-bit floating point image CountInRange.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pCounts Pointer to the number of pixels that fall into the specified range.

nLowerBound Lower bound of the specified range.

nUpperBound Upper bound of the specified range.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_RANGE_ERROR` if the lower bound is larger than the upper bound.

7.25.2.3 NppStatus nppiCountInRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp32f aLowerBound[3], Npp32f aUpperBound[3], Npp8u * pDeviceBuffer)

Three-channel 32-bit floating point image CountInRange.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.

aLowerBound Fixed size array of the lower bound of the specified range, one per channel.

aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_RANGE_ERROR` if the lower bound is larger than the upper bound.

7.25.2.4 NppStatus nppiCountInRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.
aLowerBound Fixed size array of the lower bound of the specified range, one per channel.
aUpperBound Fixed size array of the upper bound of the specified range, one per channel.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiCountInRangeGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_RANGE_ERROR` if the lower bound is larger than the upper bound.

7.25.2.5 `NppStatus nppiCountInRange_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int * pCounts, Npp8u nLowerBound, Npp8u nUpperBound, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pCounts Pointer to the number of pixels that fall into the specified range.
nLowerBound Lower bound of the specified range.
nUpperBound Upper bound of the specified range.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiCountInRangeGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_RANGE_ERROR` if the lower bound is larger than the upper bound.

7.25.2.6 `NppStatus nppiCountInRange_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.
aLowerBound Fixed size array of the lower bound of the specified range, one per channel.
aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.25.2.7 NppStatus nppiCountInRangeGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.25.2.8 NppStatus nppiCountInRangeGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.25.2.9 NppStatus nppiCountInRangeGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.25.2.10 NppStatus nppiCountInRangeGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.25.2.11 NppStatus nppiCountInRangeGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_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_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.25.2.12 NppStatus nppiCountInRangeGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.26 MaxEvery

Primitives for computing the maximal value of the pixel pair from two images.

MaxEvery

The maximum is stored into the second image.

- **NppStatus nppiMaxEvery_8u_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One-channel 8-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16u_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One-channel 16-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16s_C1IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One-channel 16-bit signed image MaxEvery.
- **NppStatus nppiMaxEvery_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One-channel 32-bit floating point image MaxEvery.
- **NppStatus nppiMaxEvery_8u_C3IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three-channel 8-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16u_C3IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16s_C3IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit signed image MaxEvery.
- **NppStatus nppiMaxEvery_32f_C3IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three-channel 32-bit floating point image MaxEvery.
- **NppStatus nppiMaxEvery_8u_C4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16u_C4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned image MaxEvery.
- **NppStatus nppiMaxEvery_16s_C4IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MaxEvery.

- [NppStatus nppiMaxEvery_32f_C4IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating point image MaxEvery.

- [NppStatus nppiMaxEvery_8u_AC4IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

- [NppStatus nppiMaxEvery_16u_AC4IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

- [NppStatus nppiMaxEvery_16s_AC4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

- [NppStatus nppiMaxEvery_32f_AC4IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

7.26.1 Detailed Description

Primitives for computing the maximal value of the pixel pair from two images.

7.26.2 Function Documentation

7.26.2.1 [NppStatus nppiMaxEvery_16s_AC4IR](#) (const [Npp16s](#) * pSrc, int nSrcStep, [Npp16s](#) * pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.26.2.2 NppStatus nppiMaxEvery_16s_C1IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.3 NppStatus nppiMaxEvery_16s_C3IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.4 NppStatus nppiMaxEvery_16s_C4IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.5 `NppStatus nppiMaxEvery_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.6 `NppStatus nppiMaxEvery_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One-channel 16-bit unsigned image MaxEvery.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.7 `NppStatus nppiMaxEvery_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three-channel 16-bit unsigned image MaxEvery.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.8 NppStatus nppiMaxEvery_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.9 NppStatus nppiMaxEvery_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.10 NppStatus nppiMaxEvery_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.11 NppStatus nppiMaxEvery_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.12 NppStatus nppiMaxEvery_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.13 NppStatus nppiMaxEvery_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.14 NppStatus nppiMaxEvery_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.15 NppStatus nppiMaxEvery_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.26.2.16 NppStatus nppiMaxEvery_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27 MinEvery

Primitives for computing the minimal value of the pixel pair from two images.

MinEvery

The minimum is stored into the second image.

- [NppStatus nppiMinEvery_8u_C1IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C1IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C1IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit signed image MinEvery.
- [NppStatus nppiMinEvery_32f_C1IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 32-bit floating point image MinEvery.
- [NppStatus nppiMinEvery_8u_C3IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C3IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C3IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit signed image MinEvery.
- [NppStatus nppiMinEvery_32f_C3IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 32-bit floating point image MinEvery.
- [NppStatus nppiMinEvery_8u_C4IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C4IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MinEvery.

- [NppStatus nppiMinEvery_32f_C4IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating point image MinEvery.

- [NppStatus nppiMinEvery_8u_AC4IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

- [NppStatus nppiMinEvery_16u_AC4IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

- [NppStatus nppiMinEvery_16s_AC4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

- [NppStatus nppiMinEvery_32f_AC4IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

7.27.1 Detailed Description

Primitives for computing the minimal value of the pixel pair from two images.

7.27.2 Function Documentation

7.27.2.1 [NppStatus nppiMinEvery_16s_AC4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.27.2.2 `NppStatus nppiMinEvery_16s_C1IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.3 `NppStatus nppiMinEvery_16s_C3IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.4 `NppStatus nppiMinEvery_16s_C4IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.5 NppStatus nppiMinEvery_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.6 NppStatus nppiMinEvery_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.7 NppStatus nppiMinEvery_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.8 NppStatus nppiMinEvery_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.9 NppStatus nppiMinEvery_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.10 NppStatus nppiMinEvery_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.11 NppStatus nppiMinEvery_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.12 NppStatus nppiMinEvery_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.13 NppStatus nppiMinEvery_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.14 `NppStatus nppiMinEvery_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.15 `NppStatus nppiMinEvery_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.27.2.16 `NppStatus nppiMinEvery_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.28 Integral

Primitives for computing the integral image of a given image.

Integral

Given an input image $pSrc$ and the specified value $nVal$, the pixel value of the integral image $pDst$ at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

If the size of the input image is $W \times H$, the size of the integral image will be $(W + 1) \times (H + 1)$.

- [NppStatus nppiIntegral_8u32s_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32s](#) nVal)
One-channel 8-bit unsigned image Integral with 32-bit signed output.
- [NppStatus nppiIntegral_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)
One-channel 8-bit unsigned image Integral with 32-bit floating point output.

7.28.1 Detailed Description

Primitives for computing the integral image of a given image.

7.28.2 Function Documentation

7.28.2.1 [NppStatus nppiIntegral_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)

One-channel 8-bit unsigned image Integral with 32-bit floating point output.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oROI* [Region-of-Interest \(ROI\)](#).
- nVal* The value to add to pDst image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.28.2.2 `NppStatus nppiIntegral_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oROI, Npp32s nVal)`

One-channel 8-bit unsigned image Integral with 32-bit signed output.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nVal The value to add to pDst image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.29 SqrIntegral

Primitives for computing both the integral and the squared integral images of a given image.

SqrIntegral

Given an input image $pSrc$ and the specified value $nVal$, the pixel value of the integral image $pDst$ at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

Given an input image $pSrc$ and the specified value $nValSqr$, the pixel value of the squared integral image $pSqr$ at coordinate (i, j) will be computed as

$$pSqr(j, i) = nValSqr + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)^2$$

If the size of the input image is $W \times H$, the size of the squared integral image will be $(W + 1) \times (H + 1)$.

- **NppStatus nppiSqrIntegral_8u32s_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **Npp32s** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp32s** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.
- **NppStatus nppiSqrIntegral_8u32s64f_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp64f** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.
- **NppStatus nppiSqrIntegral_8u32f64f_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.

7.29.1 Detailed Description

Primitives for computing both the integral and the squared integral images of a given image.

7.29.2 Function Documentation

7.29.2.1 NppStatus nppiSqrIntegral_8u32f64f_C1R (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit floating point. Destination square integral image is 64-bit double floating point.

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).
nVal The value to add to pDst image pixels
nValSqr The value to add to pSqr image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.29.2.2 `NppStatus nppiSqrIntegral_8u32s64f_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, Npp64f * pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp64f nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit signed int. Destination square integral image is 64-bit double floating point.

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).
nVal The value to add to pDst image pixels
nValSqr The value to add to pSqr image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.29.2.3 `NppStatus nppiSqrIntegral_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, Npp32s * pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp32s nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image and square integral image are 32-bit signed int.

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).

nVal The value to add to pDst image pixels

nValSqr The value to add to pSqr image pixels

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30 RectStdDev

Primitives for computing the standard deviation of the integral images.

RectStdDev

- **NppStatus nppiRectStdDev_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp64f** *pSqr, int nSqrStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)
One-channel 32-bit floating point image RectStdDev.
- **NppStatus nppiRectStdDev_32s_C1RSfs** (const **Npp32s** *pSrc, int nSrcStep, const **Npp32s** *pSqr, int nSqrStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect, int nScaleFactor)
One-channel 32-bit signed image RectStdDev, scaled by $2^{(- nScaleFactor)}$.
- **NppStatus nppiRectStdDev_32s32f_C1R** (const **Npp32s** *pSrc, int nSrcStep, const **Npp64f** *pSqr, int nSqrStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)
One-channel 32-bit signed image RectStdDev.

7.30.1 Detailed Description

Primitives for computing the standard deviation of the integral images.

The function computes the standard deviation of the pixel in the rectangular window with the integral image *pSrc* and the squared integral image *pSqr*, which can be obtained by calling [Integral](#) and [SqrIntegral](#).

The standard deviation of the pixel (*j*, *i*) can be computed using the formula:

$$pDst(j, i) = \sqrt{\max(0, \frac{\sum(SqrIntegral) \cdot N - (\sum(Integral))^2}{N^2})}$$

where $\sum(SqrIntegral) = pSqr[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSqr[j + oRect.y, i + oRect.x + oRect.width] - pSqr[j + oRect.y + oRect.height, i + oRect.x] + pSqr[j + oRect.y, i + oRect.x]$, $\sum(Integral) = pSrc[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSrc[j + oRect.y, i + oRect.x + oRect.width] - pSrc[j + oRect.y + oRect.height, i + oRect.x] + pSrc[j + oRect.y, i + oRect.x]$, $N = oRect.width \cdot oRect.height$.

The size of the *pSrc* and *pSqr* should be $(oSizeROI.width + oRect.x + oRect.width, oSizeROI.height + oRect.y + oRect.height)$.

7.30.2 Function Documentation

7.30.2.1 NppStatus nppiRectStdDev_32f_C1R (const Npp32f *pSrc, int nSrcStep, const Npp64f *pSqr, int nSqrStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect)

One-channel 32-bit floating point image RectStdDev.

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).
oRect rectangular window

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.2 NppStatus nppiRectStdDev_32s32f_C1R (const Npp32s * pSrc, int nSrcStep, const Npp64f * pSqr, int nSqrStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect)

One-channel 32-bit signed image RectStdDev.

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).
oRect rectangular window

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.3 NppStatus nppiRectStdDev_32s_C1RSfs (const Npp32s * pSrc, int nSrcStep, const Npp32s * pSqr, int nSqrStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect, int nScaleFactor)

One-channel 32-bit signed image RectStdDev, scaled by $2^{(-nScaleFactor)}$.

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).
oRect rectangular window

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31 HistogramEven

Primitives for computing the histogram of an image with evenly distributed bins.

HistogramEven

The $nLowerLevel$ (inclusive) and $nUpperLevel$ (exclusive) define the boundaries of the range, which are evenly segmented into $nLevel - 1$ bins.

The computed histogram is stored in $pHist$. The levels are calculated by another primitive [nppiEvenLevelsHost_32s](#) and are stored in a host pointer $hpLevels$. The number of levels is also $nLevel - 1$. The histogram $pHist[k]$ is defined as the total number of pixels that fall into the range: $hpLevels[k] \leq pSrc(j, i) < hpLevels[k + 1]$. The functions require additional scratch buffer for computations.

- [NppStatus nppiEvenLevelsHost_32s](#) ([Npp32s](#) *hpLevels, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel)
Compute levels with even distribution.
- [NppStatus nppiHistogramEven_8u_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) *pBuffer)
One-channel 8-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_8u_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) *pBuffer)
Three-channel 8-bit unsigned HistogramEven.
- [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)
Four-channel 8-bit unsigned HistogramEven.
- [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)
Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.
- [NppStatus nppiHistogramEven_16u_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) *pBuffer)
One-channel 16-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_16u_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) *pBuffer)
Three-channel 16-bit unsigned HistogramEven.
- [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)
Four-channel 16-bit unsigned HistogramEven.

- `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`)
Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.
- `NppStatus nppiHistogramEven_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist`, int `nLevels`, `Npp32s` `nLowerLevel`, `Npp32s` `nUpperLevel`, `Npp8u *pBuffer`)
One-channel 16-bit signed HistogramEven.
- `NppStatus nppiHistogramEven_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, int `nLevels[3]`, `Npp32s` `nLowerLevel[3]`, `Npp32s` `nUpperLevel[3]`, `Npp8u *pBuffer`)
Three-channel 16-bit signed HistogramEven.
- `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`)
Four-channel 16-bit signed HistogramEven.
- `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`)
Four-channel 16-bit signed HistogramEven ignoring alpha channel.

HistogramEvenGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramEven primitives.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C1R` (`NppiSize` `oSizeROI`, int `nLevels`, int `*hpBufferSize`)
Buffer size for `nppiHistogramEven_8u_C1R`.
- `NppStatus nppiHistogramEvenGetBufferSize_8u_C3R` (`NppiSize` `oSizeROI`, int `nLevels[3]`, int `*hpBufferSize`)
Buffer size for `nppiHistogramEven_8u_C3R`.
- `NppStatus nppiHistogramEvenGetBufferSize_8u_C4R` (`NppiSize` `oSizeROI`, int `nLevels[4]`, int `*hpBufferSize`)
Buffer size for `nppiHistogramEven_8u_C4R`.
- `NppStatus nppiHistogramEvenGetBufferSize_8u_AC4R` (`NppiSize` `oSizeROI`, int `nLevels[3]`, int `*hpBufferSize`)
Buffer size for `nppiHistogramEven_8u_AC4R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16u_C1R` (`NppiSize` `oSizeROI`, int `nLevels`, int `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_C1R`.

- `NppStatus nppiHistogramEvenGetBufferSize_16u_C3R` (`NppiSize` `oSizeROI`, `int` `nLevels[3]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_C3R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16u_C4R` (`NppiSize` `oSizeROI`, `int` `nLevels[4]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_C4R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int` `nLevels[3]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_AC4R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16s_C1R` (`NppiSize` `oSizeROI`, `int` `nLevels`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C1R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16s_C3R` (`NppiSize` `oSizeROI`, `int` `nLevels[3]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C3R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16s_C4R` (`NppiSize` `oSizeROI`, `int` `nLevels[4]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C4R`.
- `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int` `nLevels[3]`, `int` `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_AC4R`.

7.31.1 Detailed Description

Primitives for computing the histogram of an image with evenly distributed bins.

7.31.2 Function Documentation

7.31.2.1 `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.

nLowerLevel Lower boundary value of the lowest level.

nUpperLevel Upper boundary value of the greatest level.

Returns:

image_data_error_codes, or NPP_HISTO_NUMBER_OF_LEVELS_ERROR if an invalid nLevels is specified.

7.31.2.2 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)

Four-channel 16-bit signed HistogramEven ignoring alpha channel.

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](#)

7.31.2.3 NppStatus nppiHistogramEven_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)

One-channel 16-bit signed HistogramEven.

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](#)

7.31.2.4 NppStatus nppiHistogramEven_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u * *pBuffer*)

Three-channel 16-bit signed HistogramEven.

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_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.5 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*)

Four-channel 16-bit signed HistogramEven.

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](#)

7.31.2.6 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)

Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.

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](#)

7.31.2.7 NppStatus nppiHistogramEven_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)

One-channel 16-bit unsigned HistogramEven.

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](#)

7.31.2.8 NppStatus nppiHistogramEven_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u * *pBuffer*)

Three-channel 16-bit unsigned HistogramEven.

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_C3R`) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.9 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*)

Four-channel 16-bit unsigned HistogramEven.

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](#)

7.31.2.10 `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)`

Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.

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](#)

7.31.2.11 `NppStatus nppiHistogramEven_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

One-channel 8-bit unsigned HistogramEven.

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](#).

7.31.2.12 `NppStatus nppiHistogramEven_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Three-channel 8-bit unsigned HistogramEven.

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_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.31.2.13 `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)`

Four-channel 8-bit unsigned HistogramEven.

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](#).

7.31.2.14 `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)`

Buffer size for [nppiHistogramEven_16s_AC4R](#).

Parameters:

- oSizeROI* Region-of-Interest (ROI).
- nLevels* Array containing number of levels per color channel.
- hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.15 NppStatus nppiHistogramEvenGetBufferSize_16s_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Buffer size for [nppiHistogramEven_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.16 NppStatus nppiHistogramEvenGetBufferSize_16s_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.17 NppStatus nppiHistogramEvenGetBufferSize_16s_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.18 NppStatus nppiHistogramEvenGetBufferSize_16u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.19 NppStatus nppiHistogramEvenGetBufferSize_16u_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.20 NppStatus nppiHistogramEvenGetBufferSize_16u_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.21 `NppStatus nppiHistogramEvenGetBufferSize_16u_C4R` (`NppiSize oSizeROI`, `int nLevels[4]`, `int * hpBufferSize`)

Buffer size for `nppiHistogramEven_16u_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.22 `NppStatus nppiHistogramEvenGetBufferSize_8u_AC4R` (`NppiSize oSizeROI`, `int nLevels[3]`, `int * hpBufferSize`)

Buffer size for `nppiHistogramEven_8u_AC4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.23 `NppStatus nppiHistogramEvenGetBufferSize_8u_C1R` (`NppiSize oSizeROI`, `int nLevels`, `int * hpBufferSize`)

Buffer size for `nppiHistogramEven_8u_C1R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.24 NppStatus nppiHistogramEvenGetBufferSize_8u_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.31.2.25 NppStatus nppiHistogramEvenGetBufferSize_8u_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Buffer size for [nppiHistogramEven_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32 HistogramRange

Primitives for computing the histogram of an image within specified ranges.

HistogramEven

The histogram is computed according to the ranges provided in *pLevels*.

The histogram $pHist[k]$ is defined as the total number of pixels that fall into the range: $pLevels[k] \leq pSrc(j, i) < pLevels[k + 1]$. The number of the histogram bins is $nLevel - 1$. The functions require additional scratch buffer for computations.

- `NppStatus nppiHistogramRange_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist`, const `Npp32s *pLevels`, int `nLevels`, `Npp8u *pBuffer`)
One-channel 8-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)
Three-channel 8-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[4]`, const `Npp32s *pLevels[4]`, int `nLevels[4]`, `Npp8u *pBuffer`)
Four-channel 8-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)
Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.
- `NppStatus nppiHistogramRange_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist`, const `Npp32s *pLevels`, int `nLevels`, `Npp8u *pBuffer`)
One-channel 16-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)
Three-channel 16-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[4]`, const `Npp32s *pLevels[4]`, int `nLevels[4]`, `Npp8u *pBuffer`)
Four-channel 16-bit unsigned HistogramRange.
- `NppStatus nppiHistogramRange_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)
Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.
- `NppStatus nppiHistogramRange_16s_C1R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist`, const `Npp32s *pLevels`, int `nLevels`, `Npp8u *pBuffer`)
One-channel 16-bit signed HistogramRange.
- `NppStatus nppiHistogramRange_16s_C3R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)
Three-channel 16-bit signed HistogramRange.

- **NppStatus nppiHistogramRange_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32s** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 16-bit signed HistogramRange.
- **NppStatus nppiHistogramRange_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 16-bit signed HistogramRange.
- **NppStatus nppiHistogramRange_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist, const **Npp32f** *pLevels, int nLevels, **Npp8u** *pBuffer)
One-channel 32-bit floating point HistogramRange.
- **NppStatus nppiHistogramRange_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32f** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Three-channel 32-bit floating point HistogramRange.
- **NppStatus nppiHistogramRange_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32f** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 32-bit floating point HistogramRange.
- **NppStatus nppiHistogramRange_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32f** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

HistogramRangeGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramRange primitives.

- **NppStatus nppiHistogramRangeGetBufferSize_8u_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C1R.
- **NppStatus nppiHistogramRangeGetBufferSize_8u_C3R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C3R.
- **NppStatus nppiHistogramRangeGetBufferSize_8u_C4R** (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C4R.
- **NppStatus nppiHistogramRangeGetBufferSize_8u_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_AC4R.
- **NppStatus nppiHistogramRangeGetBufferSize_16u_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C1R.

- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_C3R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C3R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_C4R` (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C4R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_AC4R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C1R` (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C1R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C3R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C3R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C4R` (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C4R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_AC4R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_32f_C1R` (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C1R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_32f_C3R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C3R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_32f_C4R` (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C4R.
- **NppStatus** `nppiHistogramRangeGetBufferSize_32f_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_AC4R.

7.32.1 Detailed Description

Primitives for computing the histogram of an image within specified ranges.

7.32.2 Function Documentation

7.32.2.1 NppStatus nppiHistogramRange_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], const Npp32s * *pLevels*[3], int *nLevels*[3], Npp8u * *pBuffer*)

Four-channel 16-bit signed HistogramRange.

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_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.2 NppStatus nppiHistogramRange_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*, const Npp32s * *pLevels*, int *nLevels*, Npp8u * *pBuffer*)

One-channel 16-bit signed HistogramRange.

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_16s_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.3 NppStatus nppiHistogramRange_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], const Npp32s * *pLevels*[3], int *nLevels*[3], Npp8u * *pBuffer*)

Three-channel 16-bit signed HistogramRange.

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_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.4 `NppStatus nppiHistogramRange_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramRange.

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](#)

7.32.2.5 `NppStatus nppiHistogramRange_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.

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](#)

7.32.2.6 NppStatus nppiHistogramRange_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)

One-channel 16-bit unsigned HistogramRange.

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](#)

7.32.2.7 NppStatus nppiHistogramRange_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)

Three-channel 16-bit unsigned HistogramRange.

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_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.8 NppStatus nppiHistogramRange_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)

Four-channel 16-bit unsigned HistogramRange.

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_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.9 NppStatus nppiHistogramRange_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)

Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

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_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.10 `NppStatus nppiHistogramRange_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32f * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 32-bit floating point HistogramRange.

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_32f_C1R`) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.11 `NppStatus nppiHistogramRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 32-bit floating point HistogramRange.

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_C3R`) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.12 `NppStatus nppiHistogramRange_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32f * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 32-bit floating point HistogramRange.

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](#)

7.32.2.13 `NppStatus nppiHistogramRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.

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](#)

7.32.2.14 `NppStatus nppiHistogramRange_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 8-bit unsigned HistogramRange.

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](#)

7.32.2.15 `NppStatus nppiHistogramRange_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 8-bit unsigned HistogramRange.

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_C3R`) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.16 `NppStatus nppiHistogramRange_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramRange.

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](#)

7.32.2.17 NppStatus nppiHistogramRangeGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.18 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 Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.19 NppStatus nppiHistogramRangeGetBufferSize_16s_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.20 NppStatus nppiHistogramRangeGetBufferSize_16s_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.21 NppStatus nppiHistogramRangeGetBufferSize_16u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.22 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 Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.23 NppStatus nppiHistogramRangeGetBufferSize_16u_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.24 NppStatus nppiHistogramRangeGetBufferSize_16u_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.25 NppStatus nppiHistogramRangeGetBufferSize_32f_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.26 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 Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.27 NppStatus nppiHistogramRangeGetBufferSize_32f_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.28 NppStatus nppiHistogramRangeGetBufferSize_32f_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.29 NppStatus nppiHistogramRangeGetBufferSize_8u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.30 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 Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.31 NppStatus nppiHistogramRangeGetBufferSize_8u_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.32.2.32 NppStatus nppiHistogramRangeGetBufferSize_8u_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.33 Image Proximity

Primitives for computing the proximity measure between a source image and a template image.

Modules

- [SqrDistanceFull_Norm](#)
Primitives for computing the normalized Euclidean distance between two images with full mode.
- [SqrDistanceSame_Norm](#)
Primitives for computing the normalized Euclidean distance between two images with same mode.
- [SqrDistanceValid_Norm](#)
Primitives for computing the normalized Euclidean distance between two images with valid mode.
- [CrossCorrFull_Norm](#)
Primitives for computing the normalized cross correlation between two images with full mode.
- [CrossCorrSame_Norm](#)
Primitives for computing the normalized cross correlation between two images with same mode.
- [CrossCorrValid_Norm](#)
Primitives for computing the normalized cross correlation between two images with valid mode.
- [CrossCorrValid](#)
Primitives for computing the cross correlation between two images with valid mode.
- [CrossCorrFull_NormLevel](#)
Primitives for computing the normalized cross correlation coefficient between two images with full mode.
- [CrossCorrSame_NormLevel](#)
Primitives for computing the normalized cross correlation coefficient between two images with same mode.
- [CrossCorrValid_NormLevel](#)
Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

7.33.1 Detailed Description

Primitives for computing the proximity measure between a source image and a template image.

7.33.2 General Introduction

There are basically two approaches to compute the proximity measure for template matching, Euclidean distance and the cross correlation.

1. Euclidean distance computes the sum of the squared distance (SSD) between the corresponding pixels of the source image and the template image. The smaller the distance is, the more similar the source image and the template image is around the pixel. The anchor of the template image is used during the computations, which always lies in the geometric center of the image. Given a source image $pSrc$ ($W_s \times H_s$) and a template image $pTpl$ ($W_t \times H_t$), the Euclidean distance $D_{st}(c, r)$ between two images at pixel r and column c is computed as (s stands for source image and t for template image for short):

$$D_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]^2$$

2. Cross correlation computes the sum of the product between the corresponding pixels of the source image and the template image. The cross correlation $R_{st}(c, r)$ is calculated as:

$$R_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) \cdot pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]$$

The larger the cross correlation value is, the more similar the source image and the template image is around the pixel.

3. The cross correlation $R_{st}(c, r)$ is affected by the brightness of the images which may vary due to the lighting and exposure conditions. Therefore, NPP computes the cross correlation coefficient to circumvent this dependence. This is typically done at every step by subtracting the mean from every pixel value, i.e.,

$$\tilde{R}_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t] \cdot [pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2}) - Mean_s]$$

NPP computes the normalized values of Euclidean distance, cross correlation and the cross correlation coefficient.

1. The normalized Euclidean distance $\sigma_{st}(c, r)$ is defined as:

$$\sigma_{st}(c, r) = \frac{D_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

2. The normalized cross correlation $\rho_{st}(c, r)$ is defined as:

$$\rho_{st}(c, r) = \frac{R_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $R_{ss}(c, r)$ and $R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ denote the auto correlation of the source image and the template image individually. They are defined as:

$$R_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} pSrc(j, i)$$

$$R_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} pTpl(j, i)$$

3. Similarly, the normalized cross correlation coefficient $\gamma_{st}(c, r)$ is calculated as:

$$\gamma_{st}(c, r) = \frac{\tilde{R}_{st}(c, r)}{\sqrt{\tilde{R}_{ss}(c, r) \cdot \tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $\tilde{R}_{ss}(c, r)$ and $\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ are defined as:

$$\tilde{R}_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} [pSrc(j, i) - Mean_s]$$

$$\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t]$$

7.33.3 Categorizations

The Euclidean distance and the cross correlation are categorized into three types, full, same, and valid.

1. Full mode indicates that the anchor of the template image starts from the outside of the source image, assuming the out-of-boundary pixels are zero-padded. The size of the destination image is $(W_s + W_t - 1) \times (H_s + H_t - 1)$.
2. Same mode means that the anchor of the template image starts from the top left pixel of the source image. All the out-of-boundary pixels are also zero-padded. The size of the destination image is the same as the source one, i.e., $W_s \times H_s$.
3. Valid mode indicates that there are no out-of-boundary readings from the source image. The anchor of the template image starts from the inside of the source image. The size of the destination image is $(W_s - W_t + 1) \times (H_s - H_t + 1)$.

7.34 SqrDistanceFull_Norm

Primitives for computing the normalized Euclidean distance between two images with full mode.

SqrDistanceFull_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

- [NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C1R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image SqrDistanceFull_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 8-bit unsigned image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceFull_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

7.34.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with full mode.

7.34.2 Function Documentation

7.34.2.1 `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.2 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.3 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.4 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.5 `NppStatus nppiSqrDistanceFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.6 `NppStatus nppiSqrDistanceFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.7 `NppStatus nppiSqrDistanceFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.8 `NppStatus nppiSqrDistanceFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.9 `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.10 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.11 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.12 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.13 `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.14 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.15 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.16 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.17 `NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.18 `NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.19 `NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.20 `NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35 SqrDistanceSame_Norm

Primitives for computing the normalized Euclidean distance between two images with same mode.

SqrDistanceSame_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
One-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Three-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiSqrDistanceSame_Norm_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 32-bit floating point image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Three-channel 32-bit floating point image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image SqrDistanceSame_Norm ignoring alpha channel.
- `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 8-bit unsigned image SqrDistanceSame_Norm.

- [NppStatus nppiSqrDistanceSame_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit unsigned image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_8u32f_C4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_8u32f_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceSame_Norm_8s32f_C1R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

One-channel 8-bit signed image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_8s32f_C3R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit signed image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_8s32f_C4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceSame_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceSame_Norm_16u32f_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 16-bit unsigned image SqrDistanceSame_Norm.
- [NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm.

- `NppStatus nppiSqrDistanceSame_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

7.35.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with same mode.

7.35.2 Function Documentation

- 7.35.2.1 `NppStatus nppiSqrDistanceSame_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.35.2.2 `NppStatus nppiSqrDistanceSame_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.3 `NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.4 `NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.5 `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.6 `NppStatus nppiSqrDistanceSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.7 `NppStatus nppiSqrDistanceSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.8 `NppStatus nppiSqrDistanceSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.9 `NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.10 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.11 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.12 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.13 `NppStatus nppiSqrDistanceSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.14 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.15 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.16 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.17 `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.18 `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.19 `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.20 `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36 SqrDistanceValid_Norm

Primitives for computing the normalized Euclidean distance between two images with valid mode.

SqrDistanceValid_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
One-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\text{c}} - \text{nScaleFactor}$.
- `NppStatus nppiSqrDistanceValid_Norm_8u_C3RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Three-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\text{c}} - \text{nScaleFactor}$.
- `NppStatus nppiSqrDistanceValid_Norm_8u_C4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\text{c}} - \text{nScaleFactor}$.
- `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel, scaled by $2^{\text{c}} - \text{nScaleFactor}$.
- `NppStatus nppiSqrDistanceValid_Norm_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 32-bit floating point image SqrDistanceValid_Norm.
- `NppStatus nppiSqrDistanceValid_Norm_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Three-channel 32-bit floating point image SqrDistanceValid_Norm.
- `NppStatus nppiSqrDistanceValid_Norm_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image SqrDistanceValid_Norm.
- `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image SqrDistanceValid_Norm ignoring alpha channel.
- `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 8-bit unsigned image SqrDistanceValid_Norm.

- [NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 8-bit unsigned image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 8-bit unsigned image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 8-bit signed image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_8s32f_C3R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 8-bit signed image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_8s32f_C4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 8-bit signed image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_8s32f_AC4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 8-bit signed image SqrDistanceValid_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceValid_Norm_16u32f_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 16-bit unsigned image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_16u32f_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 16-bit unsigned image SqrDistanceValid_Norm.
- [NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 16-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_16u32f_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

7.36.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with valid mode.

7.36.2 Function Documentation

- 7.36.2.1 `NppStatus nppiSqrDistanceValid_Norm_16u32f_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.36.2.2 `NppStatus nppiSqrDistanceValid_Norm_16u32f_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)

One-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.3 `NppStatus nppiSqrDistanceValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.4 `NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.5 `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.6 `NppStatus nppiSqrDistanceValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.7 `NppStatus nppiSqrDistanceValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.8 `NppStatus nppiSqrDistanceValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.9 `NppStatus nppiSqrDistanceValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.10 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.11 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.12 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.13 `NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.14 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.15 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.16 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.17 `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs` (`const Npp8u * pSrc`, `int nSrcStep`, `NppiSize oSrcRoiSize`, `const Npp8u * pTpl`, `int nTplStep`, `NppiSize oTplRoiSize`, `Npp8u * pDst`, `int nDstStep`, `int nScaleFactor`)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.18 `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs` (`const Npp8u * pSrc`, `int nSrcStep`, `NppiSize oSrcRoiSize`, `const Npp8u * pTpl`, `int nTplStep`, `NppiSize oTplRoiSize`, `Npp8u * pDst`, `int nDstStep`, `int nScaleFactor`)

One-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.19 `NppStatus nppiSqrDistanceValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.20 `NppStatus nppiSqrDistanceValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37 CrossCorrFull_Norm

Primitives for computing the normalized cross correlation between two images with full mode.

CrossCorrFull_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
One-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.
- `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Three-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.
- `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.
- `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`)
Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel, scaled by $2^{\ell} - nScaleFactor$.
- `NppStatus nppiCrossCorrFull_Norm_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 32-bit floating point image CrossCorrFull_Norm.
- `NppStatus nppiCrossCorrFull_Norm_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Three-channel 32-bit floating point image CrossCorrFull_Norm.
- `NppStatus nppiCrossCorrFull_Norm_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image CrossCorrFull_Norm.
- `NppStatus nppiCrossCorrFull_Norm_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
Four-channel 32-bit floating point image CrossCorrFull_Norm ignoring alpha channel.
- `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 8-bit unsigned image CrossCorrFull_Norm.
- `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)

Three-channel 8-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrFull_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

7.37.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with full mode.

7.37.2 Function Documentation

7.37.2.1 `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.2 `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.3 `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.4 `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.5 `NppStatus nppiCrossCorrFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.6 `NppStatus nppiCrossCorrFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.7 `NppStatus nppiCrossCorrFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.8 `NppStatus nppiCrossCorrFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.9 `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.10 `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.11 `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.12 `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.13 `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.14 `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.15 `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.16 `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.17 `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.18 `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.19 `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.37.2.20 `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38 CrossCorrSame_Norm

Primitives for computing the normalized cross correlation between two images with same mode.

CrossCorrSame_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrSame_Norm_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 32-bit floating point image CrossCorrSame_Norm.
- `NppStatus nppiCrossCorrSame_Norm_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Three-channel 32-bit floating point image CrossCorrSame_Norm.
- `NppStatus nppiCrossCorrSame_Norm_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrSame_Norm.
- `NppStatus nppiCrossCorrSame_Norm_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrSame_Norm ignoring alpha channel.
- `NppStatus nppiCrossCorrSame_Norm_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 8-bit unsigned image CrossCorrSame_Norm.
- `NppStatus nppiCrossCorrSame_Norm_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrSame_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

7.38.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with same mode.

7.38.2 Function Documentation

7.38.2.1 `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.2 `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.3 `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.4 `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.5 `NppStatus nppiCrossCorrSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.6 `NppStatus nppiCrossCorrSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.7 `NppStatus nppiCrossCorrSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.8 `NppStatus nppiCrossCorrSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.9 `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.10 `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.11 `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.12 `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.13 `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.14 `NppStatus nppiCrossCorrSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.15 `NppStatus nppiCrossCorrSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.16 `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38.2.17 `NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.18 `NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.19 `NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.38.2.20 `NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39 CrossCorrValid_Norm

Primitives for computing the normalized cross correlation between two images with valid mode.

CrossCorrValid_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrValid_Norm_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- `NppStatus nppiCrossCorrValid_Norm_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 32-bit floating point image CrossCorrValid_Norm.
- `NppStatus nppiCrossCorrValid_Norm_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Three-channel 32-bit floating point image CrossCorrValid_Norm.
- `NppStatus nppiCrossCorrValid_Norm_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrValid_Norm.
- `NppStatus nppiCrossCorrValid_Norm_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrValid_Norm ignoring alpha channel.
- `NppStatus nppiCrossCorrValid_Norm_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 8-bit unsigned image CrossCorrValid_Norm.
- `NppStatus nppiCrossCorrValid_Norm_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrValid_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

7.39.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with valid mode.

7.39.2 Function Documentation

7.39.2.1 `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.2 `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.3 `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.4 `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.5 `NppStatus nppiCrossCorrValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.6 `NppStatus nppiCrossCorrValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.7 `NppStatus nppiCrossCorrValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.8 `NppStatus nppiCrossCorrValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.9 `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.10 `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.11 `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.12 `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.13 `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.14 `NppStatus nppiCrossCorrValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.15 `NppStatus nppiCrossCorrValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.16 `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.17 `NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.18 `NppStatus nppiCrossCorrValid_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.19 `NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.20 `NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40 CrossCorrValid

Primitives for computing the cross correlation between two images with valid mode.

CrossCorrValid

The functions compute the $R_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrValid_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 32-bit floating point images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_8u32f_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 8-bit unsigned images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_8s32f_C1R` (const `Npp8s *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8s *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 8-bit signed images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)
One-channel 16-bit unsigned images CrossCorrValid.

7.40.1 Detailed Description

Primitives for computing the cross correlation between two images with valid mode.

7.40.2 Function Documentation

- 7.40.2.1** `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`)

One-channel 16-bit unsigned images CrossCorrValid.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.2 `NppStatus nppiCrossCorrValid_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point images CrossCorrValid.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.3 `NppStatus nppiCrossCorrValid_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed images CrossCorrValid.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.4 `NppStatus nppiCrossCorrValid_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned images CrossCorrValid.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41 CrossCorrFull_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

CrossCorrFull_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.
- `NppStatus nppiCrossCorrFull_NormLevel_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 32-bit floating point image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

- **NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.
- **NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Three-channel 8-bit signed image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrFull_NormLevel ignoring alpha channel.
- **NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image CrossCorrFull_NormLevel.
- **NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrFull_NormLevel.
- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

FullNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `CrossCorrFull_NormLevel` primitives.

- `NppStatus nppiFullNormLevelGetBufferHostSize_8u_C1RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C1RSfs`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_8u_C3RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C3RSfs`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_8u_C4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C4RSfs`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_AC4RSfs`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C1R`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C3R`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C4R`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_AC4R`.
- `NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)

- Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8u32f_C1R.*
- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8u32f_C3R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8u32f_C4R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8u32f_AC4R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8s32f_C1R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8s32f_C3R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8s32f_C4R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_8s32f_AC4R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_16u32f_C1R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_16u32f_C3R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_16u32f_C4R.
 - **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for nppiCrossCorrFull_NormLevel_16u32f_AC4R.

7.41.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

7.41.2 Function Documentation

7.41.2.1 `NppStatus nppiCrossCorrFull_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.2 `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.3 NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.4 NppStatus nppiCrossCorrFull_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.5 NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.6 NppStatus nppiCrossCorrFull_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

One-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.7 `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.8 `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.9 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer](#) and [Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.10 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer](#) and [Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.11 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.12 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.13 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.14 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiFullNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.15 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.16 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.17 `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.18 `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.19 `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.20 `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.21 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.22 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.23 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.24 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.25 NppStatus nppiFullNormLevelGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.26 NppStatus nppiFullNormLevelGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.27 NppStatus nppiFullNormLevelGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.28 `NppStatus nppiFullNormLevelGetBufferHostSize_32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.29 `NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_AC4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.30 `NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C1R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.31 `NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C3R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.32 NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.33 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.34 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.35 `NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C3R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.36 `NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.37 `NppStatus nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_AC4RSfs`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.38 `NppStatus nppiFullNormLevelGetBufferHostSize_8u_C1RSfs` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C1RSfs`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.39 NppStatus nppiFullNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.41.2.40 NppStatus nppiFullNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42 CrossCorrSame_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

CrossCorrSame_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.
- `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Three-channel 32-bit floating point image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.
- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
One-channel 8-bit signed image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit signed image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrSame_NormLevel ignoring alpha channel.
- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrSame_NormLevel.
- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

SameNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `CrossCorrSame_`-`NormLevel` primitives.

- `NppStatus nppiSameNormLevelGetBufferHostSize_8u_C1RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_C1RSfs`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_8u_C3RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_C3RSfs`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_8u_C4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_C4RSfs`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_AC4RSfs`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_32f_C1R`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_32f_C3R`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_32f_C4R`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_32f_AC4R`.
- `NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u32f_C1R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u32f_C3R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u32f_C4R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u32f_AC4R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8s32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8s32f_C1R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8s32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8s32f_C3R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8s32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8s32f_C4R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_8s32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8s32f_AC4R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_16u32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_16u32f_C1R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_16u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_16u32f_C3R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_16u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_16u32f_C4R.

- **NppStatus** `nppiSameNormLevelGetBufferHostSize_16u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_16u32f_AC4R.

7.42.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

7.42.2 Function Documentation

7.42.2.1 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.2 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.3 NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp16u * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Three-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.4 NppStatus nppiCrossCorrSame_NormLevel_16u32f_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp16u * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.5 `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.6 `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.7 `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.8 `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.9 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.10 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.11 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.12 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.13 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.14 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.15 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.16 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.17 `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.18 `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.19 `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.20 `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.21 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.22 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.23 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.24 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.25 NppStatus nppiSameNormLevelGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.26 NppStatus nppiSameNormLevelGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.27 NppStatus nppiSameNormLevelGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.28 `NppStatus nppiSameNormLevelGetBufferHostSize_32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_32f_C4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.29 `NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_AC4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.30 `NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C1R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.31 `NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C3R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.32 NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.33 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.34 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.35 `NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C3R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.36 `NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C4R`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.37 `NppStatus nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_AC4RSfs`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

`hpBufferSize` Required buffer size. Important: `hpBufferSize` is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.38 `NppStatus nppiSameNormLevelGetBufferHostSize_8u_C1RSfs` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u_C1RSfs`.

Parameters:

`oSizeROI` [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.39 NppStatus nppiSameNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.42.2.40 NppStatus nppiSameNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43 CrossCorrValid_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

CrossCorrValid_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
One-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp8u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp8u *pDst`, int `nDstStep`, int `nScaleFactor`, `Npp8u *pDeviceBuffer`)
Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.
- `NppStatus nppiCrossCorrValid_NormLevel_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
One-channel 32-bit floating point image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Three-channel 32-bit floating point image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp32f *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 32-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

- [NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.
- [NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit signed image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit signed image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image CrossCorrValid_NormLevel ignoring alpha channel.
- [NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit unsigned image CrossCorrValid_NormLevel.
- [NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrValid_NormLevel.
- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSrcRoiSize`, const `Npp16u *pTpl`, int `nTplStep`, `NppiSize` `oTplRoiSize`, `Npp32f *pDst`, int `nDstStep`, `Npp8u *pDeviceBuffer`)
Four-channel 16-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

ValidNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `CrossCorrValid_` `NormLevel` primitives.

- `NppStatus nppiValidNormLevelGetBufferHostSize_8u_C1RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C1RSfs.
- `NppStatus nppiValidNormLevelGetBufferHostSize_8u_C3RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C3RSfs.
- `NppStatus nppiValidNormLevelGetBufferHostSize_8u_C4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C4RSfs.
- `NppStatus nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_AC4RSfs.
- `NppStatus nppiValidNormLevelGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C1R.
- `NppStatus nppiValidNormLevelGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C3R.
- `NppStatus nppiValidNormLevelGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C4R.
- `NppStatus nppiValidNormLevelGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)
Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_AC4R.
- `NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u32f_C1R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u32f_C3R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u32f_C4R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u32f_AC4R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8s32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8s32f_C1R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8s32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8s32f_C3R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8s32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8s32f_C4R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_8s32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8s32f_AC4R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_16u32f_C1R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_16u32f_C1R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_16u32f_C3R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_16u32f_C3R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_16u32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_16u32f_C4R.

- **NppStatus** `nppiValidNormLevelGetBufferHostSize_16u32f_AC4R` (NppiSize oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_16u32f_AC4R.

7.43.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

7.43.2 Function Documentation

7.43.2.1 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiValidNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.2 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiValidNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.3 NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.4 NppStatus nppiCrossCorrValid_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.5 NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp32f * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.6 NppStatus nppiCrossCorrValid_NormLevel_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp32f * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

One-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.7 `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R` (`const Npp32f * pSrc`, `int nSrcStep`, `NppiSize oSrcRoiSize`, `const Npp32f * pTpl`, `int nTplStep`, `NppiSize oTplRoiSize`, `Npp32f * pDst`, `int nDstStep`, `Npp8u * pDeviceBuffer`)

Three-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.8 `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R` (`const Npp32f * pSrc`, `int nSrcStep`, `NppiSize oSrcRoiSize`, `const Npp32f * pTpl`, `int nTplStep`, `NppiSize oTplRoiSize`, `Npp32f * pDst`, `int nDstStep`, `Npp8u * pDeviceBuffer`)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.9 NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.10 NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)

One-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.11 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.12 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.13 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.14 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.15 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.16 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.17 `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.18 `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.19 `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.20 `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.43.2.21 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.22 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.23 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.24 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.25 NppStatus nppiValidNormLevelGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.26 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.27 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.28 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.29 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.30 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.31 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.32 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.33 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.34 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.35 `NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.36 `NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.37 `NppStatus nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_AC4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.38 `NppStatus nppiValidNormLevelGetBufferHostSize_8u_C1RSfs (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C1RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.39 NppStatus nppiValidNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.43.2.40 NppStatus nppiValidNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44 Image Quality Index

Primitives for computing the image quality index of two images.

QualityIndex

Given two images M and N (both $W \times H$), the mathematical formula to calculate the image quality index Q between them is expressed as:

$$Q = \frac{4\sigma_{MN}\tilde{M}\tilde{N}}{[(\tilde{M}^2) + (\tilde{N}^2)][(\sigma_M)^2 + (\sigma_N)^2]}$$

where

$$\tilde{M} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} M(j, i)$$

$$\tilde{N} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} N(j, i)$$

$$\sigma_M = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}]^2}$$

$$\sigma_N = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [N(j, i) - \tilde{N}]^2}$$

$$\sigma_{MN} = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}][N(j, i) - \tilde{N}]$$

The functions require additional scratch buffer for computations.

- [NppStatus nppiQualityIndex_8u32f_C1R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit unsigned image QualityIndex.
- [NppStatus nppiQualityIndex_16u32f_C1R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit unsigned image QualityIndex.
- [NppStatus nppiQualityIndex_32f_C1R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit floating point image QualityIndex.
- [NppStatus nppiQualityIndex_8u32f_C3R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit unsigned image QualityIndex.
- [NppStatus nppiQualityIndex_16u32f_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image QualityIndex.

- **NppStatus** `nppiQualityIndex_32f_C3R` (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image QualityIndex.
- **NppStatus** `nppiQualityIndex_8u32f_AC4R` (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image QualityIndex.
- **NppStatus** `nppiQualityIndex_16u32f_AC4R` (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image QualityIndex.
- **NppStatus** `nppiQualityIndex_32f_AC4R` (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 32-bit floating point image QualityIndex.

QualityIndexGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the `QualityIndex` primitives.

- **NppStatus** `nppiQualityIndexGetBufferHostSize_8u32f_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_8u32f_C1R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_16u32f_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_16u32f_C1R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_32f_C1R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_32f_C1R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_8u32f_C3R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_8u32f_C3R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_16u32f_C3R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_16u32f_C3R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_32f_C3R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_32f_C3R`.
- **NppStatus** `nppiQualityIndexGetBufferHostSize_8u32f_AC4R` (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size (in bytes) for `nppiQualityIndex_8u32f_AC4R`.

- `NppStatus nppiQualityIndexGetBufferHostSize_16u32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size (in bytes) for `nppiQualityIndex_16u32f_AC4R`.
- `NppStatus nppiQualityIndexGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)
Buffer size (in bytes) for `nppiQualityIndex_32f_AC4R`.

7.44.1 Detailed Description

Primitives for computing the image quality index of two images.

7.44.2 Function Documentation

7.44.2.1 `NppStatus nppiQualityIndex_16u32f_AC4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `const Npp16u *pSrc2`, `int nSrc2Step`, `NppiSize oRoiSize`, `Npp32f *pDst`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit unsigned image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Pointer to the quality index](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
 Use [nppiQualityIndexGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.2 `NppStatus nppiQualityIndex_16u32f_C1R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `const Npp16u *pSrc2`, `int nSrc2Step`, `NppiSize oRoiSize`, `Npp32f *pDst`, `Npp8u *pDeviceBuffer`)

One-channel 16-bit unsigned image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.3 `NppStatus nppiQualityIndex_16u32f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.4 `NppStatus nppiQualityIndex_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.5 `NppStatus nppiQualityIndex_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.6 `NppStatus nppiQualityIndex_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.7 `NppStatus nppiQualityIndex_8u32f_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiQualityIndexGetBufferHostSize_8u32f_AC4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.8 `NppStatus nppiQualityIndex_8u32f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiQualityIndexGetBufferHostSize_8u32f_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.9 `NppStatus nppiQualityIndex_8u32f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.44.2.10 `NppStatus nppiQualityIndexGetBufferHostSize_16u32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.11 `NppStatus nppiQualityIndexGetBufferHostSize_16u32f_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_C1R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.12 NppStatus nppiQualityIndexGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.13 NppStatus nppiQualityIndexGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.14 NppStatus nppiQualityIndexGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.15 NppStatus nppiQualityIndexGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.16 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.17 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.44.2.18 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.45 MaximumError

Primitives for computing the maximum error between two images.

Functions

- `NppStatus nppiMaximumError_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_8s_C1R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_16sc_C1R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed complex image Maximum_Error.
- `NppStatus nppiMaximumError_32u_C1R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_32sc_C1R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed complex image Maximum_Error.
- `NppStatus nppiMaximumError_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image Maximum_Error.
- `NppStatus nppiMaximumError_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point complex image Maximum_Error.
- `NppStatus nppiMaximumError_64f_C1R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 64-bit floating point image Maximum_Error.

- [NppStatus nppiMaximumError_8u_C2R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_8s_C2R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_16u_C2R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_16s_C2R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_16sc_C2R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed complex image Maximum_Error.
- [NppStatus nppiMaximumError_32u_C2R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_32s_C2R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_32sc_C2R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed complex image Maximum_Error.
- [NppStatus nppiMaximumError_32f_C2R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point image Maximum_Error.
- [NppStatus nppiMaximumError_32fc_C2R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point complex image Maximum_Error.
- [NppStatus nppiMaximumError_64f_C2R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 64-bit floating point image Maximum_Error.
- [NppStatus nppiMaximumError_8u_C3R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_8s_C3R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_16s_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_16sc_C3R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed complex image Maximum_Error.
- `NppStatus nppiMaximumError_32u_C3R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_32sc_C3R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit signed complex image Maximum_Error.
- `NppStatus nppiMaximumError_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image Maximum_Error.
- `NppStatus nppiMaximumError_32fc_C3R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point complex image Maximum_Error.
- `NppStatus nppiMaximumError_64f_C3R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 64-bit floating point image Maximum_Error.
- `NppStatus nppiMaximumError_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_8s_C4R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit signed image Maximum_Error.
- `NppStatus nppiMaximumError_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Maximum_Error.
- `NppStatus nppiMaximumError_16s_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_16sc_C4R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32u_C4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_32sc_C4R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image Maximum_Error.

- `NppStatus nppiMaximumError_32fc_C4R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point complex image Maximum_Error.

- `NppStatus nppiMaximumError_64f_C4R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 64-bit floating point image Maximum_Error.

7.45.1 Detailed Description

Primitives for computing the maximum error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum error is defined as the largest absolute difference between pixels of two images. If the image is in complex format, the absolute value of the complex number is provided.

7.45.2 Function Documentation

- ### 7.45.2.1 `NppStatus nppiMaximumError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

One-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `npippiMaximumErrorGetBufferHostSize_16s_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.2 NppStatus npippiMaximumError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Two-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `npippiMaximumErrorGetBufferHostSize_16s_C2R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.3 NppStatus npippiMaximumError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Three-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `npippiMaximumErrorGetBufferHostSize_16s_C3R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.4 NppStatus nppiMaximumError_16s_C4R (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pError*, Npp8u * *pDeviceBuffer*)

Four-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.5 NppStatus nppiMaximumError_16sc_C1R (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pError*, Npp8u * *pDeviceBuffer*)

One-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.6 NppStatus nppiMaximumError_16sc_C2R (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pError*, Npp8u * *pDeviceBuffer*)

Two-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.7 `NppStatus nppiMaximumError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.8 `NppStatus nppiMaximumError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.9 `NppStatus nppiMaximumError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.10 `NppStatus nppiMaximumError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.11 `NppStatus nppiMaximumError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.12 `NppStatus nppiMaximumError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.13 `NppStatus nppiMaximumError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Maximum_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pError* Pointer to the computed error.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.14 `NppStatus nppiMaximumError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Maximum_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pError* Pointer to the computed error.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.15 `NppStatus nppiMaximumError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Maximum_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.16 `NppStatus nppiMaximumError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.17 `NppStatus nppiMaximumError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.18 `NppStatus nppiMaximumError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.19 `NppStatus nppiMaximumError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.20 `NppStatus nppiMaximumError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiMaximumErrorGetBufferHostSize_32f_C4R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.21 `NppStatus nppiMaximumError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiMaximumErrorGetBufferHostSize_16s_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.22 `NppStatus nppiMaximumError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.23 `NppStatus nppiMaximumError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.24 `NppStatus nppiMaximumError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.25 `NppStatus nppiMaximumError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.26 `NppStatus nppiMaximumError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.27 `NppStatus nppiMaximumError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.28 `NppStatus nppiMaximumError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.29 `NppStatus nppiMaximumError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.30 `NppStatus nppiMaximumError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.31 `NppStatus nppiMaximumError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.32 `NppStatus nppiMaximumError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.33 `NppStatus nppiMaximumError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.34 `NppStatus nppiMaximumError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.35 `NppStatus nppiMaximumError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.36 `NppStatus nppiMaximumError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.45.2.37 `NppStatus nppiMaximumError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.38 `NppStatus nppiMaximumError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.39 `NppStatus nppiMaximumError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.40 `NppStatus nppiMaximumError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.41 `NppStatus nppiMaximumError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.42 `NppStatus nppiMaximumError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.43 `NppStatus nppiMaximumError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.45.2.44 `NppStatus nppiMaximumError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46 AverageError

Primitives for computing the average error between two images.

Functions

- [NppStatus nppiAverageError_8u_C1R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_8s_C1R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit signed image Average_Error.
- [NppStatus nppiAverageError_16u_C1R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_16s_C1R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit signed image Average_Error.
- [NppStatus nppiAverageError_16sc_C1R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32u_C1R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_32s_C1R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit signed image Average_Error.
- [NppStatus nppiAverageError_32sc_C1R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32f_C1R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit floating point image Average_Error.
- [NppStatus nppiAverageError_32fc_C1R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit floating point complex image Average_Error.
- [NppStatus nppiAverageError_64f_C1R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
One-channel 64-bit floating point image Average_Error.

- `NppStatus nppiAverageError_8u_C2R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit unsigned image Average_Error.
- `NppStatus nppiAverageError_8s_C2R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit signed image Average_Error.
- `NppStatus nppiAverageError_16u_C2R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit unsigned image Average_Error.
- `NppStatus nppiAverageError_16s_C2R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed image Average_Error.
- `NppStatus nppiAverageError_16sc_C2R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed complex image Average_Error.
- `NppStatus nppiAverageError_32u_C2R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit unsigned image Average_Error.
- `NppStatus nppiAverageError_32s_C2R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed image Average_Error.
- `NppStatus nppiAverageError_32sc_C2R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed complex image Average_Error.
- `NppStatus nppiAverageError_32f_C2R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point image Average_Error.
- `NppStatus nppiAverageError_32fc_C2R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point complex image Average_Error.
- `NppStatus nppiAverageError_64f_C2R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 64-bit floating point image Average_Error.
- `NppStatus nppiAverageError_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image Average_Error.
- `NppStatus nppiAverageError_8s_C3R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit signed image Average_Error.

- [NppStatus nppiAverageError_16u_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_16s_C3R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed image Average_Error.
- [NppStatus nppiAverageError_16sc_C3R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32u_C3R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_32s_C3R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed image Average_Error.
- [NppStatus nppiAverageError_32sc_C3R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32f_C3R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point image Average_Error.
- [NppStatus nppiAverageError_32fc_C3R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point complex image Average_Error.
- [NppStatus nppiAverageError_64f_C3R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 64-bit floating point image Average_Error.
- [NppStatus nppiAverageError_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_8s_C4R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image Average_Error.
- [NppStatus nppiAverageError_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image Average_Error.

- `NppStatus nppiAverageError_16sc_C4R` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed complex image Average_Error.

- `NppStatus nppiAverageError_32u_C4R` (const `Npp32u *pSrc1`, int `nSrc1Step`, const `Npp32u *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit unsigned image Average_Error.

- `NppStatus nppiAverageError_32s_C4R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed image Average_Error.

- `NppStatus nppiAverageError_32sc_C4R` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed complex image Average_Error.

- `NppStatus nppiAverageError_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image Average_Error.

- `NppStatus nppiAverageError_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point complex image Average_Error.

- `NppStatus nppiAverageError_64f_C4R` (const `Npp64f *pSrc1`, int `nSrc1Step`, const `Npp64f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 64-bit floating point image Average_Error.

7.46.1 Detailed Description

Primitives for computing the average error between two images.

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the average error is defined as:

$$AverageError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

where N stands for the number of channels. If the image is in complex format, the absolute value is used for computation.

7.46.2 Function Documentation

- 7.46.2.1 `NppStatus nppiAverageError_16s_C1R`** (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

One-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.2 `NppStatus nppiAverageError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.3 `NppStatus nppiAverageError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.4 `NppStatus nppiAverageError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.5 `NppStatus nppiAverageError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.6 `NppStatus nppiAverageError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.7 `NppStatus nppiAverageError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.8 `NppStatus nppiAverageError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.9 `NppStatus nppiAverageError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.10 `NppStatus nppiAverageError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.11 `NppStatus nppiAverageError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.12 `NppStatus nppiAverageError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.13 `NppStatus nppiAverageError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.14 `NppStatus nppiAverageError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.15 `NppStatus nppiAverageError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Average_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pError* Pointer to the computed error.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.16 `NppStatus nppiAverageError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Average_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pError* Pointer to the computed error.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.17 `NppStatus nppiAverageError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image Average_Error.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.18 `NppStatus nppiAverageError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.19 `NppStatus nppiAverageError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.20 `NppStatus nppiAverageError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.21 `NppStatus nppiAverageError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.22 `NppStatus nppiAverageError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.23 `NppStatus nppiAverageError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.24 `NppStatus nppiAverageError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.25 `NppStatus nppiAverageError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.26 `NppStatus nppiAverageError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.27 `NppStatus nppiAverageError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.28 `NppStatus nppiAverageError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.29 `NppStatus nppiAverageError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.30 `NppStatus nppiAverageError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.31 `NppStatus nppiAverageError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.32 `NppStatus nppiAverageError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.33 `NppStatus nppiAverageError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.34 `NppStatus nppiAverageError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.35 `NppStatus nppiAverageError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.36 `NppStatus nppiAverageError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.46.2.37 `NppStatus nppiAverageError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.38 `NppStatus nppiAverageError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.39 `NppStatus nppiAverageError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.40 `NppStatus nppiAverageError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.41 `NppStatus nppiAverageError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.42 `NppStatus nppiAverageError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.43 `NppStatus nppiAverageError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.46.2.44 `NppStatus nppiAverageError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47 MaximumRelativeError

Primitives for computing the maximum relative error between two images.

Functions

- `NppStatus nppiMaximumRelativeError_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C1R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16sc_C1R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32u_C1R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32sc_C1R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_64f_C1R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 64-bit floating point image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_8u_C2R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C2R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16u_C2R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C2R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16sc_C2R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32u_C2R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32s_C2R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32sc_C2R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32f_C2R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32fc_C2R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_64f_C2R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 64-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C3R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit signed image MaximumRelative_Error.

- [NppStatus nppiMaximumRelativeError_16u_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_16s_C3R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_16sc_C3R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_32u_C3R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_32s_C3R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_32sc_C3R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_32f_C3R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_32fc_C3R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point complex image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_64f_C3R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 64-bit floating point image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_8s_C4R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiMaximumRelativeError_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_16sc_C4R` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32u_C4R` (const `Npp32u *pSrc1`, int `nSrc1Step`, const `Npp32u *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit unsigned image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32s_C4R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32sc_C4R` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_64f_C4R` (const `Npp64f *pSrc1`, int `nSrc1Step`, const `Npp64f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 64-bit floating point image MaximumRelative_Error.

7.47.1 Detailed Description

Primitives for computing the maximum relative error between two images.

Given two images `pSrc1` and `pSrc2` both with width `W` and height `H`, the maximum relative error is defined as:

$$\text{MaximumRelativeError} = \max \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

If the image is in complex format, the absolute value is used for computation. For multiple channels, the maximum relative error of all the channels is returned.

7.47.2 Function Documentation

- ### 7.47.2.1 `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

One-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.2 `NppStatus nppiMaximumRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.3 `NppStatus nppiMaximumRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.4 `NppStatus nppiMaximumRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.5 `NppStatus nppiMaximumRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.6 `NppStatus nppiMaximumRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.7 `NppStatus nppiMaximumRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.8 `NppStatus nppiMaximumRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.9 `NppStatus nppiMaximumRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.10 `NppStatus nppiMaximumRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.11 `NppStatus nppiMaximumRelativeError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.12 `NppStatus nppiMaximumRelativeError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.13 `NppStatus nppiMaximumRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.14 `NppStatus nppiMaximumRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.15 `NppStatus nppiMaximumRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.16 `NppStatus nppiMaximumRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.17 `NppStatus nppiMaximumRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.18 `NppStatus nppiMaximumRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.19 `NppStatus nppiMaximumRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.20 `NppStatus nppiMaximumRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.21 `NppStatus nppiMaximumRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.22 `NppStatus nppiMaximumRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.23 `NppStatus nppiMaximumRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.24 `NppStatus nppiMaximumRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.25 `NppStatus nppiMaximumRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.26 `NppStatus nppiMaximumRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.27 `NppStatus nppiMaximumRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.28 `NppStatus nppiMaximumRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.29 `NppStatus nppiMaximumRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.30 `NppStatus nppiMaximumRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.31 `NppStatus nppiMaximumRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.32 `NppStatus nppiMaximumRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.33 `NppStatus nppiMaximumRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.34 `NppStatus nppiMaximumRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.35 `NppStatus nppiMaximumRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image `MaximumRelative_Error`.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.36 `NppStatus nppiMaximumRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image `MaximumRelative_Error`.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.47.2.37 `NppStatus nppiMaximumRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.38 `NppStatus nppiMaximumRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.39 `NppStatus nppiMaximumRelativeError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.40 `NppStatus nppiMaximumRelativeError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.41 `NppStatus nppiMaximumRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.42 `NppStatus nppiMaximumRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.43 `NppStatus nppiMaximumRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.47.2.44 `NppStatus nppiMaximumRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48 AverageRelativeError

Primitives for computing the average relative error between two images.

Functions

- `NppStatus nppiAverageRelativeError_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_8s_C1R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_16sc_C1R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_32u_C1R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_32sc_C1R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiAverageRelativeError_64f_C1R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 64-bit floating point image MaximumRelative_Error.

- [NppStatus nppiAverageRelativeError_8u_C2R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_8s_C2R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16u_C2R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16s_C2R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16sc_C2R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32u_C2R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32s_C2R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32sc_C2R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32f_C2R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32fc_C2R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_64f_C2R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 64-bit floating point image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_8u_C3R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_8s_C3R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit signed image MaximumRelative_Error.

- **NppStatus** [nppiAverageRelativeError_16u_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_16s_C3R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_16sc_C3R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed complex image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_32u_C3R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit unsigned image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_32s_C3R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_32sc_C3R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed complex image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_32f_C3R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_32fc_C3R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point complex image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_64f_C3R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 64-bit floating point image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_8s_C4R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image MaximumRelative_Error.
- **NppStatus** [nppiAverageRelativeError_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_16sc_C4R` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 16-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_32u_C4R` (const `Npp32u *pSrc1`, int `nSrc1Step`, const `Npp32u *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit unsigned image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_32s_C4R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_32sc_C4R` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 32-bit floating point complex image MaximumRelative_Error.

- `NppStatus nppiAverageRelativeError_64f_C4R` (const `Npp64f *pSrc1`, int `nSrc1Step`, const `Npp64f *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

Four-channel 64-bit floating point image MaximumRelative_Error.

7.48.1 Detailed Description

Primitives for computing the average relative error between two images.

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the maximum relative error is defined as:

$$AverageRelativeError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

where N is the number of channels. If the image is in complex format, the absolute value is used for computation.

7.48.2 Function Documentation

- ### 7.48.2.1 `NppStatus nppiAverageRelativeError_16s_C1R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `NppiSize oSizeROI`, `Npp64f *pError`, `Npp8u *pDeviceBuffer`)

One-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.2 `NppStatus nppiAverageRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.3 `NppStatus nppiAverageRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.4 `NppStatus nppiAverageRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.5 `NppStatus nppiAverageRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.6 NppStatus nppiAverageRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Two-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.7 NppStatus nppiAverageRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Three-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.8 NppStatus nppiAverageRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Four-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.9 NppStatus nppiAverageRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.10 NppStatus nppiAverageRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Two-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.11 `NppStatus nppiAverageRelativeError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.12 `NppStatus nppiAverageRelativeError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.13 `NppStatus nppiAverageRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.14 `NppStatus nppiAverageRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.15 `NppStatus nppiAverageRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.16 `NppStatus nppiAverageRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.17 `NppStatus nppiAverageRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.18 `NppStatus nppiAverageRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.19 `NppStatus nppiAverageRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.20 `NppStatus nppiAverageRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.21 `NppStatus nppiAverageRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.22 `NppStatus nppiAverageRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.23 `NppStatus nppiAverageRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.24 `NppStatus nppiAverageRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.25 `NppStatus nppiAverageRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.26 `NppStatus nppiAverageRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.27 `NppStatus nppiAverageRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.28 `NppStatus nppiAverageRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.29 `NppStatus nppiAverageRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.30 `NppStatus nppiAverageRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.31 `NppStatus nppiAverageRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.32 `NppStatus nppiAverageRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.33 `NppStatus nppiAverageRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.34 `NppStatus nppiAverageRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.35 `NppStatus nppiAverageRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image `MaximumRelative_Error`.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.48.2.36 `NppStatus nppiAverageRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image `MaximumRelative_Error`.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.48.2.37 `NppStatus nppiAverageRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.38 `NppStatus nppiAverageRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.39 `NppStatus nppiAverageRelativeError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.40 `NppStatus nppiAverageRelativeError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.41 `NppStatus nppiAverageRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.42 `NppStatus nppiAverageRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.43 `NppStatus nppiAverageRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.48.2.44 `NppStatus nppiAverageRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.49 Linear Transforms

Linear image transformations.

Modules

- [Fourier Transforms](#)

7.49.1 Detailed Description

Linear image transformations.

These functions can be found in either the nppi or nppist libraries. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

7.50 Fourier Transforms

Functions

- `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.

7.50.1 Function Documentation

7.50.1.1 `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](#)

7.50.1.2 `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

Chapter 8

Data Structure Documentation

8.1 NPP_ALIGN_16 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.
- [Npp64f re](#)
Real part.
- [Npp64f im](#)
Imaginary part.

8.1.1 Detailed Description

Complex Number This struct represents a long long complex number.

Complex Number This struct represents a double floating-point complex number.

8.1.2 Field Documentation

8.1.2.1 Npp64f NPP_ALIGN_16::im

Imaginary part.

8.1.2.2 Npp64s NPP_ALIGN_16::im

Imaginary part.

8.1.2.3 Npp64f NPP_ALIGN_16::re

Real part.

8.1.2.4 Npp64s NPP_ALIGN_16::re

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h

8.2 NPP_ALIGN_8 Struct Reference

Complex Number This struct represents an unsigned int complex number.

```
#include <nppdefs.h>
```

Data Fields

- [Npp32u re](#)
Real part.
- [Npp32u im](#)
Imaginary part.
- [Npp32s re](#)
Real part.
- [Npp32s im](#)
Imaginary part.
- [Npp32f re](#)
Real part.
- [Npp32f im](#)
Imaginary part.

8.2.1 Detailed Description

Complex Number This struct represents an unsigned int complex number.

Complex Number This struct represents a single floating-point complex number.

Complex Number This struct represents a signed int complex number.

8.2.2 Field Documentation

8.2.2.1 Npp32f NPP_ALIGN_8::im

Imaginary part.

8.2.2.2 Npp32s NPP_ALIGN_8::im

Imaginary part.

8.2.2.3 Npp32u NPP_ALIGN_8::im

Imaginary part.

8.2.2.4 Npp32f NPP_ALIGN_8::re

Real part.

8.2.2.5 Npp32s NPP_ALIGN_8::re

Real part.

8.2.2.6 Npp32u NPP_ALIGN_8::re

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h

8.3 NppiHaarBuffer Struct Reference

```
#include <nppdefs.h>
```

Data Fields

- int `haarBufferSize`
size of the buffer
- `Npp32s * haarBuffer`
buffer

8.3.1 Field Documentation

8.3.1.1 `Npp32s* NppiHaarBuffer::haarBuffer`

`buffer`

8.3.1.2 `int NppiHaarBuffer::haarBufferSize`

size of the buffer

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

8.4 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`

8.4.1 Field Documentation

8.4.1.1 `Npp32s* NppiHaarClassifier_32f::classifiers`

packed classifier data 40 bytes each

8.4.1.2 `NppiSize NppiHaarClassifier_32f::classifierSize`

8.4.1.3 `size_t NppiHaarClassifier_32f::classifierStep`

8.4.1.4 `Npp32s* NppiHaarClassifier_32f::counterDevice`

8.4.1.5 `int NppiHaarClassifier_32f::numClassifiers`

number of classifiers

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

8.5 NppiPoint Struct Reference

2D Point

```
#include <nppdefs.h>
```

Data Fields

- `int x`
x-coordinate.
- `int y`
y-coordinate.

8.5.1 Detailed Description

2D Point

8.5.2 Field Documentation

8.5.2.1 `int NppiPoint::x`

x-coordinate.

8.5.2.2 `int NppiPoint::y`

y-coordinate.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

8.6 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 (lowest memory address).
- `int y`
y-coordinate of upper left corner (lowest memory address).
- `int width`
Rectangle width.
- `int height`
Rectangle height.

8.6.1 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.

8.6.2 Field Documentation

8.6.2.1 `int NppiRect::height`

Rectangle height.

8.6.2.2 `int NppiRect::width`

Rectangle width.

8.6.2.3 `int NppiRect::x`

x-coordinate of upper left corner (lowest memory address).

8.6.2.4 `int NppiRect::y`

y-coordinate of upper left corner (lowest memory address).

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

8.7 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.

8.7.1 Detailed Description

2D Size This struct typically represents the size of a rectangular region in two space.

8.7.2 Field Documentation

8.7.2.1 `int NppiSize::height`

Rectangle height.

8.7.2.2 `int NppiSize::width`

Rectangle width.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

8.8 NppLibraryVersion Struct Reference

```
#include <nppdefs.h>
```

Data Fields

- int `major`
Major version number.
- int `minor`
Minor version number.
- int `build`
Build number.

8.8.1 Field Documentation

8.8.1.1 int NppLibraryVersion::build

Build number.

This reflects the nightly build this release was made from.

8.8.1.2 int NppLibraryVersion::major

Major version number.

8.8.1.3 int NppLibraryVersion::minor

Minor version number.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r8.0/NPP/npp/include/nppdefs.h`

Index

- `__align__`
 - `npp_basic_types`, 48, 49
- AverageError, 704
- AverageRelativeError, 751
- Basic NPP Data Types, 46
- build
 - `NppLibraryVersion`, 788
- classifiers
 - `NppiHaarClassifier_32f`, 784
- classifierSize
 - `NppiHaarClassifier_32f`, 784
- classifierStep
 - `NppiHaarClassifier_32f`, 784
- core_npp
 - `nppGetGpuComputeCapability`, 28
 - `nppGetGpuDeviceProperties`, 28
 - `nppGetGpuName`, 28
 - `nppGetGpuNumSMs`, 28
 - `nppGetLibVersion`, 28
 - `nppGetMaxThreadsPerBlock`, 29
 - `nppGetMaxThreadsPerSM`, 29
 - `nppGetStream`, 29
 - `nppGetStreamMaxThreadsPerSM`, 29
 - `nppGetStreamNumSMs`, 29
 - `nppSetStream`, 29
- counterDevice
 - `NppiHaarClassifier_32f`, 784
- CountInRange., 483
- CrossCorrFull_Norm, 576
- CrossCorrFull_NormLevel, 612
- crosscorrfullnorm
 - `nppiCrossCorrFull_Norm_16u32f_AC4R`, 578
 - `nppiCrossCorrFull_Norm_16u32f_C1R`, 578
 - `nppiCrossCorrFull_Norm_16u32f_C3R`, 578
 - `nppiCrossCorrFull_Norm_16u32f_C4R`, 579
 - `nppiCrossCorrFull_Norm_32f_AC4R`, 579
 - `nppiCrossCorrFull_Norm_32f_C1R`, 580
 - `nppiCrossCorrFull_Norm_32f_C3R`, 580
 - `nppiCrossCorrFull_Norm_32f_C4R`, 581
 - `nppiCrossCorrFull_Norm_8s32f_AC4R`, 581
 - `nppiCrossCorrFull_Norm_8s32f_C1R`, 581
 - `nppiCrossCorrFull_Norm_8s32f_C3R`, 582
 - `nppiCrossCorrFull_Norm_8s32f_C4R`, 582
 - `nppiCrossCorrFull_Norm_8u32f_AC4R`, 583
 - `nppiCrossCorrFull_Norm_8u32f_C1R`, 583
 - `nppiCrossCorrFull_Norm_8u32f_C3R`, 584
 - `nppiCrossCorrFull_Norm_8u32f_C4R`, 584
 - `nppiCrossCorrFull_Norm_8u_AC4RSfs`, 584
 - `nppiCrossCorrFull_Norm_8u_C1RSfs`, 585
 - `nppiCrossCorrFull_Norm_8u_C3RSfs`, 585
 - `nppiCrossCorrFull_Norm_8u_C4RSfs`, 586
- crosscorrfullnormlevel
 - `nppiCrossCorrFull_NormLevel_16u32f_-AC4R`, 616
 - `nppiCrossCorrFull_NormLevel_16u32f_C1R`, 616
 - `nppiCrossCorrFull_NormLevel_16u32f_C3R`, 616
 - `nppiCrossCorrFull_NormLevel_16u32f_C4R`, 617
 - `nppiCrossCorrFull_NormLevel_32f_AC4R`, 617
 - `nppiCrossCorrFull_NormLevel_32f_C1R`, 618
 - `nppiCrossCorrFull_NormLevel_32f_C3R`, 618
 - `nppiCrossCorrFull_NormLevel_32f_C4R`, 619
 - `nppiCrossCorrFull_NormLevel_8s32f_AC4R`, 619
 - `nppiCrossCorrFull_NormLevel_8s32f_C1R`, 620
 - `nppiCrossCorrFull_NormLevel_8s32f_C3R`, 620
 - `nppiCrossCorrFull_NormLevel_8s32f_C4R`, 621
 - `nppiCrossCorrFull_NormLevel_8u32f_AC4R`, 621
 - `nppiCrossCorrFull_NormLevel_8u32f_C1R`, 622
 - `nppiCrossCorrFull_NormLevel_8u32f_C3R`, 622
 - `nppiCrossCorrFull_NormLevel_8u32f_C4R`, 623
 - `nppiCrossCorrFull_NormLevel_8u_AC4RSfs`, 623
 - `nppiCrossCorrFull_NormLevel_8u_C1RSfs`, 624
 - `nppiCrossCorrFull_NormLevel_8u_C3RSfs`, 624

- nppiCrossCorrFull_NormLevel_8u_C4RSfs, 625
- nppiFullNormLevelGetBufferHostSize_16u32f_AC4R, 625
- nppiFullNormLevelGetBufferHostSize_16u32f_C1R, 626
- nppiFullNormLevelGetBufferHostSize_16u32f_C3R, 626
- nppiFullNormLevelGetBufferHostSize_16u32f_C4R, 626
- nppiFullNormLevelGetBufferHostSize_32f_AC4R, 627
- nppiFullNormLevelGetBufferHostSize_32f_C1R, 627
- nppiFullNormLevelGetBufferHostSize_32f_C3R, 627
- nppiFullNormLevelGetBufferHostSize_32f_C4R, 627
- nppiFullNormLevelGetBufferHostSize_8s32f_AC4R, 628
- nppiFullNormLevelGetBufferHostSize_8s32f_C1R, 628
- nppiFullNormLevelGetBufferHostSize_8s32f_C3R, 628
- nppiFullNormLevelGetBufferHostSize_8s32f_C4R, 629
- nppiFullNormLevelGetBufferHostSize_8u32f_AC4R, 629
- nppiFullNormLevelGetBufferHostSize_8u32f_C1R, 629
- nppiFullNormLevelGetBufferHostSize_8u32f_C3R, 629
- nppiFullNormLevelGetBufferHostSize_8u32f_C4R, 630
- nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs, 630
- nppiFullNormLevelGetBufferHostSize_8u_C1RSfs, 630
- nppiFullNormLevelGetBufferHostSize_8u_C3RSfs, 631
- nppiFullNormLevelGetBufferHostSize_8u_C4RSfs, 631
- CrossCorrSame_Norm, 587
- CrossCorrSame_NormLevel, 632
- crosscorrmenorm
 - nppiCrossCorrSame_Norm_16u32f_AC4R, 589
 - nppiCrossCorrSame_Norm_16u32f_C1R, 589
 - nppiCrossCorrSame_Norm_16u32f_C3R, 589
 - nppiCrossCorrSame_Norm_16u32f_C4R, 590
 - nppiCrossCorrSame_Norm_32f_AC4R, 590
 - nppiCrossCorrSame_Norm_32f_C1R, 591
 - nppiCrossCorrSame_Norm_32f_C3R, 591
 - nppiCrossCorrSame_Norm_32f_C4R, 592
 - nppiCrossCorrSame_Norm_8s32f_AC4R, 592
 - nppiCrossCorrSame_Norm_8s32f_C1R, 592
 - nppiCrossCorrSame_Norm_8s32f_C3R, 593
 - nppiCrossCorrSame_Norm_8s32f_C4R, 593
 - nppiCrossCorrSame_Norm_8u32f_AC4R, 594
 - nppiCrossCorrSame_Norm_8u32f_C1R, 594
 - nppiCrossCorrSame_Norm_8u32f_C3R, 595
 - nppiCrossCorrSame_Norm_8u32f_C4R, 595
 - nppiCrossCorrSame_Norm_8u_AC4RSfs, 595
 - nppiCrossCorrSame_Norm_8u_C1RSfs, 596
 - nppiCrossCorrSame_Norm_8u_C3RSfs, 596
 - nppiCrossCorrSame_Norm_8u_C4RSfs, 597
- crosscorrmenormlevel
 - nppiCrossCorrSame_NormLevel_16u32f_AC4R, 636
 - nppiCrossCorrSame_NormLevel_16u32f_C1R, 636
 - nppiCrossCorrSame_NormLevel_16u32f_C3R, 636
 - nppiCrossCorrSame_NormLevel_16u32f_C4R, 637
 - nppiCrossCorrSame_NormLevel_32f_AC4R, 637
 - nppiCrossCorrSame_NormLevel_32f_C1R, 638
 - nppiCrossCorrSame_NormLevel_32f_C3R, 638
 - nppiCrossCorrSame_NormLevel_32f_C4R, 639
 - nppiCrossCorrSame_NormLevel_8s32f_AC4R, 639
 - nppiCrossCorrSame_NormLevel_8s32f_C1R, 640
 - nppiCrossCorrSame_NormLevel_8s32f_C3R, 640
 - nppiCrossCorrSame_NormLevel_8s32f_C4R, 641
 - nppiCrossCorrSame_NormLevel_8u32f_AC4R, 641
 - nppiCrossCorrSame_NormLevel_8u32f_C1R, 642
 - nppiCrossCorrSame_NormLevel_8u32f_C3R, 642
 - nppiCrossCorrSame_NormLevel_8u32f_C4R, 643
 - nppiCrossCorrSame_NormLevel_8u_AC4RSfs, 643
 - nppiCrossCorrSame_NormLevel_8u_C1RSfs, 644
 - nppiCrossCorrSame_NormLevel_8u_C3RSfs, 644
 - nppiCrossCorrSame_NormLevel_8u_C4RSfs, 645

- nppiSameNormLevelGetBufferHostSize_16u32f_AC4R, 645
- nppiSameNormLevelGetBufferHostSize_16u32f_C1R, 646
- nppiSameNormLevelGetBufferHostSize_16u32f_C3R, 646
- nppiSameNormLevelGetBufferHostSize_16u32f_C4R, 646
- nppiSameNormLevelGetBufferHostSize_32f_AC4R, 647
- nppiSameNormLevelGetBufferHostSize_32f_C1R, 647
- nppiSameNormLevelGetBufferHostSize_32f_C3R, 647
- nppiSameNormLevelGetBufferHostSize_32f_C4R, 647
- nppiSameNormLevelGetBufferHostSize_8s32f_AC4R, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_C1R, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_C3R, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_C4R, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_AC4R, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_C1R, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_C3R, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_C4R, 650
- nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs, 650
- nppiSameNormLevelGetBufferHostSize_8u_C1RSfs, 650
- nppiSameNormLevelGetBufferHostSize_8u_C3RSfs, 651
- nppiSameNormLevelGetBufferHostSize_8u_C4RSfs, 651
- CrossCorrValid, 609
- crosscorrvalid
 - nppiCrossCorrValid_16u32f_C1R, 609
 - nppiCrossCorrValid_32f_C1R, 610
 - nppiCrossCorrValid_8s32f_C1R, 610
 - nppiCrossCorrValid_8u32f_C1R, 610
- CrossCorrValid_Norm, 598
- CrossCorrValid_NormLevel, 652
- crosscorrvalidnorm
 - nppiCrossCorrValid_Norm_16u32f_AC4R, 600
 - nppiCrossCorrValid_Norm_16u32f_C1R, 600
 - nppiCrossCorrValid_Norm_16u32f_C3R, 600
 - nppiCrossCorrValid_Norm_16u32f_C4R, 601
 - nppiCrossCorrValid_Norm_32f_AC4R, 601
 - nppiCrossCorrValid_Norm_32f_C1R, 602
 - nppiCrossCorrValid_Norm_32f_C3R, 602
 - nppiCrossCorrValid_Norm_32f_C4R, 603
 - nppiCrossCorrValid_Norm_8s32f_AC4R, 603
 - nppiCrossCorrValid_Norm_8s32f_C1R, 603
 - nppiCrossCorrValid_Norm_8s32f_C3R, 604
 - nppiCrossCorrValid_Norm_8s32f_C4R, 604
 - nppiCrossCorrValid_Norm_8u32f_AC4R, 605
 - nppiCrossCorrValid_Norm_8u32f_C1R, 605
 - nppiCrossCorrValid_Norm_8u32f_C3R, 606
 - nppiCrossCorrValid_Norm_8u32f_C4R, 606
 - nppiCrossCorrValid_Norm_8u_AC4RSfs, 606
 - nppiCrossCorrValid_Norm_8u_C1RSfs, 607
 - nppiCrossCorrValid_Norm_8u_C3RSfs, 607
 - nppiCrossCorrValid_Norm_8u_C4RSfs, 608
- crosscorrvalidnormlevel
 - nppiCrossCorrValid_NormLevel_16u32f_AC4R, 656
 - nppiCrossCorrValid_NormLevel_16u32f_C1R, 656
 - nppiCrossCorrValid_NormLevel_16u32f_C3R, 656
 - nppiCrossCorrValid_NormLevel_16u32f_C4R, 657
 - nppiCrossCorrValid_NormLevel_32f_AC4R, 657
 - nppiCrossCorrValid_NormLevel_32f_C1R, 658
 - nppiCrossCorrValid_NormLevel_32f_C3R, 658
 - nppiCrossCorrValid_NormLevel_32f_C4R, 659
 - nppiCrossCorrValid_NormLevel_8s32f_AC4R, 659
 - nppiCrossCorrValid_NormLevel_8s32f_C1R, 660
 - nppiCrossCorrValid_NormLevel_8s32f_C3R, 660
 - nppiCrossCorrValid_NormLevel_8s32f_C4R, 661
 - nppiCrossCorrValid_NormLevel_8u32f_AC4R, 661
 - nppiCrossCorrValid_NormLevel_8u32f_C1R, 662
 - nppiCrossCorrValid_NormLevel_8u32f_C3R, 662
 - nppiCrossCorrValid_NormLevel_8u32f_C4R, 663
 - nppiCrossCorrValid_NormLevel_8u_AC4RSfs, 663
 - nppiCrossCorrValid_NormLevel_8u_C1RSfs, 664

- nppiAverageError_8u_C3R, 726
- nppiAverageError_8u_C4R, 726
- image_average_relative_error
 - nppiAverageRelativeError_16s_C1R, 754
 - nppiAverageRelativeError_16s_C2R, 755
 - nppiAverageRelativeError_16s_C3R, 755
 - nppiAverageRelativeError_16s_C4R, 756
 - nppiAverageRelativeError_16sc_C1R, 756
 - nppiAverageRelativeError_16sc_C2R, 757
 - nppiAverageRelativeError_16sc_C3R, 757
 - nppiAverageRelativeError_16sc_C4R, 757
 - nppiAverageRelativeError_16u_C1R, 758
 - nppiAverageRelativeError_16u_C2R, 758
 - nppiAverageRelativeError_16u_C3R, 759
 - nppiAverageRelativeError_16u_C4R, 759
 - nppiAverageRelativeError_32f_C1R, 760
 - nppiAverageRelativeError_32f_C2R, 760
 - nppiAverageRelativeError_32f_C3R, 761
 - nppiAverageRelativeError_32f_C4R, 761
 - nppiAverageRelativeError_32fc_C1R, 762
 - nppiAverageRelativeError_32fc_C2R, 762
 - nppiAverageRelativeError_32fc_C3R, 762
 - nppiAverageRelativeError_32fc_C4R, 763
 - nppiAverageRelativeError_32s_C1R, 763
 - nppiAverageRelativeError_32s_C2R, 764
 - nppiAverageRelativeError_32s_C3R, 764
 - nppiAverageRelativeError_32s_C4R, 765
 - nppiAverageRelativeError_32sc_C1R, 765
 - nppiAverageRelativeError_32sc_C2R, 766
 - nppiAverageRelativeError_32sc_C3R, 766
 - nppiAverageRelativeError_32sc_C4R, 767
 - nppiAverageRelativeError_32u_C1R, 767
 - nppiAverageRelativeError_32u_C2R, 767
 - nppiAverageRelativeError_32u_C3R, 768
 - nppiAverageRelativeError_32u_C4R, 768
 - nppiAverageRelativeError_64f_C1R, 769
 - nppiAverageRelativeError_64f_C2R, 769
 - nppiAverageRelativeError_64f_C3R, 770
 - nppiAverageRelativeError_64f_C4R, 770
 - nppiAverageRelativeError_8s_C1R, 771
 - nppiAverageRelativeError_8s_C2R, 771
 - nppiAverageRelativeError_8s_C3R, 772
 - nppiAverageRelativeError_8s_C4R, 772
 - nppiAverageRelativeError_8u_C1R, 772
 - nppiAverageRelativeError_8u_C2R, 773
 - nppiAverageRelativeError_8u_C3R, 773
 - nppiAverageRelativeError_8u_C4R, 774
- image_count_in_range
 - nppiCountInRange_32f_AC4R, 484
 - nppiCountInRange_32f_C1R, 484
 - nppiCountInRange_32f_C3R, 485
 - nppiCountInRange_8u_AC4R, 485
 - nppiCountInRange_8u_C1R, 486
 - nppiCountInRange_8u_C3R, 486
- nppiCountInRangeGetBufferHostSize_32f_AC4R, 487
- nppiCountInRangeGetBufferHostSize_32f_C1R, 487
- nppiCountInRangeGetBufferHostSize_32f_C3R, 487
- nppiCountInRangeGetBufferHostSize_8u_AC4R, 487
- nppiCountInRangeGetBufferHostSize_8u_C1R, 488
- nppiCountInRangeGetBufferHostSize_8u_C3R, 488
- image_dot_prod
 - nppiDotProd_16s64f_AC4R, 462
 - nppiDotProd_16s64f_C1R, 462
 - nppiDotProd_16s64f_C3R, 463
 - nppiDotProd_16s64f_C4R, 463
 - nppiDotProd_16u64f_AC4R, 464
 - nppiDotProd_16u64f_C1R, 464
 - nppiDotProd_16u64f_C3R, 465
 - nppiDotProd_16u64f_C4R, 465
 - nppiDotProd_32f64f_AC4R, 465
 - nppiDotProd_32f64f_C1R, 466
 - nppiDotProd_32f64f_C3R, 466
 - nppiDotProd_32f64f_C4R, 467
 - nppiDotProd_32s64f_AC4R, 467
 - nppiDotProd_32s64f_C1R, 468
 - nppiDotProd_32s64f_C3R, 468
 - nppiDotProd_32s64f_C4R, 468
 - nppiDotProd_32u64f_AC4R, 469
 - nppiDotProd_32u64f_C1R, 469
 - nppiDotProd_32u64f_C3R, 470
 - nppiDotProd_32u64f_C4R, 470
 - nppiDotProd_8s64f_AC4R, 471
 - nppiDotProd_8s64f_C1R, 471
 - nppiDotProd_8s64f_C3R, 471
 - nppiDotProd_8s64f_C4R, 472
 - nppiDotProd_8u64f_AC4R, 472
 - nppiDotProd_8u64f_C1R, 473
 - nppiDotProd_8u64f_C3R, 473
 - nppiDotProd_8u64f_C4R, 473
 - nppiDotProdGetBufferHostSize_16s64f_AC4R, 474
 - nppiDotProdGetBufferHostSize_16s64f_C1R, 474
 - nppiDotProdGetBufferHostSize_16s64f_C3R, 474
 - nppiDotProdGetBufferHostSize_16s64f_C4R, 475
 - nppiDotProdGetBufferHostSize_16u64f_AC4R, 475
 - nppiDotProdGetBufferHostSize_16u64f_C1R, 475

- nppiDotProdGetBufferHostSize_16u64f_C3R, 476
- nppiDotProdGetBufferHostSize_16u64f_C4R, 476
- nppiDotProdGetBufferHostSize_32f64f_-AC4R, 476
- nppiDotProdGetBufferHostSize_32f64f_C1R, 476
- nppiDotProdGetBufferHostSize_32f64f_C3R, 477
- nppiDotProdGetBufferHostSize_32f64f_C4R, 477
- nppiDotProdGetBufferHostSize_32s64f_-AC4R, 477
- nppiDotProdGetBufferHostSize_32s64f_C1R, 478
- nppiDotProdGetBufferHostSize_32s64f_C3R, 478
- nppiDotProdGetBufferHostSize_32s64f_C4R, 478
- nppiDotProdGetBufferHostSize_32u64f_-AC4R, 478
- nppiDotProdGetBufferHostSize_32u64f_C1R, 479
- nppiDotProdGetBufferHostSize_32u64f_C3R, 479
- nppiDotProdGetBufferHostSize_32u64f_C4R, 479
- nppiDotProdGetBufferHostSize_8s64f_-AC4R, 480
- nppiDotProdGetBufferHostSize_8s64f_C1R, 480
- nppiDotProdGetBufferHostSize_8s64f_C3R, 480
- nppiDotProdGetBufferHostSize_8s64f_C4R, 480
- nppiDotProdGetBufferHostSize_8u64f_-AC4R, 481
- nppiDotProdGetBufferHostSize_8u64f_C1R, 481
- nppiDotProdGetBufferHostSize_8u64f_C3R, 481
- nppiDotProdGetBufferHostSize_8u64f_C4R, 482
- image_fourier_transforms
 - nppiMagnitude_32fc32f_C1R, 776
 - nppiMagnitudeSqr_32fc32f_C1R, 776
- image_histogrameven
 - nppiEvenLevelsHost_32s, 513
 - nppiHistogramEven_16s_AC4R, 514
 - nppiHistogramEven_16s_C1R, 514
 - nppiHistogramEven_16s_C3R, 514
 - nppiHistogramEven_16s_C4R, 515
 - nppiHistogramEven_16u_AC4R, 515
 - nppiHistogramEven_16u_C1R, 516
 - nppiHistogramEven_16u_C3R, 516
 - nppiHistogramEven_16u_C4R, 517
 - nppiHistogramEven_8u_AC4R, 517
 - nppiHistogramEven_8u_C1R, 518
 - nppiHistogramEven_8u_C3R, 518
 - nppiHistogramEven_8u_C4R, 519
 - nppiHistogramEvenGetBufferSize_16s_-AC4R, 519
 - nppiHistogramEvenGetBufferSize_16s_C1R, 519
 - nppiHistogramEvenGetBufferSize_16s_C3R, 520
 - nppiHistogramEvenGetBufferSize_16s_C4R, 520
 - nppiHistogramEvenGetBufferSize_16u_-AC4R, 520
 - nppiHistogramEvenGetBufferSize_16u_C1R, 521
 - nppiHistogramEvenGetBufferSize_16u_C3R, 521
 - nppiHistogramEvenGetBufferSize_16u_C4R, 521
 - nppiHistogramEvenGetBufferSize_8u_AC4R, 522
 - nppiHistogramEvenGetBufferSize_8u_C1R, 522
 - nppiHistogramEvenGetBufferSize_8u_C3R, 522
 - nppiHistogramEvenGetBufferSize_8u_C4R, 523
- image_histogramrange
 - nppiHistogramRange_16s_AC4R, 527
 - nppiHistogramRange_16s_C1R, 527
 - nppiHistogramRange_16s_C3R, 527
 - nppiHistogramRange_16s_C4R, 528
 - nppiHistogramRange_16u_AC4R, 528
 - nppiHistogramRange_16u_C1R, 529
 - nppiHistogramRange_16u_C3R, 529
 - nppiHistogramRange_16u_C4R, 529
 - nppiHistogramRange_32f_AC4R, 530
 - nppiHistogramRange_32f_C1R, 530
 - nppiHistogramRange_32f_C3R, 531
 - nppiHistogramRange_32f_C4R, 531
 - nppiHistogramRange_8u_AC4R, 532
 - nppiHistogramRange_8u_C1R, 532
 - nppiHistogramRange_8u_C3R, 533
 - nppiHistogramRange_8u_C4R, 533
 - nppiHistogramRangeGetBufferSize_16s_-AC4R, 533
 - nppiHistogramRangeGetBufferSize_16s_-C1R, 534
 - nppiHistogramRangeGetBufferSize_16s_-C3R, 534

- nppiHistogramRangeGetBufferSize_16s_-C4R, 534
- nppiHistogramRangeGetBufferSize_16u_-AC4R, 535
- nppiHistogramRangeGetBufferSize_16u_-C1R, 535
- nppiHistogramRangeGetBufferSize_16u_-C3R, 535
- nppiHistogramRangeGetBufferSize_16u_-C4R, 536
- nppiHistogramRangeGetBufferSize_32f_-AC4R, 536
- nppiHistogramRangeGetBufferSize_32f_C1R, 536
- nppiHistogramRangeGetBufferSize_32f_C3R, 537
- nppiHistogramRangeGetBufferSize_32f_C4R, 537
- nppiHistogramRangeGetBufferSize_8u_-AC4R, 537
- nppiHistogramRangeGetBufferSize_8u_C1R, 538
- nppiHistogramRangeGetBufferSize_8u_C3R, 538
- nppiHistogramRangeGetBufferSize_8u_C4R, 538
- image_inf_norm
 - nppiNorm_Inf_16s_AC4R, 260
 - nppiNorm_Inf_16s_C1R, 260
 - nppiNorm_Inf_16s_C3R, 260
 - nppiNorm_Inf_16s_C4R, 261
 - nppiNorm_Inf_16u_AC4R, 261
 - nppiNorm_Inf_16u_C1MR, 261
 - nppiNorm_Inf_16u_C1R, 262
 - nppiNorm_Inf_16u_C3CMR, 262
 - nppiNorm_Inf_16u_C3R, 263
 - nppiNorm_Inf_16u_C4R, 263
 - nppiNorm_Inf_32f_AC4R, 263
 - nppiNorm_Inf_32f_C1MR, 264
 - nppiNorm_Inf_32f_C1R, 264
 - nppiNorm_Inf_32f_C3CMR, 265
 - nppiNorm_Inf_32f_C3R, 265
 - nppiNorm_Inf_32f_C4R, 265
 - nppiNorm_Inf_32s_C1R, 266
 - nppiNorm_Inf_8s_C1MR, 266
 - nppiNorm_Inf_8s_C3CMR, 267
 - nppiNorm_Inf_8u_AC4R, 267
 - nppiNorm_Inf_8u_C1MR, 267
 - nppiNorm_Inf_8u_C1R, 268
 - nppiNorm_Inf_8u_C3CMR, 268
 - nppiNorm_Inf_8u_C3R, 269
 - nppiNorm_Inf_8u_C4R, 269
 - nppiNormInfGetBufferHostSize_16s_AC4R, 269
 - nppiNormInfGetBufferHostSize_16s_C1R, 270
 - nppiNormInfGetBufferHostSize_16s_C3R, 270
 - nppiNormInfGetBufferHostSize_16s_C4R, 270
 - nppiNormInfGetBufferHostSize_16u_AC4R, 271
 - nppiNormInfGetBufferHostSize_16u_C1MR, 271
 - nppiNormInfGetBufferHostSize_16u_C1R, 271
 - nppiNormInfGetBufferHostSize_16u_-C3CMR, 271
 - nppiNormInfGetBufferHostSize_16u_C3R, 272
 - nppiNormInfGetBufferHostSize_16u_C4R, 272
 - nppiNormInfGetBufferHostSize_32f_AC4R, 272
 - nppiNormInfGetBufferHostSize_32f_C1MR, 273
 - nppiNormInfGetBufferHostSize_32f_C1R, 273
 - nppiNormInfGetBufferHostSize_32f_-C3CMR, 273
 - nppiNormInfGetBufferHostSize_32f_C3R, 273
 - nppiNormInfGetBufferHostSize_32f_C4R, 274
 - nppiNormInfGetBufferHostSize_32s_C1R, 274
 - nppiNormInfGetBufferHostSize_8s_C1MR, 274
 - nppiNormInfGetBufferHostSize_8s_C3CMR, 275
 - nppiNormInfGetBufferHostSize_8u_AC4R, 275
 - nppiNormInfGetBufferHostSize_8u_C1MR, 275
 - nppiNormInfGetBufferHostSize_8u_C1R, 275
 - nppiNormInfGetBufferHostSize_8u_C3CMR, 276
 - nppiNormInfGetBufferHostSize_8u_C3R, 276
 - nppiNormInfGetBufferHostSize_8u_C4R, 276
- image_inf_normdiff
 - nppiNormDiff_Inf_16s_AC4R, 324
 - nppiNormDiff_Inf_16s_C1R, 324
 - nppiNormDiff_Inf_16s_C3R, 325
 - nppiNormDiff_Inf_16s_C4R, 325
 - nppiNormDiff_Inf_16u_AC4R, 326
 - nppiNormDiff_Inf_16u_C1MR, 326
 - nppiNormDiff_Inf_16u_C1R, 327
 - nppiNormDiff_Inf_16u_C3CMR, 327

- nppiNormDiff_Inf_16u_C3R, 328
- nppiNormDiff_Inf_16u_C4R, 328
- nppiNormDiff_Inf_32f_AC4R, 328
- nppiNormDiff_Inf_32f_C1MR, 329
- nppiNormDiff_Inf_32f_C1R, 329
- nppiNormDiff_Inf_32f_C3CMR, 330
- nppiNormDiff_Inf_32f_C3R, 330
- nppiNormDiff_Inf_32f_C4R, 331
- nppiNormDiff_Inf_8s_C1MR, 331
- nppiNormDiff_Inf_8s_C3CMR, 332
- nppiNormDiff_Inf_8u_AC4R, 332
- nppiNormDiff_Inf_8u_C1MR, 333
- nppiNormDiff_Inf_8u_C1R, 333
- nppiNormDiff_Inf_8u_C3CMR, 334
- nppiNormDiff_Inf_8u_C3R, 334
- nppiNormDiff_Inf_8u_C4R, 335
- nppiNormDiffInfGetBufferHostSize_16s_-AC4R, 335
- nppiNormDiffInfGetBufferHostSize_16s_-C1R, 335
- nppiNormDiffInfGetBufferHostSize_16s_-C3R, 336
- nppiNormDiffInfGetBufferHostSize_16s_-C4R, 336
- nppiNormDiffInfGetBufferHostSize_16u_-AC4R, 336
- nppiNormDiffInfGetBufferHostSize_16u_-C1MR, 337
- nppiNormDiffInfGetBufferHostSize_16u_-C1R, 337
- nppiNormDiffInfGetBufferHostSize_16u_-C3CMR, 337
- nppiNormDiffInfGetBufferHostSize_16u_-C3R, 337
- nppiNormDiffInfGetBufferHostSize_16u_-C4R, 338
- nppiNormDiffInfGetBufferHostSize_32f_-AC4R, 338
- nppiNormDiffInfGetBufferHostSize_32f_-C1MR, 338
- nppiNormDiffInfGetBufferHostSize_32f_-C1R, 339
- nppiNormDiffInfGetBufferHostSize_32f_-C3CMR, 339
- nppiNormDiffInfGetBufferHostSize_32f_-C3R, 339
- nppiNormDiffInfGetBufferHostSize_32f_-C4R, 339
- nppiNormDiffInfGetBufferHostSize_8s_-C1MR, 340
- nppiNormDiffInfGetBufferHostSize_8s_-C3CMR, 340
- nppiNormDiffInfGetBufferHostSize_8u_-AC4R, 340
- nppiNormDiffInfGetBufferHostSize_8u_-C1MR, 341
- nppiNormDiffInfGetBufferHostSize_8u_C1R, 341
- nppiNormDiffInfGetBufferHostSize_8u_-C3CMR, 341
- nppiNormDiffInfGetBufferHostSize_8u_C3R, 341
- nppiNormDiffInfGetBufferHostSize_8u_C4R, 342
- image_inf_normrel
 - nppiNormRel_Inf_16s_AC4R, 393
 - nppiNormRel_Inf_16s_C1R, 393
 - nppiNormRel_Inf_16s_C3R, 394
 - nppiNormRel_Inf_16s_C4R, 394
 - nppiNormRel_Inf_16u_AC4R, 395
 - nppiNormRel_Inf_16u_C1MR, 395
 - nppiNormRel_Inf_16u_C1R, 396
 - nppiNormRel_Inf_16u_C3CMR, 396
 - nppiNormRel_Inf_16u_C3R, 397
 - nppiNormRel_Inf_16u_C4R, 397
 - nppiNormRel_Inf_32f_AC4R, 397
 - nppiNormRel_Inf_32f_C1MR, 398
 - nppiNormRel_Inf_32f_C1R, 398
 - nppiNormRel_Inf_32f_C3CMR, 399
 - nppiNormRel_Inf_32f_C3R, 399
 - nppiNormRel_Inf_32f_C4R, 400
 - nppiNormRel_Inf_8s_C1MR, 400
 - nppiNormRel_Inf_8s_C3CMR, 401
 - nppiNormRel_Inf_8u_AC4R, 401
 - nppiNormRel_Inf_8u_C1MR, 402
 - nppiNormRel_Inf_8u_C1R, 402
 - nppiNormRel_Inf_8u_C3CMR, 403
 - nppiNormRel_Inf_8u_C3R, 403
 - nppiNormRel_Inf_8u_C4R, 404
 - nppiNormRelInfGetBufferHostSize_16s_-AC4R, 404
 - nppiNormRelInfGetBufferHostSize_16s_-C1R, 405
 - nppiNormRelInfGetBufferHostSize_16s_-C3R, 405
 - nppiNormRelInfGetBufferHostSize_16s_-C4R, 405
 - nppiNormRelInfGetBufferHostSize_16u_-AC4R, 405
 - nppiNormRelInfGetBufferHostSize_16u_-C1MR, 406
 - nppiNormRelInfGetBufferHostSize_16u_-C1R, 406
 - nppiNormRelInfGetBufferHostSize_16u_-C3CMR, 406
 - nppiNormRelInfGetBufferHostSize_16u_-C3R, 407

- nppiNormRelInfGetBufferHostSize_16u_-C4R, 407
- nppiNormRelInfGetBufferHostSize_32f_-AC4R, 407
- nppiNormRelInfGetBufferHostSize_32f_-C1MR, 407
- nppiNormRelInfGetBufferHostSize_32f_C1R, 408
- nppiNormRelInfGetBufferHostSize_32f_-C3CMR, 408
- nppiNormRelInfGetBufferHostSize_32f_C3R, 408
- nppiNormRelInfGetBufferHostSize_32f_C4R, 409
- nppiNormRelInfGetBufferHostSize_32s_-C1R, 409
- nppiNormRelInfGetBufferHostSize_8s_-C1MR, 409
- nppiNormRelInfGetBufferHostSize_8s_-C3CMR, 409
- nppiNormRelInfGetBufferHostSize_8u_-AC4R, 410
- nppiNormRelInfGetBufferHostSize_8u_-C1MR, 410
- nppiNormRelInfGetBufferHostSize_8u_C1R, 410
- nppiNormRelInfGetBufferHostSize_8u_-C3CMR, 411
- nppiNormRelInfGetBufferHostSize_8u_C3R, 411
- nppiNormRelInfGetBufferHostSize_8u_C4R, 411
- image_integral
 - nppiIntegral_8u32f_C1R, 503
 - nppiIntegral_8u32s_C1R, 503
- image_L1_norm
 - nppiNorm_L1_16s_AC4R, 282
 - nppiNorm_L1_16s_C1R, 282
 - nppiNorm_L1_16s_C3R, 282
 - nppiNorm_L1_16s_C4R, 283
 - nppiNorm_L1_16u_AC4R, 283
 - nppiNorm_L1_16u_C1MR, 283
 - nppiNorm_L1_16u_C1R, 284
 - nppiNorm_L1_16u_C3CMR, 284
 - nppiNorm_L1_16u_C3R, 285
 - nppiNorm_L1_16u_C4R, 285
 - nppiNorm_L1_32f_AC4R, 285
 - nppiNorm_L1_32f_C1MR, 286
 - nppiNorm_L1_32f_C1R, 286
 - nppiNorm_L1_32f_C3CMR, 286
 - nppiNorm_L1_32f_C3R, 287
 - nppiNorm_L1_32f_C4R, 287
 - nppiNorm_L1_8s_C1MR, 288
 - nppiNorm_L1_8s_C3CMR, 288
 - nppiNorm_L1_8u_AC4R, 288
 - nppiNorm_L1_8u_C1MR, 289
 - nppiNorm_L1_8u_C1R, 289
 - nppiNorm_L1_8u_C3CMR, 290
 - nppiNorm_L1_8u_C3R, 290
 - nppiNorm_L1_8u_C4R, 290
 - nppiNormL1GetBufferHostSize_16s_AC4R, 291
 - nppiNormL1GetBufferHostSize_16s_C1R, 291
 - nppiNormL1GetBufferHostSize_16s_C3R, 291
 - nppiNormL1GetBufferHostSize_16s_C4R, 292
 - nppiNormL1GetBufferHostSize_16u_AC4R, 292
 - nppiNormL1GetBufferHostSize_16u_C1MR, 292
 - nppiNormL1GetBufferHostSize_16u_C1R, 293
 - nppiNormL1GetBufferHostSize_16u_-C3CMR, 293
 - nppiNormL1GetBufferHostSize_16u_C3R, 293
 - nppiNormL1GetBufferHostSize_16u_C4R, 293
 - nppiNormL1GetBufferHostSize_32f_AC4R, 294
 - nppiNormL1GetBufferHostSize_32f_C1MR, 294
 - nppiNormL1GetBufferHostSize_32f_C1R, 294
 - nppiNormL1GetBufferHostSize_32f_-C3CMR, 295
 - nppiNormL1GetBufferHostSize_32f_C3R, 295
 - nppiNormL1GetBufferHostSize_32f_C4R, 295
 - nppiNormL1GetBufferHostSize_8s_C1MR, 295
 - nppiNormL1GetBufferHostSize_8s_C3CMR, 296
 - nppiNormL1GetBufferHostSize_8u_AC4R, 296
 - nppiNormL1GetBufferHostSize_8u_C1MR, 296
 - nppiNormL1GetBufferHostSize_8u_C1R, 297
 - nppiNormL1GetBufferHostSize_8u_C3CMR, 297
 - nppiNormL1GetBufferHostSize_8u_C3R, 297
 - nppiNormL1GetBufferHostSize_8u_C4R, 297
- image_L1_normdiff
 - nppiNormDiff_L1_16s_AC4R, 347
 - nppiNormDiff_L1_16s_C1R, 347

- nppiNormDiff_L1_16s_C3R, 348
- nppiNormDiff_L1_16s_C4R, 348
- nppiNormDiff_L1_16u_AC4R, 349
- nppiNormDiff_L1_16u_C1MR, 349
- nppiNormDiff_L1_16u_C1R, 349
- nppiNormDiff_L1_16u_C3CMR, 350
- nppiNormDiff_L1_16u_C3R, 350
- nppiNormDiff_L1_16u_C4R, 351
- nppiNormDiff_L1_32f_AC4R, 351
- nppiNormDiff_L1_32f_C1MR, 352
- nppiNormDiff_L1_32f_C1R, 352
- nppiNormDiff_L1_32f_C3CMR, 353
- nppiNormDiff_L1_32f_C3R, 353
- nppiNormDiff_L1_32f_C4R, 354
- nppiNormDiff_L1_8s_C1MR, 354
- nppiNormDiff_L1_8s_C3CMR, 355
- nppiNormDiff_L1_8u_AC4R, 355
- nppiNormDiff_L1_8u_C1MR, 356
- nppiNormDiff_L1_8u_C1R, 356
- nppiNormDiff_L1_8u_C3CMR, 356
- nppiNormDiff_L1_8u_C3R, 357
- nppiNormDiff_L1_8u_C4R, 357
- nppiNormDiffL1GetBufferHostSize_16s_-AC4R, 358
- nppiNormDiffL1GetBufferHostSize_16s_-C1R, 358
- nppiNormDiffL1GetBufferHostSize_16s_-C3R, 358
- nppiNormDiffL1GetBufferHostSize_16s_-C4R, 359
- nppiNormDiffL1GetBufferHostSize_16u_-AC4R, 359
- nppiNormDiffL1GetBufferHostSize_16u_-C1MR, 359
- nppiNormDiffL1GetBufferHostSize_16u_-C1R, 360
- nppiNormDiffL1GetBufferHostSize_16u_-C3CMR, 360
- nppiNormDiffL1GetBufferHostSize_16u_-C3R, 360
- nppiNormDiffL1GetBufferHostSize_16u_-C4R, 360
- nppiNormDiffL1GetBufferHostSize_32f_-AC4R, 361
- nppiNormDiffL1GetBufferHostSize_32f_-C1MR, 361
- nppiNormDiffL1GetBufferHostSize_32f_-C1R, 361
- nppiNormDiffL1GetBufferHostSize_32f_-C3CMR, 362
- nppiNormDiffL1GetBufferHostSize_32f_-C3R, 362
- nppiNormDiffL1GetBufferHostSize_32f_-C4R, 362
- nppiNormDiffL1GetBufferHostSize_8s_-C1MR, 362
- nppiNormDiffL1GetBufferHostSize_8s_-C3CMR, 363
- nppiNormDiffL1GetBufferHostSize_8u_-AC4R, 363
- nppiNormDiffL1GetBufferHostSize_8u_-C1MR, 363
- nppiNormDiffL1GetBufferHostSize_8u_C1R, 364
- nppiNormDiffL1GetBufferHostSize_8u_-C3CMR, 364
- nppiNormDiffL1GetBufferHostSize_8u_C3R, 364
- nppiNormDiffL1GetBufferHostSize_8u_C4R, 364
- image_L1_normrel
 - nppiNormRel_L1_16s_AC4R, 416
 - nppiNormRel_L1_16s_C1R, 416
 - nppiNormRel_L1_16s_C3R, 417
 - nppiNormRel_L1_16s_C4R, 417
 - nppiNormRel_L1_16u_AC4R, 418
 - nppiNormRel_L1_16u_C1MR, 418
 - nppiNormRel_L1_16u_C1R, 419
 - nppiNormRel_L1_16u_C3CMR, 419
 - nppiNormRel_L1_16u_C3R, 419
 - nppiNormRel_L1_16u_C4R, 420
 - nppiNormRel_L1_32f_AC4R, 420
 - nppiNormRel_L1_32f_C1MR, 421
 - nppiNormRel_L1_32f_C1R, 421
 - nppiNormRel_L1_32f_C3CMR, 422
 - nppiNormRel_L1_32f_C3R, 422
 - nppiNormRel_L1_32f_C4R, 423
 - nppiNormRel_L1_8s_C1MR, 423
 - nppiNormRel_L1_8s_C3CMR, 424
 - nppiNormRel_L1_8u_AC4R, 424
 - nppiNormRel_L1_8u_C1MR, 425
 - nppiNormRel_L1_8u_C1R, 425
 - nppiNormRel_L1_8u_C3CMR, 426
 - nppiNormRel_L1_8u_C3R, 426
 - nppiNormRel_L1_8u_C4R, 427
 - nppiNormRelL1GetBufferHostSize_16s_-AC4R, 427
 - nppiNormRelL1GetBufferHostSize_16s_C1R, 427
 - nppiNormRelL1GetBufferHostSize_16s_C3R, 428
 - nppiNormRelL1GetBufferHostSize_16s_C4R, 428
 - nppiNormRelL1GetBufferHostSize_16u_-AC4R, 428
 - nppiNormRelL1GetBufferHostSize_16u_-C1MR, 429

- nppiNormRelL1GetBufferHostSize_16u_-C1R, 429
- nppiNormRelL1GetBufferHostSize_16u_-C3CMR, 429
- nppiNormRelL1GetBufferHostSize_16u_-C3R, 429
- nppiNormRelL1GetBufferHostSize_16u_-C4R, 430
- nppiNormRelL1GetBufferHostSize_32f_-AC4R, 430
- nppiNormRelL1GetBufferHostSize_32f_-C1MR, 430
- nppiNormRelL1GetBufferHostSize_32f_C1R, 431
- nppiNormRelL1GetBufferHostSize_32f_-C3CMR, 431
- nppiNormRelL1GetBufferHostSize_32f_C3R, 431
- nppiNormRelL1GetBufferHostSize_32f_C4R, 431
- nppiNormRelL1GetBufferHostSize_8s_-C1MR, 432
- nppiNormRelL1GetBufferHostSize_8s_-C3CMR, 432
- nppiNormRelL1GetBufferHostSize_8u_-AC4R, 432
- nppiNormRelL1GetBufferHostSize_8u_-C1MR, 433
- nppiNormRelL1GetBufferHostSize_8u_C1R, 433
- nppiNormRelL1GetBufferHostSize_8u_-C3CMR, 433
- nppiNormRelL1GetBufferHostSize_8u_C3R, 433
- nppiNormRelL1GetBufferHostSize_8u_C4R, 434
- image_L2_norm
 - nppiNorm_L2_16s_AC4R, 303
 - nppiNorm_L2_16s_C1R, 303
 - nppiNorm_L2_16s_C3R, 303
 - nppiNorm_L2_16s_C4R, 304
 - nppiNorm_L2_16u_AC4R, 304
 - nppiNorm_L2_16u_C1MR, 304
 - nppiNorm_L2_16u_C1R, 305
 - nppiNorm_L2_16u_C3CMR, 305
 - nppiNorm_L2_16u_C3R, 306
 - nppiNorm_L2_16u_C4R, 306
 - nppiNorm_L2_32f_AC4R, 306
 - nppiNorm_L2_32f_C1MR, 307
 - nppiNorm_L2_32f_C1R, 307
 - nppiNorm_L2_32f_C3CMR, 307
 - nppiNorm_L2_32f_C3R, 308
 - nppiNorm_L2_32f_C4R, 308
 - nppiNorm_L2_8s_C1MR, 309
 - nppiNorm_L2_8s_C3CMR, 309
 - nppiNorm_L2_8u_AC4R, 309
 - nppiNorm_L2_8u_C1MR, 310
 - nppiNorm_L2_8u_C1R, 310
 - nppiNorm_L2_8u_C3CMR, 311
 - nppiNorm_L2_8u_C3R, 311
 - nppiNorm_L2_8u_C4R, 311
 - nppiNormL2GetBufferHostSize_16s_AC4R, 312
 - nppiNormL2GetBufferHostSize_16s_C1R, 312
 - nppiNormL2GetBufferHostSize_16s_C3R, 312
 - nppiNormL2GetBufferHostSize_16s_C4R, 313
 - nppiNormL2GetBufferHostSize_16u_AC4R, 313
 - nppiNormL2GetBufferHostSize_16u_C1MR, 313
 - nppiNormL2GetBufferHostSize_16u_C1R, 314
 - nppiNormL2GetBufferHostSize_16u_-C3CMR, 314
 - nppiNormL2GetBufferHostSize_16u_C3R, 314
 - nppiNormL2GetBufferHostSize_16u_C4R, 314
 - nppiNormL2GetBufferHostSize_32f_AC4R, 315
 - nppiNormL2GetBufferHostSize_32f_C1MR, 315
 - nppiNormL2GetBufferHostSize_32f_C1R, 315
 - nppiNormL2GetBufferHostSize_32f_-C3CMR, 316
 - nppiNormL2GetBufferHostSize_32f_C3R, 316
 - nppiNormL2GetBufferHostSize_32f_C4R, 316
 - nppiNormL2GetBufferHostSize_8s_C1MR, 316
 - nppiNormL2GetBufferHostSize_8s_C3CMR, 317
 - nppiNormL2GetBufferHostSize_8u_AC4R, 317
 - nppiNormL2GetBufferHostSize_8u_C1MR, 317
 - nppiNormL2GetBufferHostSize_8u_C1R, 318
 - nppiNormL2GetBufferHostSize_8u_C3CMR, 318
 - nppiNormL2GetBufferHostSize_8u_C3R, 318
 - nppiNormL2GetBufferHostSize_8u_C4R, 318
- image_L2_normdiff
 - nppiNormDiff_L2_16s_AC4R, 370

- nppiNormDiff_L2_16s_C1R, 370
- nppiNormDiff_L2_16s_C3R, 371
- nppiNormDiff_L2_16s_C4R, 371
- nppiNormDiff_L2_16u_AC4R, 372
- nppiNormDiff_L2_16u_C1MR, 372
- nppiNormDiff_L2_16u_C1R, 372
- nppiNormDiff_L2_16u_C3CMR, 373
- nppiNormDiff_L2_16u_C3R, 373
- nppiNormDiff_L2_16u_C4R, 374
- nppiNormDiff_L2_32f_AC4R, 374
- nppiNormDiff_L2_32f_C1MR, 375
- nppiNormDiff_L2_32f_C1R, 375
- nppiNormDiff_L2_32f_C3CMR, 376
- nppiNormDiff_L2_32f_C3R, 376
- nppiNormDiff_L2_32f_C4R, 377
- nppiNormDiff_L2_8s_C1MR, 377
- nppiNormDiff_L2_8s_C3CMR, 378
- nppiNormDiff_L2_8u_AC4R, 378
- nppiNormDiff_L2_8u_C1MR, 379
- nppiNormDiff_L2_8u_C1R, 379
- nppiNormDiff_L2_8u_C3CMR, 379
- nppiNormDiff_L2_8u_C3R, 380
- nppiNormDiff_L2_8u_C4R, 380
- nppiNormDiffL2GetBufferHostSize_16s_-AC4R, 381
- nppiNormDiffL2GetBufferHostSize_16s_-C1R, 381
- nppiNormDiffL2GetBufferHostSize_16s_-C3R, 381
- nppiNormDiffL2GetBufferHostSize_16s_-C4R, 382
- nppiNormDiffL2GetBufferHostSize_16u_-AC4R, 382
- nppiNormDiffL2GetBufferHostSize_16u_-C1MR, 382
- nppiNormDiffL2GetBufferHostSize_16u_-C1R, 383
- nppiNormDiffL2GetBufferHostSize_16u_-C3CMR, 383
- nppiNormDiffL2GetBufferHostSize_16u_-C3R, 383
- nppiNormDiffL2GetBufferHostSize_16u_-C4R, 383
- nppiNormDiffL2GetBufferHostSize_32f_-AC4R, 384
- nppiNormDiffL2GetBufferHostSize_32f_-C1MR, 384
- nppiNormDiffL2GetBufferHostSize_32f_-C1R, 384
- nppiNormDiffL2GetBufferHostSize_32f_-C3CMR, 385
- nppiNormDiffL2GetBufferHostSize_32f_-C3R, 385
- nppiNormDiffL2GetBufferHostSize_32f_-C4R, 385
- nppiNormDiffL2GetBufferHostSize_8s_-C1MR, 385
- nppiNormDiffL2GetBufferHostSize_8s_-C1MR, 385
- nppiNormDiffL2GetBufferHostSize_8s_-C3CMR, 386
- nppiNormDiffL2GetBufferHostSize_8u_-AC4R, 386
- nppiNormDiffL2GetBufferHostSize_8u_-C1MR, 386
- nppiNormDiffL2GetBufferHostSize_8u_C1R, 387
- nppiNormDiffL2GetBufferHostSize_8u_-C3CMR, 387
- nppiNormDiffL2GetBufferHostSize_8u_C3R, 387
- nppiNormDiffL2GetBufferHostSize_8u_C4R, 387
- image_L2_normrel
 - nppiNormRel_L2_16s_AC4R, 439
 - nppiNormRel_L2_16s_C1R, 439
 - nppiNormRel_L2_16s_C3R, 440
 - nppiNormRel_L2_16s_C4R, 440
 - nppiNormRel_L2_16u_AC4R, 441
 - nppiNormRel_L2_16u_C1MR, 441
 - nppiNormRel_L2_16u_C1R, 442
 - nppiNormRel_L2_16u_C3CMR, 442
 - nppiNormRel_L2_16u_C3R, 442
 - nppiNormRel_L2_16u_C4R, 443
 - nppiNormRel_L2_32f_AC4R, 443
 - nppiNormRel_L2_32f_C1MR, 444
 - nppiNormRel_L2_32f_C1R, 444
 - nppiNormRel_L2_32f_C3CMR, 445
 - nppiNormRel_L2_32f_C3R, 445
 - nppiNormRel_L2_32f_C4R, 446
 - nppiNormRel_L2_8s_C1MR, 446
 - nppiNormRel_L2_8s_C3CMR, 447
 - nppiNormRel_L2_8u_AC4R, 447
 - nppiNormRel_L2_8u_C1MR, 448
 - nppiNormRel_L2_8u_C1R, 448
 - nppiNormRel_L2_8u_C3CMR, 449
 - nppiNormRel_L2_8u_C3R, 449
 - nppiNormRel_L2_8u_C4R, 450
 - nppiNormRelL2GetBufferHostSize_16s_-AC4R, 450
 - nppiNormRelL2GetBufferHostSize_16s_C1R, 450
 - nppiNormRelL2GetBufferHostSize_16s_C3R, 451
 - nppiNormRelL2GetBufferHostSize_16s_C4R, 451
 - nppiNormRelL2GetBufferHostSize_16u_-AC4R, 451

- nppiNormReLL2GetBufferHostSize_16u_-C1MR, 452
- nppiNormReLL2GetBufferHostSize_16u_-C1R, 452
- nppiNormReLL2GetBufferHostSize_16u_-C3CMR, 452
- nppiNormReLL2GetBufferHostSize_16u_-C3R, 452
- nppiNormReLL2GetBufferHostSize_16u_-C4R, 453
- nppiNormReLL2GetBufferHostSize_32f_-AC4R, 453
- nppiNormReLL2GetBufferHostSize_32f_-C1MR, 453
- nppiNormReLL2GetBufferHostSize_32f_-C1R, 454
- nppiNormReLL2GetBufferHostSize_32f_-C3CMR, 454
- nppiNormReLL2GetBufferHostSize_32f_-C3R, 454
- nppiNormReLL2GetBufferHostSize_32f_-C4R, 454
- nppiNormReLL2GetBufferHostSize_8s_-C1MR, 455
- nppiNormReLL2GetBufferHostSize_8s_-C3CMR, 455
- nppiNormReLL2GetBufferHostSize_8u_-AC4R, 455
- nppiNormReLL2GetBufferHostSize_8u_-C1MR, 456
- nppiNormReLL2GetBufferHostSize_8u_-C1R, 456
- nppiNormReLL2GetBufferHostSize_8u_-C3CMR, 456
- nppiNormReLL2GetBufferHostSize_8u_-C3R, 456
- nppiNormReLL2GetBufferHostSize_8u_-C4R, 457
- image_max
 - nppiMax_16s_AC4R, 161
 - nppiMax_16s_C1R, 161
 - nppiMax_16s_C3R, 162
 - nppiMax_16s_C4R, 162
 - nppiMax_16u_AC4R, 162
 - nppiMax_16u_C1R, 163
 - nppiMax_16u_C3R, 163
 - nppiMax_16u_C4R, 164
 - nppiMax_32f_AC4R, 164
 - nppiMax_32f_C1R, 164
 - nppiMax_32f_C3R, 165
 - nppiMax_32f_C4R, 165
 - nppiMax_8u_AC4R, 165
 - nppiMax_8u_C1R, 166
 - nppiMax_8u_C3R, 166
 - nppiMax_8u_C4R, 167
 - nppiMaxGetBufferHostSize_16s_AC4R, 167
 - nppiMaxGetBufferHostSize_16s_C1R, 167
 - nppiMaxGetBufferHostSize_16s_C3R, 167
 - nppiMaxGetBufferHostSize_16s_C4R, 168
 - nppiMaxGetBufferHostSize_16u_AC4R, 168
 - nppiMaxGetBufferHostSize_16u_C1R, 168
 - nppiMaxGetBufferHostSize_16u_C3R, 169
 - nppiMaxGetBufferHostSize_16u_C4R, 169
 - nppiMaxGetBufferHostSize_32f_AC4R, 169
 - nppiMaxGetBufferHostSize_32f_C1R, 169
 - nppiMaxGetBufferHostSize_32f_C3R, 170
 - nppiMaxGetBufferHostSize_32f_C4R, 170
 - nppiMaxGetBufferHostSize_8u_AC4R, 170
 - nppiMaxGetBufferHostSize_8u_C1R, 170
 - nppiMaxGetBufferHostSize_8u_C3R, 171
 - nppiMaxGetBufferHostSize_8u_C4R, 171
- image_max_index
 - nppiMaxIndx_16s_AC4R, 174
 - nppiMaxIndx_16s_C1R, 175
 - nppiMaxIndx_16s_C3R, 175
 - nppiMaxIndx_16s_C4R, 175
 - nppiMaxIndx_16u_AC4R, 176
 - nppiMaxIndx_16u_C1R, 176
 - nppiMaxIndx_16u_C3R, 177
 - nppiMaxIndx_16u_C4R, 177
 - nppiMaxIndx_32f_AC4R, 177
 - nppiMaxIndx_32f_C1R, 178
 - nppiMaxIndx_32f_C3R, 178
 - nppiMaxIndx_32f_C4R, 179
 - nppiMaxIndx_8u_AC4R, 179
 - nppiMaxIndx_8u_C1R, 179
 - nppiMaxIndx_8u_C3R, 180
 - nppiMaxIndx_8u_C4R, 180
 - nppiMaxIndxGetBufferHostSize_16s_AC4R, 181
 - nppiMaxIndxGetBufferHostSize_16s_C1R, 181
 - nppiMaxIndxGetBufferHostSize_16s_C3R, 181
 - nppiMaxIndxGetBufferHostSize_16s_C4R, 182
 - nppiMaxIndxGetBufferHostSize_16u_AC4R, 182
 - nppiMaxIndxGetBufferHostSize_16u_C1R, 182
 - nppiMaxIndxGetBufferHostSize_16u_C3R, 182
 - nppiMaxIndxGetBufferHostSize_16u_C4R, 183
 - nppiMaxIndxGetBufferHostSize_32f_AC4R, 183
 - nppiMaxIndxGetBufferHostSize_32f_C1R, 183

- nppiMaxIdxGetBufferHostSize_32f_C3R, 184
- nppiMaxIdxGetBufferHostSize_32f_C4R, 184
- nppiMaxIdxGetBufferHostSize_8u_AC4R, 184
- nppiMaxIdxGetBufferHostSize_8u_C1R, 184
- nppiMaxIdxGetBufferHostSize_8u_C3R, 185
- nppiMaxIdxGetBufferHostSize_8u_C4R, 185
- image_maxevery
 - nppiMaxEvery_16s_AC4IR, 490
 - nppiMaxEvery_16s_C1IR, 490
 - nppiMaxEvery_16s_C3IR, 491
 - nppiMaxEvery_16s_C4IR, 491
 - nppiMaxEvery_16u_AC4IR, 491
 - nppiMaxEvery_16u_C1IR, 492
 - nppiMaxEvery_16u_C3IR, 492
 - nppiMaxEvery_16u_C4IR, 492
 - nppiMaxEvery_32f_AC4IR, 493
 - nppiMaxEvery_32f_C1IR, 493
 - nppiMaxEvery_32f_C3IR, 493
 - nppiMaxEvery_32f_C4IR, 494
 - nppiMaxEvery_8u_AC4IR, 494
 - nppiMaxEvery_8u_C1IR, 494
 - nppiMaxEvery_8u_C3IR, 495
 - nppiMaxEvery_8u_C4IR, 495
- image_maximum_error
 - nppiMaximumError_16s_C1R, 684
 - nppiMaximumError_16s_C2R, 685
 - nppiMaximumError_16s_C3R, 685
 - nppiMaximumError_16s_C4R, 685
 - nppiMaximumError_16sc_C1R, 686
 - nppiMaximumError_16sc_C2R, 686
 - nppiMaximumError_16sc_C3R, 687
 - nppiMaximumError_16sc_C4R, 687
 - nppiMaximumError_16u_C1R, 688
 - nppiMaximumError_16u_C2R, 688
 - nppiMaximumError_16u_C3R, 688
 - nppiMaximumError_16u_C4R, 689
 - nppiMaximumError_32f_C1R, 689
 - nppiMaximumError_32f_C2R, 690
 - nppiMaximumError_32f_C3R, 690
 - nppiMaximumError_32f_C4R, 691
 - nppiMaximumError_32fc_C1R, 691
 - nppiMaximumError_32fc_C2R, 692
 - nppiMaximumError_32fc_C3R, 692
 - nppiMaximumError_32fc_C4R, 692
 - nppiMaximumError_32s_C1R, 693
 - nppiMaximumError_32s_C2R, 693
 - nppiMaximumError_32s_C3R, 694
 - nppiMaximumError_32s_C4R, 694
 - nppiMaximumError_32sc_C1R, 695
 - nppiMaximumError_32sc_C2R, 695
 - nppiMaximumError_32sc_C3R, 695
 - nppiMaximumError_32sc_C4R, 696
 - nppiMaximumError_32u_C1R, 696
 - nppiMaximumError_32u_C2R, 697
 - nppiMaximumError_32u_C3R, 697
 - nppiMaximumError_32u_C4R, 698
 - nppiMaximumError_64f_C1R, 698
 - nppiMaximumError_64f_C2R, 698
 - nppiMaximumError_64f_C3R, 699
 - nppiMaximumError_64f_C4R, 699
 - nppiMaximumError_8s_C1R, 700
 - nppiMaximumError_8s_C2R, 700
 - nppiMaximumError_8s_C3R, 701
 - nppiMaximumError_8s_C4R, 701
 - nppiMaximumError_8u_C1R, 701
 - nppiMaximumError_8u_C2R, 702
 - nppiMaximumError_8u_C3R, 702
 - nppiMaximumError_8u_C4R, 703
- image_maximum_relative_error
 - nppiMaximumRelativeError_16s_C1R, 730
 - nppiMaximumRelativeError_16s_C2R, 731
 - nppiMaximumRelativeError_16s_C3R, 731
 - nppiMaximumRelativeError_16s_C4R, 732
 - nppiMaximumRelativeError_16sc_C1R, 732
 - nppiMaximumRelativeError_16sc_C2R, 733
 - nppiMaximumRelativeError_16sc_C3R, 733
 - nppiMaximumRelativeError_16sc_C4R, 733
 - nppiMaximumRelativeError_16u_C1R, 734
 - nppiMaximumRelativeError_16u_C2R, 734
 - nppiMaximumRelativeError_16u_C3R, 735
 - nppiMaximumRelativeError_16u_C4R, 735
 - nppiMaximumRelativeError_32f_C1R, 736
 - nppiMaximumRelativeError_32f_C2R, 736
 - nppiMaximumRelativeError_32f_C3R, 737
 - nppiMaximumRelativeError_32f_C4R, 737
 - nppiMaximumRelativeError_32fc_C1R, 738
 - nppiMaximumRelativeError_32fc_C2R, 738
 - nppiMaximumRelativeError_32fc_C3R, 738
 - nppiMaximumRelativeError_32fc_C4R, 739
 - nppiMaximumRelativeError_32s_C1R, 739
 - nppiMaximumRelativeError_32s_C2R, 740
 - nppiMaximumRelativeError_32s_C3R, 740
 - nppiMaximumRelativeError_32s_C4R, 741
 - nppiMaximumRelativeError_32sc_C1R, 741
 - nppiMaximumRelativeError_32sc_C2R, 742
 - nppiMaximumRelativeError_32sc_C3R, 742
 - nppiMaximumRelativeError_32sc_C4R, 743
 - nppiMaximumRelativeError_32u_C1R, 743
 - nppiMaximumRelativeError_32u_C2R, 743
 - nppiMaximumRelativeError_32u_C3R, 744
 - nppiMaximumRelativeError_32u_C4R, 744
 - nppiMaximumRelativeError_64f_C1R, 745

- nppiMaximumRelativeError_64f_C2R, 745
- nppiMaximumRelativeError_64f_C3R, 746
- nppiMaximumRelativeError_64f_C4R, 746
- nppiMaximumRelativeError_8s_C1R, 747
- nppiMaximumRelativeError_8s_C2R, 747
- nppiMaximumRelativeError_8s_C3R, 748
- nppiMaximumRelativeError_8s_C4R, 748
- nppiMaximumRelativeError_8u_C1R, 748
- nppiMaximumRelativeError_8u_C2R, 749
- nppiMaximumRelativeError_8u_C3R, 749
- nppiMaximumRelativeError_8u_C4R, 750
- image_mean
 - nppiMean_16s_AC4R, 221
 - nppiMean_16s_C1R, 221
 - nppiMean_16s_C3R, 221
 - nppiMean_16s_C4R, 222
 - nppiMean_16u_AC4R, 222
 - nppiMean_16u_C1MR, 222
 - nppiMean_16u_C1R, 223
 - nppiMean_16u_C3CMR, 223
 - nppiMean_16u_C3R, 223
 - nppiMean_16u_C4R, 224
 - nppiMean_32f_AC4R, 224
 - nppiMean_32f_C1MR, 225
 - nppiMean_32f_C1R, 225
 - nppiMean_32f_C3CMR, 225
 - nppiMean_32f_C3R, 226
 - nppiMean_32f_C4R, 226
 - nppiMean_8s_C1MR, 227
 - nppiMean_8s_C3CMR, 227
 - nppiMean_8u_AC4R, 228
 - nppiMean_8u_C1MR, 228
 - nppiMean_8u_C1R, 228
 - nppiMean_8u_C3CMR, 229
 - nppiMean_8u_C3R, 229
 - nppiMean_8u_C4R, 230
 - nppiMeanGetBufferHostSize_16s_AC4R, 230
 - nppiMeanGetBufferHostSize_16s_C1R, 230
 - nppiMeanGetBufferHostSize_16s_C3R, 231
 - nppiMeanGetBufferHostSize_16s_C4R, 231
 - nppiMeanGetBufferHostSize_16u_AC4R, 231
 - nppiMeanGetBufferHostSize_16u_C1MR, 231
 - nppiMeanGetBufferHostSize_16u_C1R, 232
 - nppiMeanGetBufferHostSize_16u_C3CMR, 232
 - nppiMeanGetBufferHostSize_16u_C3R, 232
 - nppiMeanGetBufferHostSize_16u_C4R, 233
 - nppiMeanGetBufferHostSize_32f_AC4R, 233
 - nppiMeanGetBufferHostSize_32f_C1MR, 233
 - nppiMeanGetBufferHostSize_32f_C1R, 233
 - nppiMeanGetBufferHostSize_32f_C3CMR, 234
 - nppiMeanGetBufferHostSize_32f_C3R, 234
 - nppiMeanGetBufferHostSize_32f_C4R, 234
 - nppiMeanGetBufferHostSize_8s_C1MR, 235
 - nppiMeanGetBufferHostSize_8u_AC4R, 235
 - nppiMeanGetBufferHostSize_8u_C1MR, 235
 - nppiMeanGetBufferHostSize_8u_C1R, 236
 - nppiMeanGetBufferHostSize_8u_C3CMR, 236
 - nppiMeanGetBufferHostSize_8u_C3R, 236
 - nppiMeanGetBufferHostSize_8u_C4R, 237
- image_mean_stddev
 - nppiMean_StdDev_16u_C1MR, 241
 - nppiMean_StdDev_16u_C1R, 241
 - nppiMean_StdDev_16u_C3CMR, 242
 - nppiMean_StdDev_16u_C3CR, 242
 - nppiMean_StdDev_32f_C1MR, 243
 - nppiMean_StdDev_32f_C1R, 243
 - nppiMean_StdDev_32f_C3CMR, 244
 - nppiMean_StdDev_32f_C3CR, 244
 - nppiMean_StdDev_8s_C1MR, 245
 - nppiMean_StdDev_8s_C1R, 245
 - nppiMean_StdDev_8s_C3CMR, 246
 - nppiMean_StdDev_8s_C3CR, 246
 - nppiMean_StdDev_8u_C1MR, 247
 - nppiMean_StdDev_8u_C1R, 247
 - nppiMean_StdDev_8u_C3CMR, 248
 - nppiMean_StdDev_8u_C3CR, 248
 - nppiMeanStdDevGetBufferHostSize_16u_C1MR, 249
 - nppiMeanStdDevGetBufferHostSize_16u_C1R, 249
 - nppiMeanStdDevGetBufferHostSize_16u_C3CMR, 249
 - nppiMeanStdDevGetBufferHostSize_16u_C3CR, 250
 - nppiMeanStdDevGetBufferHostSize_32f_C1MR, 250
 - nppiMeanStdDevGetBufferHostSize_32f_C1R, 250
 - nppiMeanStdDevGetBufferHostSize_32f_C3CMR, 251
 - nppiMeanStdDevGetBufferHostSize_32f_C3CR, 251
 - nppiMeanStdDevGetBufferHostSize_8s_C1MR, 251
 - nppiMeanStdDevGetBufferHostSize_8s_C1R, 251
 - nppiMeanStdDevGetBufferHostSize_8s_C3CMR, 252
 - nppiMeanStdDevGetBufferHostSize_8s_C3CR, 252
 - nppiMeanStdDevGetBufferHostSize_8u_C1MR, 252

- nppiMeanStdDevGetBufferHostSize_8u_C1R, 253
- nppiMeanStdDevGetBufferHostSize_8u_C3CMR, 253
- nppiMeanStdDevGetBufferHostSize_8u_C3CR, 253
- image_min
 - nppiMin_16s_AC4R, 134
 - nppiMin_16s_C1R, 134
 - nppiMin_16s_C3R, 135
 - nppiMin_16s_C4R, 135
 - nppiMin_16u_AC4R, 135
 - nppiMin_16u_C1R, 136
 - nppiMin_16u_C3R, 136
 - nppiMin_16u_C4R, 137
 - nppiMin_32f_AC4R, 137
 - nppiMin_32f_C1R, 137
 - nppiMin_32f_C3R, 138
 - nppiMin_32f_C4R, 138
 - nppiMin_8u_AC4R, 138
 - nppiMin_8u_C1R, 139
 - nppiMin_8u_C3R, 139
 - nppiMin_8u_C4R, 140
 - nppiMinGetBufferHostSize_16s_AC4R, 140
 - nppiMinGetBufferHostSize_16s_C1R, 140
 - nppiMinGetBufferHostSize_16s_C3R, 140
 - nppiMinGetBufferHostSize_16s_C4R, 141
 - nppiMinGetBufferHostSize_16u_AC4R, 141
 - nppiMinGetBufferHostSize_16u_C1R, 141
 - nppiMinGetBufferHostSize_16u_C3R, 141
 - nppiMinGetBufferHostSize_16u_C4R, 142
 - nppiMinGetBufferHostSize_32f_AC4R, 142
 - nppiMinGetBufferHostSize_32f_C1R, 142
 - nppiMinGetBufferHostSize_32f_C3R, 142
 - nppiMinGetBufferHostSize_32f_C4R, 143
 - nppiMinGetBufferHostSize_8u_AC4R, 143
 - nppiMinGetBufferHostSize_8u_C1R, 143
 - nppiMinGetBufferHostSize_8u_C3R, 143
 - nppiMinGetBufferHostSize_8u_C4R, 144
- image_min_index
 - nppiMinIndx_16s_AC4R, 147
 - nppiMinIndx_16s_C1R, 148
 - nppiMinIndx_16s_C3R, 148
 - nppiMinIndx_16s_C4R, 148
 - nppiMinIndx_16u_AC4R, 149
 - nppiMinIndx_16u_C1R, 149
 - nppiMinIndx_16u_C3R, 150
 - nppiMinIndx_16u_C4R, 150
 - nppiMinIndx_32f_AC4R, 150
 - nppiMinIndx_32f_C1R, 151
 - nppiMinIndx_32f_C3R, 151
 - nppiMinIndx_32f_C4R, 152
 - nppiMinIndx_8u_AC4R, 152
 - nppiMinIndx_8u_C1R, 152
 - nppiMinIndx_8u_C3R, 153
 - nppiMinIndx_8u_C4R, 153
 - nppiMinIndxGetBufferHostSize_16s_AC4R, 154
 - nppiMinIndxGetBufferHostSize_16s_C1R, 154
 - nppiMinIndxGetBufferHostSize_16s_C3R, 155
 - nppiMinIndxGetBufferHostSize_16s_C4R, 155
 - nppiMinIndxGetBufferHostSize_16u_AC4R, 155
 - nppiMinIndxGetBufferHostSize_16u_C1R, 155
 - nppiMinIndxGetBufferHostSize_16u_C3R, 155
 - nppiMinIndxGetBufferHostSize_16u_C4R, 156
 - nppiMinIndxGetBufferHostSize_32f_AC4R, 156
 - nppiMinIndxGetBufferHostSize_32f_C1R, 156
 - nppiMinIndxGetBufferHostSize_32f_C3R, 157
 - nppiMinIndxGetBufferHostSize_32f_C4R, 157
 - nppiMinIndxGetBufferHostSize_8u_AC4R, 157
 - nppiMinIndxGetBufferHostSize_8u_C1R, 157
 - nppiMinIndxGetBufferHostSize_8u_C3R, 158
 - nppiMinIndxGetBufferHostSize_8u_C4R, 158
- image_min_max
 - nppiMinMax_16s_AC4R, 188
 - nppiMinMax_16s_C1R, 188
 - nppiMinMax_16s_C3R, 189
 - nppiMinMax_16s_C4R, 189
 - nppiMinMax_16u_AC4R, 190
 - nppiMinMax_16u_C1R, 190
 - nppiMinMax_16u_C3R, 190
 - nppiMinMax_16u_C4R, 191
 - nppiMinMax_32f_AC4R, 191
 - nppiMinMax_32f_C1R, 192
 - nppiMinMax_32f_C3R, 192
 - nppiMinMax_32f_C4R, 192
 - nppiMinMax_8u_AC4R, 193
 - nppiMinMax_8u_C1R, 193
 - nppiMinMax_8u_C3R, 194
 - nppiMinMax_8u_C4R, 194
 - nppiMinMaxGetBufferHostSize_16s_AC4R, 194
 - nppiMinMaxGetBufferHostSize_16s_C1R, 195
 - nppiMinMaxGetBufferHostSize_16s_C3R, 195

- nppiMinMaxGetBufferHostSize_16s_C4R, 195
- nppiMinMaxGetBufferHostSize_16u_AC4R, 196
- nppiMinMaxGetBufferHostSize_16u_C1R, 196
- nppiMinMaxGetBufferHostSize_16u_C3R, 196
- nppiMinMaxGetBufferHostSize_16u_C4R, 196
- nppiMinMaxGetBufferHostSize_32f_AC4R, 197
- nppiMinMaxGetBufferHostSize_32f_C1R, 197
- nppiMinMaxGetBufferHostSize_32f_C3R, 197
- nppiMinMaxGetBufferHostSize_32f_C4R, 198
- nppiMinMaxGetBufferHostSize_8u_AC4R, 198
- nppiMinMaxGetBufferHostSize_8u_C1R, 198
- nppiMinMaxGetBufferHostSize_8u_C3R, 198
- nppiMinMaxGetBufferHostSize_8u_C4R, 199
- image_min_max_index
 - nppiMinMaxIdx_16u_C1MR, 203
 - nppiMinMaxIdx_16u_C1R, 204
 - nppiMinMaxIdx_16u_C3CMR, 204
 - nppiMinMaxIdx_16u_C3CR, 205
 - nppiMinMaxIdx_32f_C1MR, 205
 - nppiMinMaxIdx_32f_C1R, 206
 - nppiMinMaxIdx_32f_C3CMR, 206
 - nppiMinMaxIdx_32f_C3CR, 207
 - nppiMinMaxIdx_8s_C1MR, 208
 - nppiMinMaxIdx_8s_C1R, 208
 - nppiMinMaxIdx_8s_C3CMR, 209
 - nppiMinMaxIdx_8s_C3CR, 209
 - nppiMinMaxIdx_8u_C1MR, 210
 - nppiMinMaxIdx_8u_C1R, 210
 - nppiMinMaxIdx_8u_C3CMR, 211
 - nppiMinMaxIdx_8u_C3CR, 211
 - nppiMinMaxIdxGetBufferHostSize_16u_C1MR, 212
 - nppiMinMaxIdxGetBufferHostSize_16u_C1R, 212
 - nppiMinMaxIdxGetBufferHostSize_16u_C3CMR, 212
 - nppiMinMaxIdxGetBufferHostSize_16u_C3CR, 213
 - nppiMinMaxIdxGetBufferHostSize_32f_C1MR, 213
 - nppiMinMaxIdxGetBufferHostSize_32f_C1R, 213
 - nppiMinMaxIdxGetBufferHostSize_32f_C3CMR, 214
- nppiMinMaxIdxGetBufferHostSize_32f_C3CR, 214
- nppiMinMaxIdxGetBufferHostSize_8s_C1MR, 214
- nppiMinMaxIdxGetBufferHostSize_8s_C1R, 214
- nppiMinMaxIdxGetBufferHostSize_8s_C3CMR, 215
- nppiMinMaxIdxGetBufferHostSize_8s_C3CR, 215
- nppiMinMaxIdxGetBufferHostSize_8u_C1MR, 215
- nppiMinMaxIdxGetBufferHostSize_8u_C1R, 216
- nppiMinMaxIdxGetBufferHostSize_8u_C3CMR, 216
- nppiMinMaxIdxGetBufferHostSize_8u_C3CR, 216
- image_minevery
 - nppiMinEvery_16s_AC4IR, 497
 - nppiMinEvery_16s_C1IR, 497
 - nppiMinEvery_16s_C3IR, 498
 - nppiMinEvery_16s_C4IR, 498
 - nppiMinEvery_16u_AC4IR, 498
 - nppiMinEvery_16u_C1IR, 499
 - nppiMinEvery_16u_C3IR, 499
 - nppiMinEvery_16u_C4IR, 499
 - nppiMinEvery_32f_AC4IR, 500
 - nppiMinEvery_32f_C1IR, 500
 - nppiMinEvery_32f_C3IR, 500
 - nppiMinEvery_32f_C4IR, 501
 - nppiMinEvery_8u_AC4IR, 501
 - nppiMinEvery_8u_C1IR, 501
 - nppiMinEvery_8u_C3IR, 502
 - nppiMinEvery_8u_C4IR, 502
- image_quality_index
 - nppiQualityIndex_16u32f_AC4R, 674
 - nppiQualityIndex_16u32f_C1R, 674
 - nppiQualityIndex_16u32f_C3R, 675
 - nppiQualityIndex_32f_AC4R, 675
 - nppiQualityIndex_32f_C1R, 676
 - nppiQualityIndex_32f_C3R, 676
 - nppiQualityIndex_8u32f_AC4R, 676
 - nppiQualityIndex_8u32f_C1R, 677
 - nppiQualityIndex_8u32f_C3R, 677
 - nppiQualityIndexGetBufferHostSize_16u32f_AC4R, 678
 - nppiQualityIndexGetBufferHostSize_16u32f_C1R, 678
 - nppiQualityIndexGetBufferHostSize_16u32f_C3R, 678
 - nppiQualityIndexGetBufferHostSize_32f_AC4R, 679

- nppiQualityIndexGetBufferHostSize_32f_-C1R, 679
- nppiQualityIndexGetBufferHostSize_32f_-C3R, 679
- nppiQualityIndexGetBufferHostSize_8u32f_-AC4R, 680
- nppiQualityIndexGetBufferHostSize_8u32f_-C1R, 680
- nppiQualityIndexGetBufferHostSize_8u32f_-C3R, 680
- image_rectstddev
 - nppiRectStdDev_32f_C1R, 508
 - nppiRectStdDev_32s32f_C1R, 509
 - nppiRectStdDev_32s_C1RSfs, 509
- image_sqrintegral
 - nppiSqrIntegral_8u32f64f_C1R, 505
 - nppiSqrIntegral_8u32s64f_C1R, 506
 - nppiSqrIntegral_8u32s_C1R, 506
- image_statistics_functions
 - nppiAverageErrorGetBufferHostSize_16s_-C1R, 66
 - nppiAverageErrorGetBufferHostSize_16s_-C2R, 66
 - nppiAverageErrorGetBufferHostSize_16s_-C3R, 66
 - nppiAverageErrorGetBufferHostSize_16s_-C4R, 67
 - nppiAverageErrorGetBufferHostSize_16sc_-C1R, 67
 - nppiAverageErrorGetBufferHostSize_16sc_-C2R, 67
 - nppiAverageErrorGetBufferHostSize_16sc_-C3R, 67
 - nppiAverageErrorGetBufferHostSize_16sc_-C4R, 68
 - nppiAverageErrorGetBufferHostSize_16u_-C1R, 68
 - nppiAverageErrorGetBufferHostSize_16u_-C2R, 68
 - nppiAverageErrorGetBufferHostSize_16u_-C3R, 69
 - nppiAverageErrorGetBufferHostSize_16u_-C4R, 69
 - nppiAverageErrorGetBufferHostSize_32f_-C1R, 69
 - nppiAverageErrorGetBufferHostSize_32f_-C2R, 69
 - nppiAverageErrorGetBufferHostSize_32f_-C3R, 70
 - nppiAverageErrorGetBufferHostSize_32f_-C4R, 70
 - nppiAverageErrorGetBufferHostSize_32fc_-C1R, 70
 - nppiAverageErrorGetBufferHostSize_32fc_-C2R, 71
 - nppiAverageErrorGetBufferHostSize_32fc_-C3R, 71
 - nppiAverageErrorGetBufferHostSize_32fc_-C4R, 71
 - nppiAverageErrorGetBufferHostSize_32s_-C1R, 71
 - nppiAverageErrorGetBufferHostSize_32s_-C2R, 72
 - nppiAverageErrorGetBufferHostSize_32s_-C3R, 72
 - nppiAverageErrorGetBufferHostSize_32s_-C4R, 72
 - nppiAverageErrorGetBufferHostSize_32sc_-C1R, 73
 - nppiAverageErrorGetBufferHostSize_32sc_-C2R, 73
 - nppiAverageErrorGetBufferHostSize_32sc_-C3R, 73
 - nppiAverageErrorGetBufferHostSize_32sc_-C4R, 73
 - nppiAverageErrorGetBufferHostSize_32u_-C1R, 74
 - nppiAverageErrorGetBufferHostSize_32u_-C2R, 74
 - nppiAverageErrorGetBufferHostSize_32u_-C3R, 74
 - nppiAverageErrorGetBufferHostSize_32u_-C4R, 75
 - nppiAverageErrorGetBufferHostSize_64f_-C1R, 75
 - nppiAverageErrorGetBufferHostSize_64f_-C2R, 75
 - nppiAverageErrorGetBufferHostSize_64f_-C3R, 75
 - nppiAverageErrorGetBufferHostSize_64f_-C4R, 76
 - nppiAverageErrorGetBufferHostSize_8s_-C1R, 76
 - nppiAverageErrorGetBufferHostSize_8s_-C2R, 76
 - nppiAverageErrorGetBufferHostSize_8s_-C3R, 77
 - nppiAverageErrorGetBufferHostSize_8s_-C4R, 77
 - nppiAverageErrorGetBufferHostSize_8u_-C1R, 77
 - nppiAverageErrorGetBufferHostSize_8u_-C2R, 77
 - nppiAverageErrorGetBufferHostSize_8u_-C3R, 78
 - nppiAverageErrorGetBufferHostSize_8u_-C4R, 78

- nppiAverageRelativeErrorGetBufferHostSize_16s_C1R, 78
 nppiAverageRelativeErrorGetBufferHostSize_16s_C2R, 79
 nppiAverageRelativeErrorGetBufferHostSize_16s_C3R, 79
 nppiAverageRelativeErrorGetBufferHostSize_16s_C4R, 79
 nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R, 79
 nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R, 80
 nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R, 80
 nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R, 80
 nppiAverageRelativeErrorGetBufferHostSize_16u_C1R, 81
 nppiAverageRelativeErrorGetBufferHostSize_16u_C2R, 81
 nppiAverageRelativeErrorGetBufferHostSize_16u_C3R, 81
 nppiAverageRelativeErrorGetBufferHostSize_16u_C4R, 81
 nppiAverageRelativeErrorGetBufferHostSize_32f_C1R, 82
 nppiAverageRelativeErrorGetBufferHostSize_32f_C2R, 82
 nppiAverageRelativeErrorGetBufferHostSize_32f_C3R, 82
 nppiAverageRelativeErrorGetBufferHostSize_32f_C4R, 83
 nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R, 83
 nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R, 83
 nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R, 83
 nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R, 84
 nppiAverageRelativeErrorGetBufferHostSize_32s_C1R, 84
 nppiAverageRelativeErrorGetBufferHostSize_32s_C2R, 84
 nppiAverageRelativeErrorGetBufferHostSize_32s_C3R, 85
 nppiAverageRelativeErrorGetBufferHostSize_32s_C4R, 85
 nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R, 85
 nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R, 85
 nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R, 86
 nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R, 86
 nppiAverageRelativeErrorGetBufferHostSize_32u_C1R, 86
 nppiAverageRelativeErrorGetBufferHostSize_32u_C2R, 87
 nppiAverageRelativeErrorGetBufferHostSize_32u_C3R, 87
 nppiAverageRelativeErrorGetBufferHostSize_32u_C4R, 87
 nppiAverageRelativeErrorGetBufferHostSize_64f_C1R, 87
 nppiAverageRelativeErrorGetBufferHostSize_64f_C2R, 88
 nppiAverageRelativeErrorGetBufferHostSize_64f_C3R, 88
 nppiAverageRelativeErrorGetBufferHostSize_64f_C4R, 88
 nppiAverageRelativeErrorGetBufferHostSize_8s_C1R, 89
 nppiAverageRelativeErrorGetBufferHostSize_8s_C2R, 89
 nppiAverageRelativeErrorGetBufferHostSize_8s_C3R, 89
 nppiAverageRelativeErrorGetBufferHostSize_8s_C4R, 89
 nppiAverageRelativeErrorGetBufferHostSize_8u_C1R, 90
 nppiAverageRelativeErrorGetBufferHostSize_8u_C2R, 90
 nppiAverageRelativeErrorGetBufferHostSize_8u_C3R, 90
 nppiAverageRelativeErrorGetBufferHostSize_8u_C4R, 91
 nppiMaximumErrorGetBufferHostSize_16s_C1R, 91
 nppiMaximumErrorGetBufferHostSize_16s_C2R, 91
 nppiMaximumErrorGetBufferHostSize_16s_C3R, 91
 nppiMaximumErrorGetBufferHostSize_16s_C4R, 92
 nppiMaximumErrorGetBufferHostSize_16sc_C1R, 92
 nppiMaximumErrorGetBufferHostSize_16sc_C2R, 92
 nppiMaximumErrorGetBufferHostSize_16sc_C3R, 93
 nppiMaximumErrorGetBufferHostSize_16sc_C4R, 93
 nppiMaximumErrorGetBufferHostSize_16u_C1R, 93
 nppiMaximumErrorGetBufferHostSize_16u_C2R, 93

- nppiMaximumErrorGetBufferHostSize_16u_-
 C3R, 94
 nppiMaximumErrorGetBufferHostSize_16u_-
 C4R, 94
 nppiMaximumErrorGetBufferHostSize_32f_-
 C1R, 94
 nppiMaximumErrorGetBufferHostSize_32f_-
 C2R, 95
 nppiMaximumErrorGetBufferHostSize_32f_-
 C3R, 95
 nppiMaximumErrorGetBufferHostSize_32f_-
 C4R, 95
 nppiMaximumErrorGetBufferHostSize_-
 32fc_C1R, 95
 nppiMaximumErrorGetBufferHostSize_-
 32fc_C2R, 96
 nppiMaximumErrorGetBufferHostSize_-
 32fc_C3R, 96
 nppiMaximumErrorGetBufferHostSize_-
 32fc_C4R, 96
 nppiMaximumErrorGetBufferHostSize_32s_-
 C1R, 97
 nppiMaximumErrorGetBufferHostSize_32s_-
 C2R, 97
 nppiMaximumErrorGetBufferHostSize_32s_-
 C3R, 97
 nppiMaximumErrorGetBufferHostSize_32s_-
 C4R, 97
 nppiMaximumErrorGetBufferHostSize_-
 32sc_C1R, 98
 nppiMaximumErrorGetBufferHostSize_-
 32sc_C2R, 98
 nppiMaximumErrorGetBufferHostSize_-
 32sc_C3R, 98
 nppiMaximumErrorGetBufferHostSize_-
 32sc_C4R, 99
 nppiMaximumErrorGetBufferHostSize_32u_-
 C1R, 99
 nppiMaximumErrorGetBufferHostSize_32u_-
 C2R, 99
 nppiMaximumErrorGetBufferHostSize_32u_-
 C3R, 99
 nppiMaximumErrorGetBufferHostSize_32u_-
 C4R, 100
 nppiMaximumErrorGetBufferHostSize_64f_-
 C1R, 100
 nppiMaximumErrorGetBufferHostSize_64f_-
 C2R, 100
 nppiMaximumErrorGetBufferHostSize_64f_-
 C3R, 101
 nppiMaximumErrorGetBufferHostSize_64f_-
 C4R, 101
 nppiMaximumErrorGetBufferHostSize_8s_-
 C1R, 101
 nppiMaximumErrorGetBufferHostSize_8s_-
 C2R, 101
 nppiMaximumErrorGetBufferHostSize_8s_-
 C3R, 102
 nppiMaximumErrorGetBufferHostSize_8s_-
 C4R, 102
 nppiMaximumErrorGetBufferHostSize_8u_-
 C1R, 102
 nppiMaximumErrorGetBufferHostSize_8u_-
 C2R, 103
 nppiMaximumErrorGetBufferHostSize_8u_-
 C3R, 103
 nppiMaximumErrorGetBufferHostSize_8u_-
 C4R, 103
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16s_C1R, 103
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16s_C2R, 104
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16s_C3R, 104
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16s_C4R, 104
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16sc_C1R, 105
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16sc_C2R, 105
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16sc_C3R, 105
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16sc_C4R, 105
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16u_C1R, 106
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16u_C2R, 106
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16u_C3R, 106
 nppiMaximumRelativeErrorGetBufferHostSize_-
 16u_C4R, 107
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32f_C1R, 107
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32f_C2R, 107
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32f_C3R, 107
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32f_C4R, 108
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32fc_C1R, 108
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32fc_C2R, 108
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32fc_C3R, 109
 nppiMaximumRelativeErrorGetBufferHostSize_-
 32fc_C4R, 109

- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C1R, 109
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C2R, 109
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C3R, 110
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C4R, 110
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C1R, 110
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C2R, 111
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C3R, 111
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C4R, 111
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C1R, 111
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C2R, 112
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C3R, 112
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C4R, 112
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C1R, 113
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C2R, 113
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C3R, 113
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C4R, 113
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C1R, 114
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C2R, 114
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C3R, 114
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C4R, 115
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C1R, 115
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C2R, 115
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C3R, 115
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C4R, 116
- image_sum
 - nppiSum_16s_AC4R, 120
 - nppiSum_16s_C1R, 120
 - nppiSum_16s_C3R, 120
 - nppiSum_16s_C4R, 121
 - nppiSum_16u_AC4R, 121
 - nppiSum_16u_C1R, 121
 - nppiSum_16u_C3R, 122
 - nppiSum_16u_C4R, 122
 - nppiSum_32f_AC4R, 122
 - nppiSum_32f_C1R, 123
 - nppiSum_32f_C3R, 123
 - nppiSum_32f_C4R, 123
 - nppiSum_8u64s_C1R, 124
 - nppiSum_8u64s_C4R, 124
 - nppiSum_8u_AC4R, 125
 - nppiSum_8u_C1R, 125
 - nppiSum_8u_C3R, 125
 - nppiSum_8u_C4R, 126
 - nppiSumGetBufferHostSize_16s_AC4R, 126
 - nppiSumGetBufferHostSize_16s_C1R, 126
 - nppiSumGetBufferHostSize_16s_C3R, 127
 - nppiSumGetBufferHostSize_16s_C4R, 127
 - nppiSumGetBufferHostSize_16u_AC4R, 127
 - nppiSumGetBufferHostSize_16u_C1R, 127
 - nppiSumGetBufferHostSize_16u_C3R, 128
 - nppiSumGetBufferHostSize_16u_C4R, 128
 - nppiSumGetBufferHostSize_32f_AC4R, 128
 - nppiSumGetBufferHostSize_32f_C1R, 129
 - nppiSumGetBufferHostSize_32f_C3R, 129
 - nppiSumGetBufferHostSize_32f_C4R, 129
 - nppiSumGetBufferHostSize_8u64s_C1R, 129
 - nppiSumGetBufferHostSize_8u64s_C4R, 130
 - nppiSumGetBufferHostSize_8u_AC4R, 130
 - nppiSumGetBufferHostSize_8u_C1R, 130
 - nppiSumGetBufferHostSize_8u_C3R, 131
 - nppiSumGetBufferHostSize_8u_C4R, 131
- Integral, 503
- Linear Transforms, 775
- major
 - NppLibraryVersion, 788
- Max, 159
- MaxEvery, 489
- MaximumError, 681
- MaximumRelativeError, 727
- MaxIdx, 172
- Mean, 217
- Mean_StdDev, 238
- Min, 132
- MinEvery, 496
- MinIdx, 145
- MinMax, 186
- MinMaxIdx, 200
- minor
 - NppLibraryVersion, 788
- Norm_Inf, 256
- Norm_L1, 278

- Norm_L2, [299](#)
- NormDiff_Inf, [320](#)
- NormDiff_L1, [343](#)
- NormDiff_L2, [366](#)
- NormRel_Inf, [389](#)
- NormRel_L1, [412](#)
- NormRel_L2, [435](#)
- NPP Core, [27](#)
- NPP Type Definitions and Constants, [31](#)
- Npp16s
 - [npp_basic_types](#), [47](#)
- Npp16sc
 - [npp_basic_types](#), [49](#)
- Npp16u
 - [npp_basic_types](#), [47](#)
- Npp16uc
 - [npp_basic_types](#), [49](#)
- Npp32f
 - [npp_basic_types](#), [47](#)
- Npp32fc
 - [npp_basic_types](#), [47](#)
- Npp32s
 - [npp_basic_types](#), [47](#)
- Npp32sc
 - [npp_basic_types](#), [47](#)
- Npp32u
 - [npp_basic_types](#), [48](#)
- Npp32uc
 - [npp_basic_types](#), [48](#)
- Npp64f
 - [npp_basic_types](#), [48](#)
- Npp64fc
 - [npp_basic_types](#), [48](#)
- Npp64s
 - [npp_basic_types](#), [48](#)
- Npp64sc
 - [npp_basic_types](#), [48](#)
- Npp64u
 - [npp_basic_types](#), [48](#)
- Npp8s
 - [npp_basic_types](#), [48](#)
- Npp8u
 - [npp_basic_types](#), [48](#)
- Npp8uc
 - [npp_basic_types](#), [49](#)
- NPP_AFFINE_QUAD_INCORRECT_WARNING
 - [typedefs_npp](#), [44](#)
- NPP_ALG_HINT_ACCURATE
 - [typedefs_npp](#), [39](#)
- NPP_ALG_HINT_FAST
 - [typedefs_npp](#), [39](#)
- NPP_ALG_HINT_NONE
 - [typedefs_npp](#), [39](#)
- NPP_ALIGNMENT_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_ANCHOR_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_BAD_ARGUMENT_ERROR
 - [typedefs_npp](#), [44](#)
- NPP_BORDER_CONSTANT
 - [typedefs_npp](#), [40](#)
- NPP_BORDER_MIRROR
 - [typedefs_npp](#), [40](#)
- NPP_BORDER_NONE
 - [typedefs_npp](#), [40](#)
- NPP_BORDER_REPLICATE
 - [typedefs_npp](#), [40](#)
- NPP_BORDER_UNDEFINED
 - [typedefs_npp](#), [40](#)
- NPP_BORDER_WRAP
 - [typedefs_npp](#), [40](#)
- NPP_BOTH_AXIS
 - [typedefs_npp](#), [40](#)
- NPP_CHANNEL_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_CHANNEL_ORDER_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_CMP_EQ
 - [typedefs_npp](#), [39](#)
- NPP_CMP_GREATER
 - [typedefs_npp](#), [39](#)
- NPP_CMP_GREATER_EQ
 - [typedefs_npp](#), [39](#)
- NPP_CMP_LESS
 - [typedefs_npp](#), [38](#)
- NPP_CMP_LESS_EQ
 - [typedefs_npp](#), [38](#)
- NPP_COEFFICIENT_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_COI_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_CONTEXT_MATCH_ERROR
 - [typedefs_npp](#), [44](#)
- NPP_CORRUPTED_DATA_ERROR
 - [typedefs_npp](#), [43](#)
- NPP_CUDA_1_0
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_1_1
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_1_2
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_1_3
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_2_0
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_2_1
 - [typedefs_npp](#), [39](#)
- NPP_CUDA_3_0

- typedefs_npp, 39
- NPP_CUDA_3_2
 - typedefs_npp, 39
- NPP_CUDA_3_5
 - typedefs_npp, 39
- NPP_CUDA_3_7
 - typedefs_npp, 39
- NPP_CUDA_5_0
 - typedefs_npp, 39
- NPP_CUDA_5_2
 - typedefs_npp, 39
- NPP_CUDA_5_3
 - typedefs_npp, 39
- NPP_CUDA_6_0
 - typedefs_npp, 39
- NPP_CUDA_KERNEL_EXECUTION_ERROR
 - typedefs_npp, 43
- NPP_CUDA_NOT_CAPABLE
 - typedefs_npp, 39
- NPP_CUDA_UNKNOWN_VERSION
 - typedefs_npp, 39
- NPP_DATA_TYPE_ERROR
 - typedefs_npp, 44
- NPP_DIVIDE_BY_ZERO_ERROR
 - typedefs_npp, 44
- NPP_DIVIDE_BY_ZERO_WARNING
 - typedefs_npp, 44
- NPP_DIVISOR_ERROR
 - typedefs_npp, 43
- NPP_DOUBLE_SIZE_WARNING
 - typedefs_npp, 44
- NPP_ERROR
 - typedefs_npp, 44
- NPP_ERROR_RESERVED
 - typedefs_npp, 44
- NPP_FFT_FLAG_ERROR
 - typedefs_npp, 44
- NPP_FFT_ORDER_ERROR
 - typedefs_npp, 44
- NPP_FILTER_SCHARR
 - typedefs_npp, 40
- NPP_FILTER_SOBEL
 - typedefs_npp, 40
- NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR
 - typedefs_npp, 43
- NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR
 - typedefs_npp, 43
- NPP_HORIZONTAL_AXIS
 - typedefs_npp, 40
- NPP_INTERPOLATION_ERROR
 - typedefs_npp, 44
- NPP_INVALID_DEVICE_POINTER_ERROR
 - typedefs_npp, 43
- NPP_INVALID_HOST_POINTER_ERROR
 - typedefs_npp, 43
- NPP_LUT_NUMBER_OF_LEVELS_ERROR
 - typedefs_npp, 43
- NPP_LUT_PALETTE_BITSIZE_ERROR
 - typedefs_npp, 43
- NPP_MASK_SIZE_11_X_11
 - typedefs_npp, 41
- NPP_MASK_SIZE_13_X_13
 - typedefs_npp, 41
- NPP_MASK_SIZE_15_X_15
 - typedefs_npp, 41
- NPP_MASK_SIZE_1_X_3
 - typedefs_npp, 41
- NPP_MASK_SIZE_1_X_5
 - typedefs_npp, 41
- NPP_MASK_SIZE_3_X_1
 - typedefs_npp, 41
- NPP_MASK_SIZE_3_X_3
 - typedefs_npp, 41
- NPP_MASK_SIZE_5_X_1
 - typedefs_npp, 41
- NPP_MASK_SIZE_5_X_5
 - typedefs_npp, 41
- NPP_MASK_SIZE_7_X_7
 - typedefs_npp, 41
- NPP_MASK_SIZE_9_X_9
 - typedefs_npp, 41
- NPP_MASK_SIZE_ERROR
 - typedefs_npp, 43
- NPP_MEMCPY_ERROR
 - typedefs_npp, 43
- NPP_MEMFREE_ERROR
 - typedefs_npp, 43
- NPP_MEMORY_ALLOCATION_ERR
 - typedefs_npp, 44
- NPP_MEMSET_ERROR
 - typedefs_npp, 43
- NPP_MIRROR_FLIP_ERROR
 - typedefs_npp, 44
- NPP_MISALIGNED_DST_ROI_WARNING
 - typedefs_npp, 44
- NPP_MOMENT_00_ZERO_ERROR
 - typedefs_npp, 44
- NPP_NO_ERROR
 - typedefs_npp, 44
- NPP_NO_MEMORY_ERROR
 - typedefs_npp, 44
- NPP_NO_OPERATION_WARNING
 - typedefs_npp, 44
- NPP_NOT_EVEN_STEP_ERROR
 - typedefs_npp, 43
- NPP_NOT_IMPLEMENTED_ERROR

- typedefs_npp, 44
- NPP_NOT_SUFFICIENT_COMPUTE_-
CAPABILITY
typedefs_npp, 43
- NPP_NOT_SUPPORTED_MODE_ERROR
typedefs_npp, 43
- NPP_NULL_POINTER_ERROR
typedefs_npp, 44
- NPP_NUMBER_OF_CHANNELS_ERROR
typedefs_npp, 43
- NPP_OUT_OFF_RANGE_ERROR
typedefs_npp, 44
- NPP_OVERFLOW_ERROR
typedefs_npp, 43
- NPP_QUADRANGLE_ERROR
typedefs_npp, 43
- NPP_QUALITY_INDEX_ERROR
typedefs_npp, 43
- NPP_RANGE_ERROR
typedefs_npp, 44
- NPP_RECTANGLE_ERROR
typedefs_npp, 43
- NPP_RESIZE_FACTOR_ERROR
typedefs_npp, 44
- NPP_RESIZE_NO_OPERATION_ERROR
typedefs_npp, 43
- NPP_RND_FINANCIAL
typedefs_npp, 42
- NPP_RND_NEAR
typedefs_npp, 42
- NPP_RND_ZERO
typedefs_npp, 42
- NPP_ROUND_MODE_NOT_SUPPORTED_-
ERROR
typedefs_npp, 43
- NPP_ROUND_NEAREST_TIES_AWAY_-
FROM_ZERO
typedefs_npp, 42
- NPP_ROUND_NEAREST_TIES_TO_EVEN
typedefs_npp, 42
- NPP_ROUND_TOWARD_ZERO
typedefs_npp, 42
- NPP_SCALE_RANGE_ERROR
typedefs_npp, 44
- NPP_SIZE_ERROR
typedefs_npp, 44
- NPP_STEP_ERROR
typedefs_npp, 44
- NPP_STRIDE_ERROR
typedefs_npp, 43
- NPP_SUCCESS
typedefs_npp, 44
- NPP_TEXTURE_BIND_ERROR
typedefs_npp, 43
- NPP_THRESHOLD_ERROR
typedefs_npp, 44
- NPP_THRESHOLD_NEGATIVE_LEVEL_-
ERROR
typedefs_npp, 44
- NPP_VERTICAL_AXIS
typedefs_npp, 40
- NPP_WRONG_INTERSECTION_QUAD_-
WARNING
typedefs_npp, 44
- NPP_WRONG_INTERSECTION_ROI_ERROR
typedefs_npp, 43
- NPP_WRONG_INTERSECTION_ROI_-
WARNING
typedefs_npp, 44
- NPP_ZC_MODE_NOT_SUPPORTED_ERROR
typedefs_npp, 43
- NPP_ZERO_MASK_VALUE_ERROR
typedefs_npp, 43
- NPP_ALIGN_16, 779
im, 779
re, 780
- NPP_ALIGN_8, 781
im, 781
re, 781, 782
- npp_basic_types
__align__, 48, 49
Npp16s, 47
Npp16sc, 49
Npp16u, 47
Npp16uc, 49
Npp32f, 47
Npp32fc, 47
Npp32s, 47
Npp32sc, 47
Npp32u, 48
Npp32uc, 48
Npp64f, 48
Npp64fc, 48
Npp64s, 48
Npp64sc, 48
Npp64u, 48
Npp8s, 48
Npp8u, 48
Npp8uc, 49
- NPP_MAX_16S
typedefs_npp, 37
- NPP_MAX_16U
typedefs_npp, 37
- NPP_MAX_32S
typedefs_npp, 37
- NPP_MAX_32U
typedefs_npp, 37
- NPP_MAX_64S

- typedefs_npp, 37
- NPP_MAX_64U
 - typedefs_npp, 37
- NPP_MAX_8S
 - typedefs_npp, 37
- NPP_MAX_8U
 - typedefs_npp, 37
- NPP_MAXABS_32F
 - typedefs_npp, 37
- NPP_MAXABS_64F
 - typedefs_npp, 37
- NPP_MIN_16S
 - typedefs_npp, 37
- NPP_MIN_16U
 - typedefs_npp, 38
- NPP_MIN_32S
 - typedefs_npp, 38
- NPP_MIN_32U
 - typedefs_npp, 38
- NPP_MIN_64S
 - typedefs_npp, 38
- NPP_MIN_64U
 - typedefs_npp, 38
- NPP_MIN_8S
 - typedefs_npp, 38
- NPP_MIN_8U
 - typedefs_npp, 38
- NPP_MINABS_32F
 - typedefs_npp, 38
- NPP_MINABS_64F
 - typedefs_npp, 38
- NppCmpOp
 - typedefs_npp, 38
- nppGetGpuComputeCapability
 - core_npp, 28
- nppGetGpuDeviceProperties
 - core_npp, 28
- nppGetGpuName
 - core_npp, 28
- nppGetGpuNumSMs
 - core_npp, 28
- nppGetLibVersion
 - core_npp, 28
- nppGetMaxThreadsPerBlock
 - core_npp, 29
- nppGetMaxThreadsPerSM
 - core_npp, 29
- nppGetStream
 - core_npp, 29
- nppGetStreamMaxThreadsPerSM
 - core_npp, 29
- nppGetStreamNumSMs
 - core_npp, 29
- NppGpuComputeCapability
 - typedefs_npp, 39
- NppHintAlgorithm
 - typedefs_npp, 39
- NPPI_BAYER_BGGR
 - typedefs_npp, 40
- NPPI_BAYER_GBRG
 - typedefs_npp, 40
- NPPI_BAYER_GRBG
 - typedefs_npp, 40
- NPPI_BAYER_RRGB
 - typedefs_npp, 40
- NPPI_INTER_CUBIC
 - typedefs_npp, 41
- NPPI_INTER_CUBIC2P_B05C03
 - typedefs_npp, 41
- NPPI_INTER_CUBIC2P_BSPLINE
 - typedefs_npp, 41
- NPPI_INTER_CUBIC2P_CATMULLROM
 - typedefs_npp, 41
- NPPI_INTER_LANCZOS
 - typedefs_npp, 41
- NPPI_INTER_LANCZOS3_ADVANCED
 - typedefs_npp, 41
- NPPI_INTER_LINEAR
 - typedefs_npp, 41
- NPPI_INTER_NN
 - typedefs_npp, 41
- NPPI_INTER_SUPER
 - typedefs_npp, 41
- NPPI_INTER_UNDEFINED
 - typedefs_npp, 41
- NPPI_OP_ALPHA_ATOP
 - typedefs_npp, 39
- NPPI_OP_ALPHA_ATOP_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_IN
 - typedefs_npp, 39
- NPPI_OP_ALPHA_IN_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_OUT
 - typedefs_npp, 39
- NPPI_OP_ALPHA_OUT_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_OVER
 - typedefs_npp, 39
- NPPI_OP_ALPHA_OVER_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_PLUS
 - typedefs_npp, 39
- NPPI_OP_ALPHA_PLUS_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_PREMUL
 - typedefs_npp, 40
- NPPI_OP_ALPHA_XOR

- typedefs_npp, 39
- NPPI_OP_ALPHA_XOR_PREMUL
 - typedefs_npp, 40
- NPPI_SMOOTH_EDGE
 - typedefs_npp, 41
- nppiACTable
 - typedefs_npp, 41
- NppiAlphaOp
 - typedefs_npp, 39
- nppiAverageError_16s_C1R
 - image_average_error, 707
- nppiAverageError_16s_C2R
 - image_average_error, 708
- nppiAverageError_16s_C3R
 - image_average_error, 708
- nppiAverageError_16s_C4R
 - image_average_error, 709
- nppiAverageError_16sc_C1R
 - image_average_error, 709
- nppiAverageError_16sc_C2R
 - image_average_error, 709
- nppiAverageError_16sc_C3R
 - image_average_error, 710
- nppiAverageError_16sc_C4R
 - image_average_error, 710
- nppiAverageError_16u_C1R
 - image_average_error, 711
- nppiAverageError_16u_C2R
 - image_average_error, 711
- nppiAverageError_16u_C3R
 - image_average_error, 712
- nppiAverageError_16u_C4R
 - image_average_error, 712
- nppiAverageError_32f_C1R
 - image_average_error, 712
- nppiAverageError_32f_C2R
 - image_average_error, 713
- nppiAverageError_32f_C3R
 - image_average_error, 713
- nppiAverageError_32f_C4R
 - image_average_error, 714
- nppiAverageError_32fc_C1R
 - image_average_error, 714
- nppiAverageError_32fc_C2R
 - image_average_error, 715
- nppiAverageError_32fc_C3R
 - image_average_error, 715
- nppiAverageError_32fc_C4R
 - image_average_error, 716
- nppiAverageError_32s_C1R
 - image_average_error, 716
- nppiAverageError_32s_C2R
 - image_average_error, 716
- nppiAverageError_32s_C3R
 - image_average_error, 717
- nppiAverageError_32s_C4R
 - image_average_error, 717
- nppiAverageError_32sc_C1R
 - image_average_error, 718
- nppiAverageError_32sc_C2R
 - image_average_error, 718
- nppiAverageError_32sc_C3R
 - image_average_error, 719
- nppiAverageError_32sc_C4R
 - image_average_error, 719
- nppiAverageError_32u_C1R
 - image_average_error, 719
- nppiAverageError_32u_C2R
 - image_average_error, 720
- nppiAverageError_32u_C3R
 - image_average_error, 720
- nppiAverageError_32u_C4R
 - image_average_error, 721
- nppiAverageError_64f_C1R
 - image_average_error, 721
- nppiAverageError_64f_C2R
 - image_average_error, 722
- nppiAverageError_64f_C3R
 - image_average_error, 722
- nppiAverageError_64f_C4R
 - image_average_error, 723
- nppiAverageError_8s_C1R
 - image_average_error, 723
- nppiAverageError_8s_C2R
 - image_average_error, 723
- nppiAverageError_8s_C3R
 - image_average_error, 724
- nppiAverageError_8s_C4R
 - image_average_error, 724
- nppiAverageError_8u_C1R
 - image_average_error, 725
- nppiAverageError_8u_C2R
 - image_average_error, 725
- nppiAverageError_8u_C3R
 - image_average_error, 726
- nppiAverageError_8u_C4R
 - image_average_error, 726
- nppiAverageErrorGetBufferSize_16s_C1R
 - image_statistics_functions, 66
- nppiAverageErrorGetBufferSize_16s_C2R
 - image_statistics_functions, 66
- nppiAverageErrorGetBufferSize_16s_C3R
 - image_statistics_functions, 66
- nppiAverageErrorGetBufferSize_16s_C4R
 - image_statistics_functions, 67
- nppiAverageErrorGetBufferSize_16sc_C1R
 - image_statistics_functions, 67
- nppiAverageErrorGetBufferSize_16sc_C2R
 - image_statistics_functions, 67

- image_average_relative_error, 761
- nppiAverageRelativeError_32fc_C1R
 - image_average_relative_error, 762
- nppiAverageRelativeError_32fc_C2R
 - image_average_relative_error, 762
- nppiAverageRelativeError_32fc_C3R
 - image_average_relative_error, 762
- nppiAverageRelativeError_32fc_C4R
 - image_average_relative_error, 763
- nppiAverageRelativeError_32s_C1R
 - image_average_relative_error, 763
- nppiAverageRelativeError_32s_C2R
 - image_average_relative_error, 764
- nppiAverageRelativeError_32s_C3R
 - image_average_relative_error, 764
- nppiAverageRelativeError_32s_C4R
 - image_average_relative_error, 765
- nppiAverageRelativeError_32sc_C1R
 - image_average_relative_error, 765
- nppiAverageRelativeError_32sc_C2R
 - image_average_relative_error, 766
- nppiAverageRelativeError_32sc_C3R
 - image_average_relative_error, 766
- nppiAverageRelativeError_32sc_C4R
 - image_average_relative_error, 767
- nppiAverageRelativeError_32u_C1R
 - image_average_relative_error, 767
- nppiAverageRelativeError_32u_C2R
 - image_average_relative_error, 767
- nppiAverageRelativeError_32u_C3R
 - image_average_relative_error, 768
- nppiAverageRelativeError_32u_C4R
 - image_average_relative_error, 768
- nppiAverageRelativeError_64f_C1R
 - image_average_relative_error, 769
- nppiAverageRelativeError_64f_C2R
 - image_average_relative_error, 769
- nppiAverageRelativeError_64f_C3R
 - image_average_relative_error, 770
- nppiAverageRelativeError_64f_C4R
 - image_average_relative_error, 770
- nppiAverageRelativeError_8s_C1R
 - image_average_relative_error, 771
- nppiAverageRelativeError_8s_C2R
 - image_average_relative_error, 771
- nppiAverageRelativeError_8s_C3R
 - image_average_relative_error, 772
- nppiAverageRelativeError_8s_C4R
 - image_average_relative_error, 772
- nppiAverageRelativeError_8u_C1R
 - image_average_relative_error, 772
- nppiAverageRelativeError_8u_C2R
 - image_average_relative_error, 773
- nppiAverageRelativeError_8u_C3R
 - image_average_relative_error, 773
- image_average_relative_error, 773
- nppiAverageRelativeError_8u_C4R
 - image_average_relative_error, 774
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C1R
 - image_statistics_functions, 78
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C2R
 - image_statistics_functions, 79
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C3R
 - image_statistics_functions, 79
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C4R
 - image_statistics_functions, 79
- nppiAverageRelativeErrorGetBufferHostSize_-16sc_C1R
 - image_statistics_functions, 79
- nppiAverageRelativeErrorGetBufferHostSize_-16sc_C2R
 - image_statistics_functions, 80
- nppiAverageRelativeErrorGetBufferHostSize_-16sc_C3R
 - image_statistics_functions, 80
- nppiAverageRelativeErrorGetBufferHostSize_-16sc_C4R
 - image_statistics_functions, 80
- nppiAverageRelativeErrorGetBufferHostSize_-16u_C1R
 - image_statistics_functions, 81
- nppiAverageRelativeErrorGetBufferHostSize_-16u_C2R
 - image_statistics_functions, 81
- nppiAverageRelativeErrorGetBufferHostSize_-16u_C3R
 - image_statistics_functions, 81
- nppiAverageRelativeErrorGetBufferHostSize_-16u_C4R
 - image_statistics_functions, 81
- nppiAverageRelativeErrorGetBufferHostSize_-32f_C1R
 - image_statistics_functions, 82
- nppiAverageRelativeErrorGetBufferHostSize_-32f_C2R
 - image_statistics_functions, 82
- nppiAverageRelativeErrorGetBufferHostSize_-32f_C3R
 - image_statistics_functions, 82
- nppiAverageRelativeErrorGetBufferHostSize_-32f_C4R
 - image_statistics_functions, 83
- nppiAverageRelativeErrorGetBufferHostSize_-32fc_C1R
 - image_statistics_functions, 83

- nppiAverageRelativeErrorGetBufferHostSize_-32fc_C2R
image_statistics_functions, 83
- nppiAverageRelativeErrorGetBufferHostSize_-32fc_C3R
image_statistics_functions, 83
- nppiAverageRelativeErrorGetBufferHostSize_-32fc_C4R
image_statistics_functions, 84
- nppiAverageRelativeErrorGetBufferHostSize_-32s_C1R
image_statistics_functions, 84
- nppiAverageRelativeErrorGetBufferHostSize_-32s_C2R
image_statistics_functions, 84
- nppiAverageRelativeErrorGetBufferHostSize_-32s_C3R
image_statistics_functions, 85
- nppiAverageRelativeErrorGetBufferHostSize_-32s_C4R
image_statistics_functions, 85
- nppiAverageRelativeErrorGetBufferHostSize_-32sc_C1R
image_statistics_functions, 85
- nppiAverageRelativeErrorGetBufferHostSize_-32sc_C2R
image_statistics_functions, 85
- nppiAverageRelativeErrorGetBufferHostSize_-32sc_C3R
image_statistics_functions, 86
- nppiAverageRelativeErrorGetBufferHostSize_-32sc_C4R
image_statistics_functions, 86
- nppiAverageRelativeErrorGetBufferHostSize_-32u_C1R
image_statistics_functions, 86
- nppiAverageRelativeErrorGetBufferHostSize_-32u_C2R
image_statistics_functions, 87
- nppiAverageRelativeErrorGetBufferHostSize_-32u_C3R
image_statistics_functions, 87
- nppiAverageRelativeErrorGetBufferHostSize_-32u_C4R
image_statistics_functions, 87
- nppiAverageRelativeErrorGetBufferHostSize_-64f_C1R
image_statistics_functions, 87
- nppiAverageRelativeErrorGetBufferHostSize_-64f_C2R
image_statistics_functions, 88
- nppiAverageRelativeErrorGetBufferHostSize_-64f_C3R
image_statistics_functions, 88
- nppiAverageRelativeErrorGetBufferHostSize_-64f_C4R
image_statistics_functions, 88
- nppiAverageRelativeErrorGetBufferHostSize_8s_C1R
image_statistics_functions, 89
- nppiAverageRelativeErrorGetBufferHostSize_8s_C2R
image_statistics_functions, 89
- nppiAverageRelativeErrorGetBufferHostSize_8s_C3R
image_statistics_functions, 89
- nppiAverageRelativeErrorGetBufferHostSize_8s_C4R
image_statistics_functions, 89
- nppiAverageRelativeErrorGetBufferHostSize_8u_C1R
image_statistics_functions, 90
- nppiAverageRelativeErrorGetBufferHostSize_8u_C2R
image_statistics_functions, 90
- nppiAverageRelativeErrorGetBufferHostSize_8u_C3R
image_statistics_functions, 90
- nppiAverageRelativeErrorGetBufferHostSize_8u_C4R
image_statistics_functions, 91
- NppiAxis
typedefs_npp, 40
- NppiBayerGridPosition
typedefs_npp, 40
- NppiBorderType
typedefs_npp, 40
- nppiCountInRange_32f_AC4R
image_count_in_range, 484
- nppiCountInRange_32f_C1R
image_count_in_range, 484
- nppiCountInRange_32f_C3R
image_count_in_range, 485
- nppiCountInRange_8u_AC4R
image_count_in_range, 485
- nppiCountInRange_8u_C1R
image_count_in_range, 486
- nppiCountInRange_8u_C3R
image_count_in_range, 486
- nppiCountInRangeGetBufferHostSize_32f_AC4R
image_count_in_range, 487
- nppiCountInRangeGetBufferHostSize_32f_C1R
image_count_in_range, 487
- nppiCountInRangeGetBufferHostSize_32f_C3R
image_count_in_range, 487
- nppiCountInRangeGetBufferHostSize_8u_AC4R
image_count_in_range, 487
- nppiCountInRangeGetBufferHostSize_8u_C1R

- image_count_in_range, 488
- nppiCountInRangeGetBufferHostSize_8u_C3R
 - image_count_in_range, 488
- nppiCrossCorrFull_Norm_16u32f_AC4R
 - crosscorrfullnorm, 578
- nppiCrossCorrFull_Norm_16u32f_C1R
 - crosscorrfullnorm, 578
- nppiCrossCorrFull_Norm_16u32f_C3R
 - crosscorrfullnorm, 578
- nppiCrossCorrFull_Norm_16u32f_C4R
 - crosscorrfullnorm, 579
- nppiCrossCorrFull_Norm_32f_AC4R
 - crosscorrfullnorm, 579
- nppiCrossCorrFull_Norm_32f_C1R
 - crosscorrfullnorm, 580
- nppiCrossCorrFull_Norm_32f_C3R
 - crosscorrfullnorm, 580
- nppiCrossCorrFull_Norm_32f_C4R
 - crosscorrfullnorm, 581
- nppiCrossCorrFull_Norm_8s32f_AC4R
 - crosscorrfullnorm, 581
- nppiCrossCorrFull_Norm_8s32f_C1R
 - crosscorrfullnorm, 581
- nppiCrossCorrFull_Norm_8s32f_C3R
 - crosscorrfullnorm, 582
- nppiCrossCorrFull_Norm_8s32f_C4R
 - crosscorrfullnorm, 582
- nppiCrossCorrFull_Norm_8u32f_AC4R
 - crosscorrfullnorm, 583
- nppiCrossCorrFull_Norm_8u32f_C1R
 - crosscorrfullnorm, 583
- nppiCrossCorrFull_Norm_8u32f_C3R
 - crosscorrfullnorm, 584
- nppiCrossCorrFull_Norm_8u32f_C4R
 - crosscorrfullnorm, 584
- nppiCrossCorrFull_Norm_8u_AC4RSfs
 - crosscorrfullnorm, 584
- nppiCrossCorrFull_Norm_8u_C1RSfs
 - crosscorrfullnorm, 585
- nppiCrossCorrFull_Norm_8u_C3RSfs
 - crosscorrfullnorm, 585
- nppiCrossCorrFull_Norm_8u_C4RSfs
 - crosscorrfullnorm, 586
- nppiCrossCorrFull_NormLevel_16u32f_AC4R
 - crosscorrfullnormlevel, 616
- nppiCrossCorrFull_NormLevel_16u32f_C1R
 - crosscorrfullnormlevel, 616
- nppiCrossCorrFull_NormLevel_16u32f_C3R
 - crosscorrfullnormlevel, 616
- nppiCrossCorrFull_NormLevel_16u32f_C4R
 - crosscorrfullnormlevel, 617
- nppiCrossCorrFull_NormLevel_32f_AC4R
 - crosscorrfullnormlevel, 617
- nppiCrossCorrFull_NormLevel_32f_C1R
 - crosscorrfullnormlevel, 618
- nppiCrossCorrFull_NormLevel_32f_C3R
 - crosscorrfullnormlevel, 618
- nppiCrossCorrFull_NormLevel_32f_C4R
 - crosscorrfullnormlevel, 619
- nppiCrossCorrFull_NormLevel_8s32f_AC4R
 - crosscorrfullnormlevel, 619
- nppiCrossCorrFull_NormLevel_8s32f_C1R
 - crosscorrfullnormlevel, 620
- nppiCrossCorrFull_NormLevel_8s32f_C3R
 - crosscorrfullnormlevel, 620
- nppiCrossCorrFull_NormLevel_8s32f_C4R
 - crosscorrfullnormlevel, 621
- nppiCrossCorrFull_NormLevel_8u32f_AC4R
 - crosscorrfullnormlevel, 621
- nppiCrossCorrFull_NormLevel_8u32f_C1R
 - crosscorrfullnormlevel, 622
- nppiCrossCorrFull_NormLevel_8u32f_C3R
 - crosscorrfullnormlevel, 622
- nppiCrossCorrFull_NormLevel_8u32f_C4R
 - crosscorrfullnormlevel, 623
- nppiCrossCorrFull_NormLevel_8u_AC4RSfs
 - crosscorrfullnormlevel, 623
- nppiCrossCorrFull_NormLevel_8u_C1RSfs
 - crosscorrfullnormlevel, 624
- nppiCrossCorrFull_NormLevel_8u_C3RSfs
 - crosscorrfullnormlevel, 624
- nppiCrossCorrFull_NormLevel_8u_C4RSfs
 - crosscorrfullnormlevel, 625
- nppiCrossCorrSame_Norm_16u32f_AC4R
 - crosscorrmenorm, 589
- nppiCrossCorrSame_Norm_16u32f_C1R
 - crosscorrmenorm, 589
- nppiCrossCorrSame_Norm_16u32f_C3R
 - crosscorrmenorm, 589
- nppiCrossCorrSame_Norm_16u32f_C4R
 - crosscorrmenorm, 590
- nppiCrossCorrSame_Norm_32f_AC4R
 - crosscorrmenorm, 590
- nppiCrossCorrSame_Norm_32f_C1R
 - crosscorrmenorm, 591
- nppiCrossCorrSame_Norm_32f_C3R
 - crosscorrmenorm, 591
- nppiCrossCorrSame_Norm_32f_C4R
 - crosscorrmenorm, 592
- nppiCrossCorrSame_Norm_8s32f_AC4R
 - crosscorrmenorm, 592
- nppiCrossCorrSame_Norm_8s32f_C1R
 - crosscorrmenorm, 592
- nppiCrossCorrSame_Norm_8s32f_C3R
 - crosscorrmenorm, 593
- nppiCrossCorrSame_Norm_8s32f_C4R
 - crosscorrmenorm, 593
- nppiCrossCorrSame_Norm_8u32f_AC4R
 - crosscorrmenorm, 593

- crosscorrssamenorm, [594](#)
- nppiCrossCorrSame_Norm_8u32f_C1R
 - crosscorrssamenorm, [594](#)
- nppiCrossCorrSame_Norm_8u32f_C3R
 - crosscorrssamenorm, [595](#)
- nppiCrossCorrSame_Norm_8u32f_C4R
 - crosscorrssamenorm, [595](#)
- nppiCrossCorrSame_Norm_8u_AC4RSfs
 - crosscorrssamenorm, [595](#)
- nppiCrossCorrSame_Norm_8u_C1RSfs
 - crosscorrssamenorm, [596](#)
- nppiCrossCorrSame_Norm_8u_C3RSfs
 - crosscorrssamenorm, [596](#)
- nppiCrossCorrSame_Norm_8u_C4RSfs
 - crosscorrssamenorm, [597](#)
- nppiCrossCorrSame_NormLevel_16u32f_AC4R
 - crosscorrssamenormlevel, [636](#)
- nppiCrossCorrSame_NormLevel_16u32f_C1R
 - crosscorrssamenormlevel, [636](#)
- nppiCrossCorrSame_NormLevel_16u32f_C3R
 - crosscorrssamenormlevel, [636](#)
- nppiCrossCorrSame_NormLevel_16u32f_C4R
 - crosscorrssamenormlevel, [637](#)
- nppiCrossCorrSame_NormLevel_32f_AC4R
 - crosscorrssamenormlevel, [637](#)
- nppiCrossCorrSame_NormLevel_32f_C1R
 - crosscorrssamenormlevel, [638](#)
- nppiCrossCorrSame_NormLevel_32f_C3R
 - crosscorrssamenormlevel, [638](#)
- nppiCrossCorrSame_NormLevel_32f_C4R
 - crosscorrssamenormlevel, [639](#)
- nppiCrossCorrSame_NormLevel_8s32f_AC4R
 - crosscorrssamenormlevel, [639](#)
- nppiCrossCorrSame_NormLevel_8s32f_C1R
 - crosscorrssamenormlevel, [640](#)
- nppiCrossCorrSame_NormLevel_8s32f_C3R
 - crosscorrssamenormlevel, [640](#)
- nppiCrossCorrSame_NormLevel_8s32f_C4R
 - crosscorrssamenormlevel, [641](#)
- nppiCrossCorrSame_NormLevel_8u32f_AC4R
 - crosscorrssamenormlevel, [641](#)
- nppiCrossCorrSame_NormLevel_8u32f_C1R
 - crosscorrssamenormlevel, [642](#)
- nppiCrossCorrSame_NormLevel_8u32f_C3R
 - crosscorrssamenormlevel, [642](#)
- nppiCrossCorrSame_NormLevel_8u32f_C4R
 - crosscorrssamenormlevel, [643](#)
- nppiCrossCorrSame_NormLevel_8u_AC4RSfs
 - crosscorrssamenormlevel, [643](#)
- nppiCrossCorrSame_NormLevel_8u_C1RSfs
 - crosscorrssamenormlevel, [644](#)
- nppiCrossCorrSame_NormLevel_8u_C3RSfs
 - crosscorrssamenormlevel, [644](#)
- nppiCrossCorrSame_NormLevel_8u_C4RSfs
 - crosscorrssamenormlevel, [645](#)
- nppiCrossCorrValid_16u32f_C1R
 - crosscorrvalid, [609](#)
- nppiCrossCorrValid_32f_C1R
 - crosscorrvalid, [610](#)
- nppiCrossCorrValid_8s32f_C1R
 - crosscorrvalid, [610](#)
- nppiCrossCorrValid_8u32f_C1R
 - crosscorrvalid, [610](#)
- nppiCrossCorrValid_Norm_16u32f_AC4R
 - crosscorrvalidnorm, [600](#)
- nppiCrossCorrValid_Norm_16u32f_C1R
 - crosscorrvalidnorm, [600](#)
- nppiCrossCorrValid_Norm_16u32f_C3R
 - crosscorrvalidnorm, [600](#)
- nppiCrossCorrValid_Norm_16u32f_C4R
 - crosscorrvalidnorm, [601](#)
- nppiCrossCorrValid_Norm_32f_AC4R
 - crosscorrvalidnorm, [601](#)
- nppiCrossCorrValid_Norm_32f_C1R
 - crosscorrvalidnorm, [602](#)
- nppiCrossCorrValid_Norm_32f_C3R
 - crosscorrvalidnorm, [602](#)
- nppiCrossCorrValid_Norm_32f_C4R
 - crosscorrvalidnorm, [603](#)
- nppiCrossCorrValid_Norm_8s32f_AC4R
 - crosscorrvalidnorm, [603](#)
- nppiCrossCorrValid_Norm_8s32f_C1R
 - crosscorrvalidnorm, [603](#)
- nppiCrossCorrValid_Norm_8s32f_C3R
 - crosscorrvalidnorm, [604](#)
- nppiCrossCorrValid_Norm_8s32f_C4R
 - crosscorrvalidnorm, [604](#)
- nppiCrossCorrValid_Norm_8u32f_AC4R
 - crosscorrvalidnorm, [605](#)
- nppiCrossCorrValid_Norm_8u32f_C1R
 - crosscorrvalidnorm, [605](#)
- nppiCrossCorrValid_Norm_8u32f_C3R
 - crosscorrvalidnorm, [606](#)
- nppiCrossCorrValid_Norm_8u32f_C4R
 - crosscorrvalidnorm, [606](#)
- nppiCrossCorrValid_Norm_8u_AC4RSfs
 - crosscorrvalidnorm, [606](#)
- nppiCrossCorrValid_Norm_8u_C1RSfs
 - crosscorrvalidnorm, [607](#)
- nppiCrossCorrValid_Norm_8u_C3RSfs
 - crosscorrvalidnorm, [607](#)
- nppiCrossCorrValid_Norm_8u_C4RSfs
 - crosscorrvalidnorm, [608](#)
- nppiCrossCorrValid_NormLevel_16u32f_AC4R
 - crosscorrvalidnormlevel, [656](#)
- nppiCrossCorrValid_NormLevel_16u32f_C1R
 - crosscorrvalidnormlevel, [656](#)
- nppiCrossCorrValid_NormLevel_16u32f_C3R

- crosscorrvalidnormlevel, [656](#)
- nppiCrossCorrValid_NormLevel_16u32f_C4R
 - crosscorrvalidnormlevel, [657](#)
- nppiCrossCorrValid_NormLevel_32f_AC4R
 - crosscorrvalidnormlevel, [657](#)
- nppiCrossCorrValid_NormLevel_32f_C1R
 - crosscorrvalidnormlevel, [658](#)
- nppiCrossCorrValid_NormLevel_32f_C3R
 - crosscorrvalidnormlevel, [658](#)
- nppiCrossCorrValid_NormLevel_32f_C4R
 - crosscorrvalidnormlevel, [659](#)
- nppiCrossCorrValid_NormLevel_8s32f_AC4R
 - crosscorrvalidnormlevel, [659](#)
- nppiCrossCorrValid_NormLevel_8s32f_C1R
 - crosscorrvalidnormlevel, [660](#)
- nppiCrossCorrValid_NormLevel_8s32f_C3R
 - crosscorrvalidnormlevel, [660](#)
- nppiCrossCorrValid_NormLevel_8s32f_C4R
 - crosscorrvalidnormlevel, [661](#)
- nppiCrossCorrValid_NormLevel_8u32f_AC4R
 - crosscorrvalidnormlevel, [661](#)
- nppiCrossCorrValid_NormLevel_8u32f_C1R
 - crosscorrvalidnormlevel, [662](#)
- nppiCrossCorrValid_NormLevel_8u32f_C3R
 - crosscorrvalidnormlevel, [662](#)
- nppiCrossCorrValid_NormLevel_8u32f_C4R
 - crosscorrvalidnormlevel, [663](#)
- nppiCrossCorrValid_NormLevel_8u_AC4RSfs
 - crosscorrvalidnormlevel, [663](#)
- nppiCrossCorrValid_NormLevel_8u_C1RSfs
 - crosscorrvalidnormlevel, [664](#)
- nppiCrossCorrValid_NormLevel_8u_C3RSfs
 - crosscorrvalidnormlevel, [664](#)
- nppiCrossCorrValid_NormLevel_8u_C4RSfs
 - crosscorrvalidnormlevel, [665](#)
- nppiDCTable
 - typedefs_npp, [41](#)
- NppiDifferentialKernel
 - typedefs_npp, [40](#)
- nppiDotProd_16s64f_AC4R
 - image_dot_prod, [462](#)
- nppiDotProd_16s64f_C1R
 - image_dot_prod, [462](#)
- nppiDotProd_16s64f_C3R
 - image_dot_prod, [463](#)
- nppiDotProd_16s64f_C4R
 - image_dot_prod, [463](#)
- nppiDotProd_16u64f_AC4R
 - image_dot_prod, [464](#)
- nppiDotProd_16u64f_C1R
 - image_dot_prod, [464](#)
- nppiDotProd_16u64f_C3R
 - image_dot_prod, [465](#)
- nppiDotProd_16u64f_C4R
 - image_dot_prod, [465](#)
- nppiDotProd_32f64f_AC4R
 - image_dot_prod, [465](#)
- nppiDotProd_32f64f_C1R
 - image_dot_prod, [466](#)
- nppiDotProd_32f64f_C3R
 - image_dot_prod, [466](#)
- nppiDotProd_32f64f_C4R
 - image_dot_prod, [467](#)
- nppiDotProd_32s64f_AC4R
 - image_dot_prod, [467](#)
- nppiDotProd_32s64f_C1R
 - image_dot_prod, [468](#)
- nppiDotProd_32s64f_C3R
 - image_dot_prod, [468](#)
- nppiDotProd_32s64f_C4R
 - image_dot_prod, [468](#)
- nppiDotProd_32u64f_AC4R
 - image_dot_prod, [469](#)
- nppiDotProd_32u64f_C1R
 - image_dot_prod, [469](#)
- nppiDotProd_32u64f_C3R
 - image_dot_prod, [470](#)
- nppiDotProd_32u64f_C4R
 - image_dot_prod, [470](#)
- nppiDotProd_8s64f_AC4R
 - image_dot_prod, [471](#)
- nppiDotProd_8s64f_C1R
 - image_dot_prod, [471](#)
- nppiDotProd_8s64f_C3R
 - image_dot_prod, [471](#)
- nppiDotProd_8s64f_C4R
 - image_dot_prod, [472](#)
- nppiDotProd_8u64f_AC4R
 - image_dot_prod, [472](#)
- nppiDotProd_8u64f_C1R
 - image_dot_prod, [473](#)
- nppiDotProd_8u64f_C3R
 - image_dot_prod, [473](#)
- nppiDotProd_8u64f_C4R
 - image_dot_prod, [473](#)
- nppiDotProdGetBufferHostSize_16s64f_AC4R
 - image_dot_prod, [474](#)
- nppiDotProdGetBufferHostSize_16s64f_C1R
 - image_dot_prod, [474](#)
- nppiDotProdGetBufferHostSize_16s64f_C3R
 - image_dot_prod, [474](#)
- nppiDotProdGetBufferHostSize_16s64f_C4R
 - image_dot_prod, [475](#)
- nppiDotProdGetBufferHostSize_16u64f_AC4R
 - image_dot_prod, [475](#)
- nppiDotProdGetBufferHostSize_16u64f_C1R
 - image_dot_prod, [475](#)
- nppiDotProdGetBufferHostSize_16u64f_C3R
 - image_dot_prod, [475](#)

- image_dot_prod, [476](#)
- nppiDotProdGetBufferHostSize_16u64f_C4R
 - image_dot_prod, [476](#)
- nppiDotProdGetBufferHostSize_32f64f_AC4R
 - image_dot_prod, [476](#)
- nppiDotProdGetBufferHostSize_32f64f_C1R
 - image_dot_prod, [476](#)
- nppiDotProdGetBufferHostSize_32f64f_C3R
 - image_dot_prod, [477](#)
- nppiDotProdGetBufferHostSize_32f64f_C4R
 - image_dot_prod, [477](#)
- nppiDotProdGetBufferHostSize_32s64f_AC4R
 - image_dot_prod, [477](#)
- nppiDotProdGetBufferHostSize_32s64f_C1R
 - image_dot_prod, [478](#)
- nppiDotProdGetBufferHostSize_32s64f_C3R
 - image_dot_prod, [478](#)
- nppiDotProdGetBufferHostSize_32s64f_C4R
 - image_dot_prod, [478](#)
- nppiDotProdGetBufferHostSize_32u64f_AC4R
 - image_dot_prod, [478](#)
- nppiDotProdGetBufferHostSize_32u64f_C1R
 - image_dot_prod, [479](#)
- nppiDotProdGetBufferHostSize_32u64f_C3R
 - image_dot_prod, [479](#)
- nppiDotProdGetBufferHostSize_32u64f_C4R
 - image_dot_prod, [479](#)
- nppiDotProdGetBufferHostSize_8s64f_AC4R
 - image_dot_prod, [480](#)
- nppiDotProdGetBufferHostSize_8s64f_C1R
 - image_dot_prod, [480](#)
- nppiDotProdGetBufferHostSize_8s64f_C3R
 - image_dot_prod, [480](#)
- nppiDotProdGetBufferHostSize_8s64f_C4R
 - image_dot_prod, [480](#)
- nppiDotProdGetBufferHostSize_8u64f_AC4R
 - image_dot_prod, [481](#)
- nppiDotProdGetBufferHostSize_8u64f_C1R
 - image_dot_prod, [481](#)
- nppiDotProdGetBufferHostSize_8u64f_C3R
 - image_dot_prod, [481](#)
- nppiDotProdGetBufferHostSize_8u64f_C4R
 - image_dot_prod, [482](#)
- nppiEvenLevelsHost_32s
 - image_histogrameven, [513](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-AC4R
 - crosscorrfullnormlevel, [625](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-C1R
 - crosscorrfullnormlevel, [626](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-C3R
 - crosscorrfullnormlevel, [626](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-C4R
 - crosscorrfullnormlevel, [626](#)
- nppiFullNormLevelGetBufferHostSize_32f_AC4R
 - crosscorrfullnormlevel, [627](#)
- nppiFullNormLevelGetBufferHostSize_32f_C1R
 - crosscorrfullnormlevel, [627](#)
- nppiFullNormLevelGetBufferHostSize_32f_C3R
 - crosscorrfullnormlevel, [627](#)
- nppiFullNormLevelGetBufferHostSize_32f_C4R
 - crosscorrfullnormlevel, [627](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_-AC4R
 - crosscorrfullnormlevel, [628](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C1R
 - crosscorrfullnormlevel, [628](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C3R
 - crosscorrfullnormlevel, [628](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C4R
 - crosscorrfullnormlevel, [629](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-AC4R
 - crosscorrfullnormlevel, [629](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-C1R
 - crosscorrfullnormlevel, [629](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-C3R
 - crosscorrfullnormlevel, [629](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-C4R
 - crosscorrfullnormlevel, [630](#)
- nppiFullNormLevelGetBufferHostSize_8u_-AC4RSfs
 - crosscorrfullnormlevel, [630](#)
- nppiFullNormLevelGetBufferHostSize_8u_C1RSfs
 - crosscorrfullnormlevel, [630](#)
- nppiFullNormLevelGetBufferHostSize_8u_C3RSfs
 - crosscorrfullnormlevel, [631](#)
- nppiFullNormLevelGetBufferHostSize_8u_C4RSfs
 - crosscorrfullnormlevel, [631](#)
- NppiHaarBuffer, [783](#)
 - haarBuffer, [783](#)
 - haarBufferSize, [783](#)
- NppiHaarClassifier_32f, [784](#)
 - classifiers, [784](#)
 - classifierSize, [784](#)
 - classifierStep, [784](#)
 - counterDevice, [784](#)
 - numClassifiers, [784](#)
- nppiHistogramEven_16s_AC4R
 - image_histogrameven, [514](#)
- nppiHistogramEven_16s_C1R
 - image_histogrameven, [514](#)

- nppiHistogramEven_16s_C3R
image_histogrameven, [514](#)
- nppiHistogramEven_16s_C4R
image_histogrameven, [515](#)
- nppiHistogramEven_16u_AC4R
image_histogrameven, [515](#)
- nppiHistogramEven_16u_C1R
image_histogrameven, [516](#)
- nppiHistogramEven_16u_C3R
image_histogrameven, [516](#)
- nppiHistogramEven_16u_C4R
image_histogrameven, [517](#)
- nppiHistogramEven_8u_AC4R
image_histogrameven, [517](#)
- nppiHistogramEven_8u_C1R
image_histogrameven, [518](#)
- nppiHistogramEven_8u_C3R
image_histogrameven, [518](#)
- nppiHistogramEven_8u_C4R
image_histogrameven, [519](#)
- nppiHistogramEvenGetBufferSize_16s_AC4R
image_histogrameven, [519](#)
- nppiHistogramEvenGetBufferSize_16s_C1R
image_histogrameven, [519](#)
- nppiHistogramEvenGetBufferSize_16s_C3R
image_histogrameven, [520](#)
- nppiHistogramEvenGetBufferSize_16s_C4R
image_histogrameven, [520](#)
- nppiHistogramEvenGetBufferSize_16u_AC4R
image_histogrameven, [520](#)
- nppiHistogramEvenGetBufferSize_16u_C1R
image_histogrameven, [521](#)
- nppiHistogramEvenGetBufferSize_16u_C3R
image_histogrameven, [521](#)
- nppiHistogramEvenGetBufferSize_16u_C4R
image_histogrameven, [521](#)
- nppiHistogramEvenGetBufferSize_8u_AC4R
image_histogrameven, [522](#)
- nppiHistogramEvenGetBufferSize_8u_C1R
image_histogrameven, [522](#)
- nppiHistogramEvenGetBufferSize_8u_C3R
image_histogrameven, [522](#)
- nppiHistogramEvenGetBufferSize_8u_C4R
image_histogrameven, [523](#)
- nppiHistogramRange_16s_AC4R
image_histogramrange, [527](#)
- nppiHistogramRange_16s_C1R
image_histogramrange, [527](#)
- nppiHistogramRange_16s_C3R
image_histogramrange, [527](#)
- nppiHistogramRange_16s_C4R
image_histogramrange, [528](#)
- nppiHistogramRange_16u_AC4R
image_histogramrange, [528](#)
- nppiHistogramRange_16u_C1R
image_histogramrange, [529](#)
- nppiHistogramRange_16u_C3R
image_histogramrange, [529](#)
- nppiHistogramRange_16u_C4R
image_histogramrange, [529](#)
- nppiHistogramRange_32f_AC4R
image_histogramrange, [530](#)
- nppiHistogramRange_32f_C1R
image_histogramrange, [530](#)
- nppiHistogramRange_32f_C3R
image_histogramrange, [531](#)
- nppiHistogramRange_32f_C4R
image_histogramrange, [531](#)
- nppiHistogramRange_8u_AC4R
image_histogramrange, [532](#)
- nppiHistogramRange_8u_C1R
image_histogramrange, [532](#)
- nppiHistogramRange_8u_C3R
image_histogramrange, [533](#)
- nppiHistogramRange_8u_C4R
image_histogramrange, [533](#)
- nppiHistogramRangeGetBufferSize_16s_AC4R
image_histogramrange, [533](#)
- nppiHistogramRangeGetBufferSize_16s_C1R
image_histogramrange, [534](#)
- nppiHistogramRangeGetBufferSize_16s_C3R
image_histogramrange, [534](#)
- nppiHistogramRangeGetBufferSize_16s_C4R
image_histogramrange, [534](#)
- nppiHistogramRangeGetBufferSize_16u_AC4R
image_histogramrange, [535](#)
- nppiHistogramRangeGetBufferSize_16u_C1R
image_histogramrange, [535](#)
- nppiHistogramRangeGetBufferSize_16u_C3R
image_histogramrange, [535](#)
- nppiHistogramRangeGetBufferSize_16u_C4R
image_histogramrange, [536](#)
- nppiHistogramRangeGetBufferSize_32f_AC4R
image_histogramrange, [536](#)
- nppiHistogramRangeGetBufferSize_32f_C1R
image_histogramrange, [536](#)
- nppiHistogramRangeGetBufferSize_32f_C3R
image_histogramrange, [537](#)
- nppiHistogramRangeGetBufferSize_32f_C4R
image_histogramrange, [537](#)
- nppiHistogramRangeGetBufferSize_8u_AC4R
image_histogramrange, [537](#)
- nppiHistogramRangeGetBufferSize_8u_C1R
image_histogramrange, [538](#)
- nppiHistogramRangeGetBufferSize_8u_C3R
image_histogramrange, [538](#)
- nppiHistogramRangeGetBufferSize_8u_C4R
image_histogramrange, [538](#)

- NppiHuffmanTableType
 - typedefs_npp, 40
- nppiIntegral_8u32f_C1R
 - image_integral, 503
- nppiIntegral_8u32s_C1R
 - image_integral, 503
- NppiInterpolationMode
 - typedefs_npp, 41
- nppiMagnitude_32fc32f_C1R
 - image_fourier_transforms, 776
- nppiMagnitudeSqr_32fc32f_C1R
 - image_fourier_transforms, 776
- NppiMaskSize
 - typedefs_npp, 41
- nppiMax_16s_AC4R
 - image_max, 161
- nppiMax_16s_C1R
 - image_max, 161
- nppiMax_16s_C3R
 - image_max, 162
- nppiMax_16s_C4R
 - image_max, 162
- nppiMax_16u_AC4R
 - image_max, 162
- nppiMax_16u_C1R
 - image_max, 163
- nppiMax_16u_C3R
 - image_max, 163
- nppiMax_16u_C4R
 - image_max, 164
- nppiMax_32f_AC4R
 - image_max, 164
- nppiMax_32f_C1R
 - image_max, 164
- nppiMax_32f_C3R
 - image_max, 165
- nppiMax_32f_C4R
 - image_max, 165
- nppiMax_8u_AC4R
 - image_max, 165
- nppiMax_8u_C1R
 - image_max, 166
- nppiMax_8u_C3R
 - image_max, 166
- nppiMax_8u_C4R
 - image_max, 167
- nppiMaxEvery_16s_AC4IR
 - image_maxevery, 490
- nppiMaxEvery_16s_C1IR
 - image_maxevery, 490
- nppiMaxEvery_16s_C3IR
 - image_maxevery, 491
- nppiMaxEvery_16s_C4IR
 - image_maxevery, 491
- nppiMaxEvery_16u_AC4IR
 - image_maxevery, 491
- nppiMaxEvery_16u_C1IR
 - image_maxevery, 492
- nppiMaxEvery_16u_C3IR
 - image_maxevery, 492
- nppiMaxEvery_16u_C4IR
 - image_maxevery, 492
- nppiMaxEvery_32f_AC4IR
 - image_maxevery, 493
- nppiMaxEvery_32f_C1IR
 - image_maxevery, 493
- nppiMaxEvery_32f_C3IR
 - image_maxevery, 493
- nppiMaxEvery_32f_C4IR
 - image_maxevery, 494
- nppiMaxEvery_8u_AC4IR
 - image_maxevery, 494
- nppiMaxEvery_8u_C1IR
 - image_maxevery, 494
- nppiMaxEvery_8u_C3IR
 - image_maxevery, 495
- nppiMaxEvery_8u_C4IR
 - image_maxevery, 495
- nppiMaxGetBufferHostSize_16s_AC4R
 - image_max, 167
- nppiMaxGetBufferHostSize_16s_C1R
 - image_max, 167
- nppiMaxGetBufferHostSize_16s_C3R
 - image_max, 167
- nppiMaxGetBufferHostSize_16s_C4R
 - image_max, 168
- nppiMaxGetBufferHostSize_16u_AC4R
 - image_max, 168
- nppiMaxGetBufferHostSize_16u_C1R
 - image_max, 168
- nppiMaxGetBufferHostSize_16u_C3R
 - image_max, 169
- nppiMaxGetBufferHostSize_16u_C4R
 - image_max, 169
- nppiMaxGetBufferHostSize_32f_AC4R
 - image_max, 169
- nppiMaxGetBufferHostSize_32f_C1R
 - image_max, 169
- nppiMaxGetBufferHostSize_32f_C3R
 - image_max, 170
- nppiMaxGetBufferHostSize_32f_C4R
 - image_max, 170
- nppiMaxGetBufferHostSize_8u_AC4R
 - image_max, 170
- nppiMaxGetBufferHostSize_8u_C1R
 - image_max, 170
- nppiMaxGetBufferHostSize_8u_C3R
 - image_max, 171

- nppiMaxGetBufferHostSize_8u_C4R
image_max, 171
- nppiMaximumError_16s_C1R
image_maximum_error, 684
- nppiMaximumError_16s_C2R
image_maximum_error, 685
- nppiMaximumError_16s_C3R
image_maximum_error, 685
- nppiMaximumError_16s_C4R
image_maximum_error, 685
- nppiMaximumError_16sc_C1R
image_maximum_error, 686
- nppiMaximumError_16sc_C2R
image_maximum_error, 686
- nppiMaximumError_16sc_C3R
image_maximum_error, 687
- nppiMaximumError_16sc_C4R
image_maximum_error, 687
- nppiMaximumError_16u_C1R
image_maximum_error, 688
- nppiMaximumError_16u_C2R
image_maximum_error, 688
- nppiMaximumError_16u_C3R
image_maximum_error, 688
- nppiMaximumError_16u_C4R
image_maximum_error, 689
- nppiMaximumError_32f_C1R
image_maximum_error, 689
- nppiMaximumError_32f_C2R
image_maximum_error, 690
- nppiMaximumError_32f_C3R
image_maximum_error, 690
- nppiMaximumError_32f_C4R
image_maximum_error, 691
- nppiMaximumError_32fc_C1R
image_maximum_error, 691
- nppiMaximumError_32fc_C2R
image_maximum_error, 692
- nppiMaximumError_32fc_C3R
image_maximum_error, 692
- nppiMaximumError_32fc_C4R
image_maximum_error, 692
- nppiMaximumError_32s_C1R
image_maximum_error, 693
- nppiMaximumError_32s_C2R
image_maximum_error, 693
- nppiMaximumError_32s_C3R
image_maximum_error, 694
- nppiMaximumError_32s_C4R
image_maximum_error, 694
- nppiMaximumError_32sc_C1R
image_maximum_error, 695
- nppiMaximumError_32sc_C2R
image_maximum_error, 695
- nppiMaximumError_32sc_C3R
image_maximum_error, 695
- nppiMaximumError_32sc_C4R
image_maximum_error, 696
- nppiMaximumError_32u_C1R
image_maximum_error, 696
- nppiMaximumError_32u_C2R
image_maximum_error, 697
- nppiMaximumError_32u_C3R
image_maximum_error, 697
- nppiMaximumError_32u_C4R
image_maximum_error, 698
- nppiMaximumError_64f_C1R
image_maximum_error, 698
- nppiMaximumError_64f_C2R
image_maximum_error, 698
- nppiMaximumError_64f_C3R
image_maximum_error, 699
- nppiMaximumError_64f_C4R
image_maximum_error, 699
- nppiMaximumError_8s_C1R
image_maximum_error, 700
- nppiMaximumError_8s_C2R
image_maximum_error, 700
- nppiMaximumError_8s_C3R
image_maximum_error, 701
- nppiMaximumError_8s_C4R
image_maximum_error, 701
- nppiMaximumError_8u_C1R
image_maximum_error, 701
- nppiMaximumError_8u_C2R
image_maximum_error, 702
- nppiMaximumError_8u_C3R
image_maximum_error, 702
- nppiMaximumError_8u_C4R
image_maximum_error, 703
- nppiMaximumErrorGetBufferHostSize_16s_C1R
image_statistics_functions, 91
- nppiMaximumErrorGetBufferHostSize_16s_C2R
image_statistics_functions, 91
- nppiMaximumErrorGetBufferHostSize_16s_C3R
image_statistics_functions, 91
- nppiMaximumErrorGetBufferHostSize_16s_C4R
image_statistics_functions, 92
- nppiMaximumErrorGetBufferHostSize_16sc_C1R
image_statistics_functions, 92
- nppiMaximumErrorGetBufferHostSize_16sc_C2R
image_statistics_functions, 92
- nppiMaximumErrorGetBufferHostSize_16sc_C3R
image_statistics_functions, 93
- nppiMaximumErrorGetBufferHostSize_16sc_C4R
image_statistics_functions, 93
- nppiMaximumErrorGetBufferHostSize_16u_C1R
image_statistics_functions, 93

- [nppiMaximumErrorGetBufferHostSize_16u_C2R](#)
[image_statistics_functions](#), 93
- [nppiMaximumErrorGetBufferHostSize_16u_C3R](#)
[image_statistics_functions](#), 94
- [nppiMaximumErrorGetBufferHostSize_16u_C4R](#)
[image_statistics_functions](#), 94
- [nppiMaximumErrorGetBufferHostSize_32f_C1R](#)
[image_statistics_functions](#), 94
- [nppiMaximumErrorGetBufferHostSize_32f_C2R](#)
[image_statistics_functions](#), 95
- [nppiMaximumErrorGetBufferHostSize_32f_C3R](#)
[image_statistics_functions](#), 95
- [nppiMaximumErrorGetBufferHostSize_32f_C4R](#)
[image_statistics_functions](#), 95
- [nppiMaximumErrorGetBufferHostSize_32fc_C1R](#)
[image_statistics_functions](#), 95
- [nppiMaximumErrorGetBufferHostSize_32fc_C2R](#)
[image_statistics_functions](#), 96
- [nppiMaximumErrorGetBufferHostSize_32fc_C3R](#)
[image_statistics_functions](#), 96
- [nppiMaximumErrorGetBufferHostSize_32fc_C4R](#)
[image_statistics_functions](#), 96
- [nppiMaximumErrorGetBufferHostSize_32s_C1R](#)
[image_statistics_functions](#), 97
- [nppiMaximumErrorGetBufferHostSize_32s_C2R](#)
[image_statistics_functions](#), 97
- [nppiMaximumErrorGetBufferHostSize_32s_C3R](#)
[image_statistics_functions](#), 97
- [nppiMaximumErrorGetBufferHostSize_32s_C4R](#)
[image_statistics_functions](#), 97
- [nppiMaximumErrorGetBufferHostSize_32sc_C1R](#)
[image_statistics_functions](#), 98
- [nppiMaximumErrorGetBufferHostSize_32sc_C2R](#)
[image_statistics_functions](#), 98
- [nppiMaximumErrorGetBufferHostSize_32sc_C3R](#)
[image_statistics_functions](#), 98
- [nppiMaximumErrorGetBufferHostSize_32sc_C4R](#)
[image_statistics_functions](#), 99
- [nppiMaximumErrorGetBufferHostSize_32u_C1R](#)
[image_statistics_functions](#), 99
- [nppiMaximumErrorGetBufferHostSize_32u_C2R](#)
[image_statistics_functions](#), 99
- [nppiMaximumErrorGetBufferHostSize_32u_C3R](#)
[image_statistics_functions](#), 99
- [nppiMaximumErrorGetBufferHostSize_32u_C4R](#)
[image_statistics_functions](#), 100
- [nppiMaximumErrorGetBufferHostSize_64f_C1R](#)
[image_statistics_functions](#), 100
- [nppiMaximumErrorGetBufferHostSize_64f_C2R](#)
[image_statistics_functions](#), 100
- [nppiMaximumErrorGetBufferHostSize_64f_C3R](#)
[image_statistics_functions](#), 101
- [nppiMaximumErrorGetBufferHostSize_64f_C4R](#)
[image_statistics_functions](#), 101
- [nppiMaximumErrorGetBufferHostSize_8s_C1R](#)
[image_statistics_functions](#), 101
- [nppiMaximumErrorGetBufferHostSize_8s_C2R](#)
[image_statistics_functions](#), 101
- [nppiMaximumErrorGetBufferHostSize_8s_C3R](#)
[image_statistics_functions](#), 102
- [nppiMaximumErrorGetBufferHostSize_8s_C4R](#)
[image_statistics_functions](#), 102
- [nppiMaximumErrorGetBufferHostSize_8u_C1R](#)
[image_statistics_functions](#), 102
- [nppiMaximumErrorGetBufferHostSize_8u_C2R](#)
[image_statistics_functions](#), 103
- [nppiMaximumErrorGetBufferHostSize_8u_C3R](#)
[image_statistics_functions](#), 103
- [nppiMaximumErrorGetBufferHostSize_8u_C4R](#)
[image_statistics_functions](#), 103
- [nppiMaximumRelativeError_16s_C1R](#)
[image_maximum_relative_error](#), 730
- [nppiMaximumRelativeError_16s_C2R](#)
[image_maximum_relative_error](#), 731
- [nppiMaximumRelativeError_16s_C3R](#)
[image_maximum_relative_error](#), 731
- [nppiMaximumRelativeError_16s_C4R](#)
[image_maximum_relative_error](#), 732
- [nppiMaximumRelativeError_16sc_C1R](#)
[image_maximum_relative_error](#), 732
- [nppiMaximumRelativeError_16sc_C2R](#)
[image_maximum_relative_error](#), 733
- [nppiMaximumRelativeError_16sc_C3R](#)
[image_maximum_relative_error](#), 733
- [nppiMaximumRelativeError_16sc_C4R](#)
[image_maximum_relative_error](#), 733
- [nppiMaximumRelativeError_16u_C1R](#)
[image_maximum_relative_error](#), 734
- [nppiMaximumRelativeError_16u_C2R](#)
[image_maximum_relative_error](#), 734
- [nppiMaximumRelativeError_16u_C3R](#)
[image_maximum_relative_error](#), 735
- [nppiMaximumRelativeError_16u_C4R](#)
[image_maximum_relative_error](#), 735
- [nppiMaximumRelativeError_32f_C1R](#)
[image_maximum_relative_error](#), 736
- [nppiMaximumRelativeError_32f_C2R](#)
[image_maximum_relative_error](#), 736
- [nppiMaximumRelativeError_32f_C3R](#)
[image_maximum_relative_error](#), 737
- [nppiMaximumRelativeError_32f_C4R](#)
[image_maximum_relative_error](#), 737
- [nppiMaximumRelativeError_32fc_C1R](#)
[image_maximum_relative_error](#), 738
- [nppiMaximumRelativeError_32fc_C2R](#)
[image_maximum_relative_error](#), 738
- [nppiMaximumRelativeError_32fc_C3R](#)
[image_maximum_relative_error](#), 738

- nppiMaximumRelativeError_32fc_C4R
image_maximum_relative_error, 739
- nppiMaximumRelativeError_32s_C1R
image_maximum_relative_error, 739
- nppiMaximumRelativeError_32s_C2R
image_maximum_relative_error, 740
- nppiMaximumRelativeError_32s_C3R
image_maximum_relative_error, 740
- nppiMaximumRelativeError_32s_C4R
image_maximum_relative_error, 741
- nppiMaximumRelativeError_32sc_C1R
image_maximum_relative_error, 741
- nppiMaximumRelativeError_32sc_C2R
image_maximum_relative_error, 742
- nppiMaximumRelativeError_32sc_C3R
image_maximum_relative_error, 742
- nppiMaximumRelativeError_32sc_C4R
image_maximum_relative_error, 743
- nppiMaximumRelativeError_32u_C1R
image_maximum_relative_error, 743
- nppiMaximumRelativeError_32u_C2R
image_maximum_relative_error, 743
- nppiMaximumRelativeError_32u_C3R
image_maximum_relative_error, 744
- nppiMaximumRelativeError_32u_C4R
image_maximum_relative_error, 744
- nppiMaximumRelativeError_64f_C1R
image_maximum_relative_error, 745
- nppiMaximumRelativeError_64f_C2R
image_maximum_relative_error, 745
- nppiMaximumRelativeError_64f_C3R
image_maximum_relative_error, 746
- nppiMaximumRelativeError_64f_C4R
image_maximum_relative_error, 746
- nppiMaximumRelativeError_8s_C1R
image_maximum_relative_error, 747
- nppiMaximumRelativeError_8s_C2R
image_maximum_relative_error, 747
- nppiMaximumRelativeError_8s_C3R
image_maximum_relative_error, 748
- nppiMaximumRelativeError_8s_C4R
image_maximum_relative_error, 748
- nppiMaximumRelativeError_8u_C1R
image_maximum_relative_error, 748
- nppiMaximumRelativeError_8u_C2R
image_maximum_relative_error, 749
- nppiMaximumRelativeError_8u_C3R
image_maximum_relative_error, 749
- nppiMaximumRelativeError_8u_C4R
image_maximum_relative_error, 750
- nppiMaximumRelativeErrorGetBufferHostSize_-
16s_C1R
image_statistics_functions, 103
- nppiMaximumRelativeErrorGetBufferHostSize_-
16s_C2R
image_statistics_functions, 104
- nppiMaximumRelativeErrorGetBufferHostSize_-
16s_C3R
image_statistics_functions, 104
- nppiMaximumRelativeErrorGetBufferHostSize_-
16s_C4R
image_statistics_functions, 104
- nppiMaximumRelativeErrorGetBufferHostSize_-
16sc_C1R
image_statistics_functions, 105
- nppiMaximumRelativeErrorGetBufferHostSize_-
16sc_C2R
image_statistics_functions, 105
- nppiMaximumRelativeErrorGetBufferHostSize_-
16sc_C3R
image_statistics_functions, 105
- nppiMaximumRelativeErrorGetBufferHostSize_-
16sc_C4R
image_statistics_functions, 105
- nppiMaximumRelativeErrorGetBufferHostSize_-
16u_C1R
image_statistics_functions, 106
- nppiMaximumRelativeErrorGetBufferHostSize_-
16u_C2R
image_statistics_functions, 106
- nppiMaximumRelativeErrorGetBufferHostSize_-
16u_C3R
image_statistics_functions, 106
- nppiMaximumRelativeErrorGetBufferHostSize_-
16u_C4R
image_statistics_functions, 107
- nppiMaximumRelativeErrorGetBufferHostSize_-
32f_C1R
image_statistics_functions, 107
- nppiMaximumRelativeErrorGetBufferHostSize_-
32f_C2R
image_statistics_functions, 107
- nppiMaximumRelativeErrorGetBufferHostSize_-
32f_C3R
image_statistics_functions, 107
- nppiMaximumRelativeErrorGetBufferHostSize_-
32f_C4R
image_statistics_functions, 108
- nppiMaximumRelativeErrorGetBufferHostSize_-
32fc_C1R
image_statistics_functions, 108
- nppiMaximumRelativeErrorGetBufferHostSize_-
32fc_C2R
image_statistics_functions, 108
- nppiMaximumRelativeErrorGetBufferHostSize_-
32fc_C3R
image_statistics_functions, 109

- [nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C4R](#)
[image_statistics_functions, 109](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C1R](#)
[image_statistics_functions, 109](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C2R](#)
[image_statistics_functions, 109](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C3R](#)
[image_statistics_functions, 110](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C4R](#)
[image_statistics_functions, 110](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C1R](#)
[image_statistics_functions, 110](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C2R](#)
[image_statistics_functions, 111](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C3R](#)
[image_statistics_functions, 111](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C4R](#)
[image_statistics_functions, 111](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32u_C1R](#)
[image_statistics_functions, 111](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32u_C2R](#)
[image_statistics_functions, 112](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32u_C3R](#)
[image_statistics_functions, 112](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-32u_C4R](#)
[image_statistics_functions, 112](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-64f_C1R](#)
[image_statistics_functions, 113](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-64f_C2R](#)
[image_statistics_functions, 113](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-64f_C3R](#)
[image_statistics_functions, 113](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-64f_C4R](#)
[image_statistics_functions, 113](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8s_C1R](#)
[image_statistics_functions, 114](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8s_C2R](#)
[image_statistics_functions, 114](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8s_C3R](#)
[image_statistics_functions, 114](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8s_C4R](#)
[image_statistics_functions, 115](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8u_C1R](#)
[image_statistics_functions, 115](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8u_C2R](#)
[image_statistics_functions, 115](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8u_C3R](#)
[image_statistics_functions, 115](#)
- [nppiMaximumRelativeErrorGetBufferHostSize_-8u_C4R](#)
[image_statistics_functions, 116](#)
- [nppiMaxIndx_16s_AC4R](#)
[image_max_index, 174](#)
- [nppiMaxIndx_16s_C1R](#)
[image_max_index, 175](#)
- [nppiMaxIndx_16s_C3R](#)
[image_max_index, 175](#)
- [nppiMaxIndx_16s_C4R](#)
[image_max_index, 175](#)
- [nppiMaxIndx_16u_AC4R](#)
[image_max_index, 176](#)
- [nppiMaxIndx_16u_C1R](#)
[image_max_index, 176](#)
- [nppiMaxIndx_16u_C3R](#)
[image_max_index, 177](#)
- [nppiMaxIndx_16u_C4R](#)
[image_max_index, 177](#)
- [nppiMaxIndx_32f_AC4R](#)
[image_max_index, 177](#)
- [nppiMaxIndx_32f_C1R](#)
[image_max_index, 178](#)
- [nppiMaxIndx_32f_C3R](#)
[image_max_index, 178](#)
- [nppiMaxIndx_32f_C4R](#)
[image_max_index, 179](#)
- [nppiMaxIndx_8u_AC4R](#)
[image_max_index, 179](#)
- [nppiMaxIndx_8u_C1R](#)
[image_max_index, 179](#)
- [nppiMaxIndx_8u_C3R](#)
[image_max_index, 180](#)
- [nppiMaxIndx_8u_C4R](#)
[image_max_index, 180](#)
- [nppiMaxIndxGetBufferHostSize_16s_AC4R](#)

- image_max_index, 181
- nppiMaxIdxGetBufferHostSize_16s_C1R
 - image_max_index, 181
- nppiMaxIdxGetBufferHostSize_16s_C3R
 - image_max_index, 181
- nppiMaxIdxGetBufferHostSize_16s_C4R
 - image_max_index, 182
- nppiMaxIdxGetBufferHostSize_16u_AC4R
 - image_max_index, 182
- nppiMaxIdxGetBufferHostSize_16u_C1R
 - image_max_index, 182
- nppiMaxIdxGetBufferHostSize_16u_C3R
 - image_max_index, 182
- nppiMaxIdxGetBufferHostSize_16u_C4R
 - image_max_index, 183
- nppiMaxIdxGetBufferHostSize_32f_AC4R
 - image_max_index, 183
- nppiMaxIdxGetBufferHostSize_32f_C1R
 - image_max_index, 183
- nppiMaxIdxGetBufferHostSize_32f_C3R
 - image_max_index, 184
- nppiMaxIdxGetBufferHostSize_32f_C4R
 - image_max_index, 184
- nppiMaxIdxGetBufferHostSize_8u_AC4R
 - image_max_index, 184
- nppiMaxIdxGetBufferHostSize_8u_C1R
 - image_max_index, 184
- nppiMaxIdxGetBufferHostSize_8u_C3R
 - image_max_index, 185
- nppiMaxIdxGetBufferHostSize_8u_C4R
 - image_max_index, 185
- nppiMean_16s_AC4R
 - image_mean, 221
- nppiMean_16s_C1R
 - image_mean, 221
- nppiMean_16s_C3R
 - image_mean, 221
- nppiMean_16s_C4R
 - image_mean, 222
- nppiMean_16u_AC4R
 - image_mean, 222
- nppiMean_16u_C1MR
 - image_mean, 222
- nppiMean_16u_C1R
 - image_mean, 223
- nppiMean_16u_C3CMR
 - image_mean, 223
- nppiMean_16u_C3R
 - image_mean, 223
- nppiMean_16u_C4R
 - image_mean, 224
- nppiMean_32f_AC4R
 - image_mean, 224
- nppiMean_32f_C1MR
 - image_mean, 225
- nppiMean_32f_C1R
 - image_mean, 225
- nppiMean_32f_C3CMR
 - image_mean, 225
- nppiMean_32f_C3R
 - image_mean, 226
- nppiMean_32f_C4R
 - image_mean, 226
- nppiMean_8s_C1MR
 - image_mean, 227
- nppiMean_8s_C3CMR
 - image_mean, 227
- nppiMean_8u_AC4R
 - image_mean, 228
- nppiMean_8u_C1MR
 - image_mean, 228
- nppiMean_8u_C1R
 - image_mean, 228
- nppiMean_8u_C3CMR
 - image_mean, 229
- nppiMean_8u_C3R
 - image_mean, 229
- nppiMean_8u_C4R
 - image_mean, 230
- nppiMean_StdDev_16u_C1MR
 - image_mean_stddev, 241
- nppiMean_StdDev_16u_C1R
 - image_mean_stddev, 241
- nppiMean_StdDev_16u_C3CMR
 - image_mean_stddev, 242
- nppiMean_StdDev_16u_C3CR
 - image_mean_stddev, 242
- nppiMean_StdDev_32f_C1MR
 - image_mean_stddev, 243
- nppiMean_StdDev_32f_C1R
 - image_mean_stddev, 243
- nppiMean_StdDev_32f_C3CMR
 - image_mean_stddev, 244
- nppiMean_StdDev_32f_C3CR
 - image_mean_stddev, 244
- nppiMean_StdDev_8s_C1MR
 - image_mean_stddev, 245
- nppiMean_StdDev_8s_C1R
 - image_mean_stddev, 245
- nppiMean_StdDev_8s_C3CMR
 - image_mean_stddev, 246
- nppiMean_StdDev_8s_C3CR
 - image_mean_stddev, 246
- nppiMean_StdDev_8u_C1MR
 - image_mean_stddev, 247
- nppiMean_StdDev_8u_C1R
 - image_mean_stddev, 247
- nppiMean_StdDev_8u_C3CMR

- image_mean_stddev, 248
- nppiMean_StdDev_8u_C3CR
 - image_mean_stddev, 248
- nppiMeanGetBufferHostSize_16s_AC4R
 - image_mean, 230
- nppiMeanGetBufferHostSize_16s_C1R
 - image_mean, 230
- nppiMeanGetBufferHostSize_16s_C3R
 - image_mean, 231
- nppiMeanGetBufferHostSize_16s_C4R
 - image_mean, 231
- nppiMeanGetBufferHostSize_16u_AC4R
 - image_mean, 231
- nppiMeanGetBufferHostSize_16u_C1MR
 - image_mean, 231
- nppiMeanGetBufferHostSize_16u_C1R
 - image_mean, 232
- nppiMeanGetBufferHostSize_16u_C3CMR
 - image_mean, 232
- nppiMeanGetBufferHostSize_16u_C3R
 - image_mean, 232
- nppiMeanGetBufferHostSize_16u_C4R
 - image_mean, 233
- nppiMeanGetBufferHostSize_32f_AC4R
 - image_mean, 233
- nppiMeanGetBufferHostSize_32f_C1MR
 - image_mean, 233
- nppiMeanGetBufferHostSize_32f_C1R
 - image_mean, 233
- nppiMeanGetBufferHostSize_32f_C3CMR
 - image_mean, 234
- nppiMeanGetBufferHostSize_32f_C3R
 - image_mean, 234
- nppiMeanGetBufferHostSize_32f_C4R
 - image_mean, 234
- nppiMeanGetBufferHostSize_8s_C1MR
 - image_mean, 235
- nppiMeanGetBufferHostSize_8s_C3CMR
 - image_mean, 235
- nppiMeanGetBufferHostSize_8u_AC4R
 - image_mean, 235
- nppiMeanGetBufferHostSize_8u_C1MR
 - image_mean, 235
- nppiMeanGetBufferHostSize_8u_C1R
 - image_mean, 236
- nppiMeanGetBufferHostSize_8u_C3CMR
 - image_mean, 236
- nppiMeanGetBufferHostSize_8u_C3R
 - image_mean, 236
- nppiMeanGetBufferHostSize_8u_C4R
 - image_mean, 237
- nppiMeanStdDevGetBufferHostSize_16u_C1MR
 - image_mean_stddev, 249
- nppiMeanStdDevGetBufferHostSize_16u_C1R
 - image_mean_stddev, 249
- nppiMeanStdDevGetBufferHostSize_16u_C3CMR
 - image_mean_stddev, 249
- nppiMeanStdDevGetBufferHostSize_16u_C3CR
 - image_mean_stddev, 250
- nppiMeanStdDevGetBufferHostSize_32f_C1MR
 - image_mean_stddev, 250
- nppiMeanStdDevGetBufferHostSize_32f_C1R
 - image_mean_stddev, 250
- nppiMeanStdDevGetBufferHostSize_32f_C3CMR
 - image_mean_stddev, 251
- nppiMeanStdDevGetBufferHostSize_32f_C3CR
 - image_mean_stddev, 251
- nppiMeanStdDevGetBufferHostSize_8s_C1MR
 - image_mean_stddev, 251
- nppiMeanStdDevGetBufferHostSize_8s_C1R
 - image_mean_stddev, 251
- nppiMeanStdDevGetBufferHostSize_8s_C3CMR
 - image_mean_stddev, 252
- nppiMeanStdDevGetBufferHostSize_8s_C3CR
 - image_mean_stddev, 252
- nppiMeanStdDevGetBufferHostSize_8u_C1MR
 - image_mean_stddev, 252
- nppiMeanStdDevGetBufferHostSize_8u_C1R
 - image_mean_stddev, 253
- nppiMeanStdDevGetBufferHostSize_8u_C3CMR
 - image_mean_stddev, 253
- nppiMeanStdDevGetBufferHostSize_8u_C3CR
 - image_mean_stddev, 253
- nppiMin_16s_AC4R
 - image_min, 134
- nppiMin_16s_C1R
 - image_min, 134
- nppiMin_16s_C3R
 - image_min, 135
- nppiMin_16s_C4R
 - image_min, 135
- nppiMin_16u_AC4R
 - image_min, 135
- nppiMin_16u_C1R
 - image_min, 136
- nppiMin_16u_C3R
 - image_min, 136
- nppiMin_16u_C4R
 - image_min, 137
- nppiMin_32f_AC4R
 - image_min, 137
- nppiMin_32f_C1R
 - image_min, 137
- nppiMin_32f_C3R
 - image_min, 138
- nppiMin_32f_C4R
 - image_min, 138
- nppiMin_8u_AC4R

- image_min, 138
- nppiMin_8u_C1R
 - image_min, 139
- nppiMin_8u_C3R
 - image_min, 139
- nppiMin_8u_C4R
 - image_min, 140
- nppiMinEvery_16s_AC4IR
 - image_minevery, 497
- nppiMinEvery_16s_C1IR
 - image_minevery, 497
- nppiMinEvery_16s_C3IR
 - image_minevery, 498
- nppiMinEvery_16s_C4IR
 - image_minevery, 498
- nppiMinEvery_16u_AC4IR
 - image_minevery, 498
- nppiMinEvery_16u_C1IR
 - image_minevery, 499
- nppiMinEvery_16u_C3IR
 - image_minevery, 499
- nppiMinEvery_16u_C4IR
 - image_minevery, 499
- nppiMinEvery_32f_AC4IR
 - image_minevery, 500
- nppiMinEvery_32f_C1IR
 - image_minevery, 500
- nppiMinEvery_32f_C3IR
 - image_minevery, 500
- nppiMinEvery_32f_C4IR
 - image_minevery, 501
- nppiMinEvery_8u_AC4IR
 - image_minevery, 501
- nppiMinEvery_8u_C1IR
 - image_minevery, 501
- nppiMinEvery_8u_C3IR
 - image_minevery, 502
- nppiMinEvery_8u_C4IR
 - image_minevery, 502
- nppiMinGetBufferHostSize_16s_AC4R
 - image_min, 140
- nppiMinGetBufferHostSize_16s_C1R
 - image_min, 140
- nppiMinGetBufferHostSize_16s_C3R
 - image_min, 140
- nppiMinGetBufferHostSize_16s_C4R
 - image_min, 141
- nppiMinGetBufferHostSize_16u_AC4R
 - image_min, 141
- nppiMinGetBufferHostSize_16u_C1R
 - image_min, 141
- nppiMinGetBufferHostSize_16u_C3R
 - image_min, 141
- nppiMinGetBufferHostSize_16u_C4R
 - image_min, 142
- nppiMinGetBufferHostSize_32f_AC4R
 - image_min, 142
- nppiMinGetBufferHostSize_32f_C1R
 - image_min, 142
- nppiMinGetBufferHostSize_32f_C3R
 - image_min, 142
- nppiMinGetBufferHostSize_32f_C4R
 - image_min, 143
- nppiMinGetBufferHostSize_8u_AC4R
 - image_min, 143
- nppiMinGetBufferHostSize_8u_C1R
 - image_min, 143
- nppiMinGetBufferHostSize_8u_C3R
 - image_min, 143
- nppiMinGetBufferHostSize_8u_C4R
 - image_min, 144
- nppiMinIndx_16s_AC4R
 - image_min_index, 147
- nppiMinIndx_16s_C1R
 - image_min_index, 148
- nppiMinIndx_16s_C3R
 - image_min_index, 148
- nppiMinIndx_16s_C4R
 - image_min_index, 148
- nppiMinIndx_16u_AC4R
 - image_min_index, 149
- nppiMinIndx_16u_C1R
 - image_min_index, 149
- nppiMinIndx_16u_C3R
 - image_min_index, 150
- nppiMinIndx_16u_C4R
 - image_min_index, 150
- nppiMinIndx_32f_AC4R
 - image_min_index, 150
- nppiMinIndx_32f_C1R
 - image_min_index, 151
- nppiMinIndx_32f_C3R
 - image_min_index, 151
- nppiMinIndx_32f_C4R
 - image_min_index, 152
- nppiMinIndx_8u_AC4R
 - image_min_index, 152
- nppiMinIndx_8u_C1R
 - image_min_index, 152
- nppiMinIndx_8u_C3R
 - image_min_index, 153
- nppiMinIndx_8u_C4R
 - image_min_index, 153
- nppiMinIndxGetBufferHostSize_16s_AC4R
 - image_min_index, 154
- nppiMinIndxGetBufferHostSize_16s_C1R
 - image_min_index, 154
- nppiMinIndxGetBufferHostSize_16s_C3R

- image_min_index, 154
- nppiMinIdxGetBufferHostSize_16s_C4R
 - image_min_index, 155
- nppiMinIdxGetBufferHostSize_16u_AC4R
 - image_min_index, 155
- nppiMinIdxGetBufferHostSize_16u_C1R
 - image_min_index, 155
- nppiMinIdxGetBufferHostSize_16u_C3R
 - image_min_index, 155
- nppiMinIdxGetBufferHostSize_16u_C4R
 - image_min_index, 156
- nppiMinIdxGetBufferHostSize_32f_AC4R
 - image_min_index, 156
- nppiMinIdxGetBufferHostSize_32f_C1R
 - image_min_index, 156
- nppiMinIdxGetBufferHostSize_32f_C3R
 - image_min_index, 157
- nppiMinIdxGetBufferHostSize_32f_C4R
 - image_min_index, 157
- nppiMinIdxGetBufferHostSize_8u_AC4R
 - image_min_index, 157
- nppiMinIdxGetBufferHostSize_8u_C1R
 - image_min_index, 157
- nppiMinIdxGetBufferHostSize_8u_C3R
 - image_min_index, 158
- nppiMinIdxGetBufferHostSize_8u_C4R
 - image_min_index, 158
- nppiMinMax_16s_AC4R
 - image_min_max, 188
- nppiMinMax_16s_C1R
 - image_min_max, 188
- nppiMinMax_16s_C3R
 - image_min_max, 189
- nppiMinMax_16s_C4R
 - image_min_max, 189
- nppiMinMax_16u_AC4R
 - image_min_max, 190
- nppiMinMax_16u_C1R
 - image_min_max, 190
- nppiMinMax_16u_C3R
 - image_min_max, 190
- nppiMinMax_16u_C4R
 - image_min_max, 191
- nppiMinMax_32f_AC4R
 - image_min_max, 191
- nppiMinMax_32f_C1R
 - image_min_max, 192
- nppiMinMax_32f_C3R
 - image_min_max, 192
- nppiMinMax_32f_C4R
 - image_min_max, 192
- nppiMinMax_8u_AC4R
 - image_min_max, 193
- nppiMinMax_8u_C1R
 - image_min_max, 193
- nppiMinMax_8u_C3R
 - image_min_max, 194
- nppiMinMax_8u_C4R
 - image_min_max, 194
- nppiMinMaxGetBufferHostSize_16s_AC4R
 - image_min_max, 194
- nppiMinMaxGetBufferHostSize_16s_C1R
 - image_min_max, 195
- nppiMinMaxGetBufferHostSize_16s_C3R
 - image_min_max, 195
- nppiMinMaxGetBufferHostSize_16s_C4R
 - image_min_max, 195
- nppiMinMaxGetBufferHostSize_16u_AC4R
 - image_min_max, 196
- nppiMinMaxGetBufferHostSize_16u_C1R
 - image_min_max, 196
- nppiMinMaxGetBufferHostSize_16u_C3R
 - image_min_max, 196
- nppiMinMaxGetBufferHostSize_16u_C4R
 - image_min_max, 196
- nppiMinMaxGetBufferHostSize_32f_AC4R
 - image_min_max, 197
- nppiMinMaxGetBufferHostSize_32f_C1R
 - image_min_max, 197
- nppiMinMaxGetBufferHostSize_32f_C3R
 - image_min_max, 197
- nppiMinMaxGetBufferHostSize_32f_C4R
 - image_min_max, 198
- nppiMinMaxGetBufferHostSize_8u_AC4R
 - image_min_max, 198
- nppiMinMaxGetBufferHostSize_8u_C1R
 - image_min_max, 198
- nppiMinMaxGetBufferHostSize_8u_C3R
 - image_min_max, 198
- nppiMinMaxGetBufferHostSize_8u_C4R
 - image_min_max, 199
- nppiMinMaxIdx_16u_C1MR
 - image_min_max_index, 203
- nppiMinMaxIdx_16u_C1R
 - image_min_max_index, 204
- nppiMinMaxIdx_16u_C3CMR
 - image_min_max_index, 204
- nppiMinMaxIdx_16u_C3CR
 - image_min_max_index, 205
- nppiMinMaxIdx_32f_C1MR
 - image_min_max_index, 205
- nppiMinMaxIdx_32f_C1R
 - image_min_max_index, 206
- nppiMinMaxIdx_32f_C3CMR
 - image_min_max_index, 206
- nppiMinMaxIdx_32f_C3CR
 - image_min_max_index, 207
- nppiMinMaxIdx_8s_C1MR

- image_min_max_index, 208
- nppiMinMaxIndx_8s_C1R
 - image_min_max_index, 208
- nppiMinMaxIndx_8s_C3CMR
 - image_min_max_index, 209
- nppiMinMaxIndx_8s_C3CR
 - image_min_max_index, 209
- nppiMinMaxIndx_8u_C1MR
 - image_min_max_index, 210
- nppiMinMaxIndx_8u_C1R
 - image_min_max_index, 210
- nppiMinMaxIndx_8u_C3CMR
 - image_min_max_index, 211
- nppiMinMaxIndx_8u_C3CR
 - image_min_max_index, 211
- nppiMinMaxIndxGetBufferHostSize_16u_C1MR
 - image_min_max_index, 212
- nppiMinMaxIndxGetBufferHostSize_16u_C1R
 - image_min_max_index, 212
- nppiMinMaxIndxGetBufferHostSize_16u_C3CMR
 - image_min_max_index, 212
- nppiMinMaxIndxGetBufferHostSize_16u_C3CR
 - image_min_max_index, 213
- nppiMinMaxIndxGetBufferHostSize_32f_C1MR
 - image_min_max_index, 213
- nppiMinMaxIndxGetBufferHostSize_32f_C1R
 - image_min_max_index, 213
- nppiMinMaxIndxGetBufferHostSize_32f_C3CMR
 - image_min_max_index, 214
- nppiMinMaxIndxGetBufferHostSize_32f_C3CR
 - image_min_max_index, 214
- nppiMinMaxIndxGetBufferHostSize_8s_C1MR
 - image_min_max_index, 214
- nppiMinMaxIndxGetBufferHostSize_8s_C1R
 - image_min_max_index, 214
- nppiMinMaxIndxGetBufferHostSize_8s_C3CMR
 - image_min_max_index, 215
- nppiMinMaxIndxGetBufferHostSize_8s_C3CR
 - image_min_max_index, 215
- nppiMinMaxIndxGetBufferHostSize_8u_C1MR
 - image_min_max_index, 215
- nppiMinMaxIndxGetBufferHostSize_8u_C1R
 - image_min_max_index, 216
- nppiMinMaxIndxGetBufferHostSize_8u_C3CMR
 - image_min_max_index, 216
- nppiMinMaxIndxGetBufferHostSize_8u_C3CR
 - image_min_max_index, 216
- NppiNorm
 - typedefs_npp, 41
- nppiNorm_Inf_16s_AC4R
 - image_inf_norm, 260
- nppiNorm_Inf_16s_C1R
 - image_inf_norm, 260
- nppiNorm_Inf_16s_C3R
 - image_inf_norm, 260
- nppiNorm_Inf_16s_C4R
 - image_inf_norm, 261
- nppiNorm_Inf_16u_AC4R
 - image_inf_norm, 261
- nppiNorm_Inf_16u_C1MR
 - image_inf_norm, 261
- nppiNorm_Inf_16u_C1R
 - image_inf_norm, 262
- nppiNorm_Inf_16u_C3CMR
 - image_inf_norm, 262
- nppiNorm_Inf_16u_C3R
 - image_inf_norm, 263
- nppiNorm_Inf_16u_C4R
 - image_inf_norm, 263
- nppiNorm_Inf_32f_AC4R
 - image_inf_norm, 263
- nppiNorm_Inf_32f_C1MR
 - image_inf_norm, 264
- nppiNorm_Inf_32f_C1R
 - image_inf_norm, 264
- nppiNorm_Inf_32f_C3CMR
 - image_inf_norm, 265
- nppiNorm_Inf_32f_C3R
 - image_inf_norm, 265
- nppiNorm_Inf_32f_C4R
 - image_inf_norm, 265
- nppiNorm_Inf_32s_C1R
 - image_inf_norm, 266
- nppiNorm_Inf_8s_C1MR
 - image_inf_norm, 266
- nppiNorm_Inf_8s_C3CMR
 - image_inf_norm, 267
- nppiNorm_Inf_8u_AC4R
 - image_inf_norm, 267
- nppiNorm_Inf_8u_C1MR
 - image_inf_norm, 267
- nppiNorm_Inf_8u_C1R
 - image_inf_norm, 268
- nppiNorm_Inf_8u_C3CMR
 - image_inf_norm, 268
- nppiNorm_Inf_8u_C3R
 - image_inf_norm, 269
- nppiNorm_Inf_8u_C4R
 - image_inf_norm, 269
- nppiNorm_L1_16s_AC4R
 - image_L1_norm, 282
- nppiNorm_L1_16s_C1R
 - image_L1_norm, 282
- nppiNorm_L1_16s_C3R
 - image_L1_norm, 282
- nppiNorm_L1_16s_C4R
 - image_L1_norm, 283
- nppiNorm_L1_16u_AC4R
 - image_L1_norm, 283

- image_L1_norm, 283
- nppiNorm_L1_16u_C1MR
 - image_L1_norm, 283
- nppiNorm_L1_16u_C1R
 - image_L1_norm, 284
- nppiNorm_L1_16u_C3CMR
 - image_L1_norm, 284
- nppiNorm_L1_16u_C3R
 - image_L1_norm, 285
- nppiNorm_L1_16u_C4R
 - image_L1_norm, 285
- nppiNorm_L1_32f_AC4R
 - image_L1_norm, 285
- nppiNorm_L1_32f_C1MR
 - image_L1_norm, 286
- nppiNorm_L1_32f_C1R
 - image_L1_norm, 286
- nppiNorm_L1_32f_C3CMR
 - image_L1_norm, 286
- nppiNorm_L1_32f_C3R
 - image_L1_norm, 287
- nppiNorm_L1_32f_C4R
 - image_L1_norm, 287
- nppiNorm_L1_8s_C1MR
 - image_L1_norm, 288
- nppiNorm_L1_8s_C3CMR
 - image_L1_norm, 288
- nppiNorm_L1_8u_AC4R
 - image_L1_norm, 288
- nppiNorm_L1_8u_C1MR
 - image_L1_norm, 289
- nppiNorm_L1_8u_C1R
 - image_L1_norm, 289
- nppiNorm_L1_8u_C3CMR
 - image_L1_norm, 290
- nppiNorm_L1_8u_C3R
 - image_L1_norm, 290
- nppiNorm_L1_8u_C4R
 - image_L1_norm, 290
- nppiNorm_L2_16s_AC4R
 - image_L2_norm, 303
- nppiNorm_L2_16s_C1R
 - image_L2_norm, 303
- nppiNorm_L2_16s_C3R
 - image_L2_norm, 303
- nppiNorm_L2_16s_C4R
 - image_L2_norm, 304
- nppiNorm_L2_16u_AC4R
 - image_L2_norm, 304
- nppiNorm_L2_16u_C1MR
 - image_L2_norm, 304
- nppiNorm_L2_16u_C1R
 - image_L2_norm, 305
- nppiNorm_L2_16u_C3CMR
 - image_L2_norm, 305
- nppiNorm_L2_16u_C3R
 - image_L2_norm, 306
- nppiNorm_L2_16u_C4R
 - image_L2_norm, 306
- nppiNorm_L2_32f_AC4R
 - image_L2_norm, 306
- nppiNorm_L2_32f_C1MR
 - image_L2_norm, 307
- nppiNorm_L2_32f_C1R
 - image_L2_norm, 307
- nppiNorm_L2_32f_C3CMR
 - image_L2_norm, 307
- nppiNorm_L2_32f_C3R
 - image_L2_norm, 308
- nppiNorm_L2_32f_C4R
 - image_L2_norm, 308
- nppiNorm_L2_8s_C1MR
 - image_L2_norm, 309
- nppiNorm_L2_8s_C3CMR
 - image_L2_norm, 309
- nppiNorm_L2_8u_AC4R
 - image_L2_norm, 309
- nppiNorm_L2_8u_C1MR
 - image_L2_norm, 310
- nppiNorm_L2_8u_C1R
 - image_L2_norm, 310
- nppiNorm_L2_8u_C3CMR
 - image_L2_norm, 311
- nppiNorm_L2_8u_C3R
 - image_L2_norm, 311
- nppiNorm_L2_8u_C4R
 - image_L2_norm, 311
- nppiNormDiff_Inf_16s_AC4R
 - image_inf_normdiff, 324
- nppiNormDiff_Inf_16s_C1R
 - image_inf_normdiff, 324
- nppiNormDiff_Inf_16s_C3R
 - image_inf_normdiff, 325
- nppiNormDiff_Inf_16s_C4R
 - image_inf_normdiff, 325
- nppiNormDiff_Inf_16u_AC4R
 - image_inf_normdiff, 326
- nppiNormDiff_Inf_16u_C1MR
 - image_inf_normdiff, 326
- nppiNormDiff_Inf_16u_C1R
 - image_inf_normdiff, 327
- nppiNormDiff_Inf_16u_C3CMR
 - image_inf_normdiff, 327
- nppiNormDiff_Inf_16u_C3R
 - image_inf_normdiff, 328
- nppiNormDiff_Inf_16u_C4R
 - image_inf_normdiff, 328
- nppiNormDiff_Inf_32f_AC4R

- image_inf_normdiff, 328
- nppiNormDiff_Inf_32f_C1MR
 - image_inf_normdiff, 329
- nppiNormDiff_Inf_32f_C1R
 - image_inf_normdiff, 329
- nppiNormDiff_Inf_32f_C3CMR
 - image_inf_normdiff, 330
- nppiNormDiff_Inf_32f_C3R
 - image_inf_normdiff, 330
- nppiNormDiff_Inf_32f_C4R
 - image_inf_normdiff, 331
- nppiNormDiff_Inf_8s_C1MR
 - image_inf_normdiff, 331
- nppiNormDiff_Inf_8s_C3CMR
 - image_inf_normdiff, 332
- nppiNormDiff_Inf_8u_AC4R
 - image_inf_normdiff, 332
- nppiNormDiff_Inf_8u_C1MR
 - image_inf_normdiff, 333
- nppiNormDiff_Inf_8u_C1R
 - image_inf_normdiff, 333
- nppiNormDiff_Inf_8u_C3CMR
 - image_inf_normdiff, 334
- nppiNormDiff_Inf_8u_C3R
 - image_inf_normdiff, 334
- nppiNormDiff_Inf_8u_C4R
 - image_inf_normdiff, 335
- nppiNormDiff_L1_16s_AC4R
 - image_L1_normdiff, 347
- nppiNormDiff_L1_16s_C1R
 - image_L1_normdiff, 347
- nppiNormDiff_L1_16s_C3R
 - image_L1_normdiff, 348
- nppiNormDiff_L1_16s_C4R
 - image_L1_normdiff, 348
- nppiNormDiff_L1_16u_AC4R
 - image_L1_normdiff, 349
- nppiNormDiff_L1_16u_C1MR
 - image_L1_normdiff, 349
- nppiNormDiff_L1_16u_C1R
 - image_L1_normdiff, 349
- nppiNormDiff_L1_16u_C3CMR
 - image_L1_normdiff, 350
- nppiNormDiff_L1_16u_C3R
 - image_L1_normdiff, 350
- nppiNormDiff_L1_16u_C4R
 - image_L1_normdiff, 351
- nppiNormDiff_L1_32f_AC4R
 - image_L1_normdiff, 351
- nppiNormDiff_L1_32f_C1MR
 - image_L1_normdiff, 352
- nppiNormDiff_L1_32f_C1R
 - image_L1_normdiff, 352
- nppiNormDiff_L1_32f_C3CMR
 - image_L1_normdiff, 353
- nppiNormDiff_L1_32f_C3R
 - image_L1_normdiff, 353
- nppiNormDiff_L1_32f_C4R
 - image_L1_normdiff, 354
- nppiNormDiff_L1_8s_C1MR
 - image_L1_normdiff, 354
- nppiNormDiff_L1_8s_C3CMR
 - image_L1_normdiff, 355
- nppiNormDiff_L1_8u_AC4R
 - image_L1_normdiff, 355
- nppiNormDiff_L1_8u_C1MR
 - image_L1_normdiff, 356
- nppiNormDiff_L1_8u_C1R
 - image_L1_normdiff, 356
- nppiNormDiff_L1_8u_C3CMR
 - image_L1_normdiff, 356
- nppiNormDiff_L1_8u_C3R
 - image_L1_normdiff, 357
- nppiNormDiff_L1_8u_C4R
 - image_L1_normdiff, 357
- nppiNormDiff_L2_16s_AC4R
 - image_L2_normdiff, 370
- nppiNormDiff_L2_16s_C1R
 - image_L2_normdiff, 370
- nppiNormDiff_L2_16s_C3R
 - image_L2_normdiff, 371
- nppiNormDiff_L2_16s_C4R
 - image_L2_normdiff, 371
- nppiNormDiff_L2_16u_AC4R
 - image_L2_normdiff, 372
- nppiNormDiff_L2_16u_C1MR
 - image_L2_normdiff, 372
- nppiNormDiff_L2_16u_C1R
 - image_L2_normdiff, 372
- nppiNormDiff_L2_16u_C3CMR
 - image_L2_normdiff, 373
- nppiNormDiff_L2_16u_C3R
 - image_L2_normdiff, 373
- nppiNormDiff_L2_16u_C4R
 - image_L2_normdiff, 374
- nppiNormDiff_L2_32f_AC4R
 - image_L2_normdiff, 374
- nppiNormDiff_L2_32f_C1MR
 - image_L2_normdiff, 375
- nppiNormDiff_L2_32f_C1R
 - image_L2_normdiff, 375
- nppiNormDiff_L2_32f_C3CMR
 - image_L2_normdiff, 376
- nppiNormDiff_L2_32f_C3R
 - image_L2_normdiff, 376
- nppiNormDiff_L2_32f_C4R
 - image_L2_normdiff, 377
- nppiNormDiff_L2_8s_C1MR

- image_L2_normdiff, 377
- nppiNormDiff_L2_8s_C3CMR
 - image_L2_normdiff, 378
- nppiNormDiff_L2_8u_AC4R
 - image_L2_normdiff, 378
- nppiNormDiff_L2_8u_C1MR
 - image_L2_normdiff, 379
- nppiNormDiff_L2_8u_C1R
 - image_L2_normdiff, 379
- nppiNormDiff_L2_8u_C3CMR
 - image_L2_normdiff, 379
- nppiNormDiff_L2_8u_C3R
 - image_L2_normdiff, 380
- nppiNormDiff_L2_8u_C4R
 - image_L2_normdiff, 380
- nppiNormDiffInfGetBufferHostSize_16s_AC4R
 - image_inf_normdiff, 335
- nppiNormDiffInfGetBufferHostSize_16s_C1R
 - image_inf_normdiff, 335
- nppiNormDiffInfGetBufferHostSize_16s_C3R
 - image_inf_normdiff, 336
- nppiNormDiffInfGetBufferHostSize_16s_C4R
 - image_inf_normdiff, 336
- nppiNormDiffInfGetBufferHostSize_16u_AC4R
 - image_inf_normdiff, 336
- nppiNormDiffInfGetBufferHostSize_16u_C1MR
 - image_inf_normdiff, 337
- nppiNormDiffInfGetBufferHostSize_16u_C1R
 - image_inf_normdiff, 337
- nppiNormDiffInfGetBufferHostSize_16u_C3CMR
 - image_inf_normdiff, 337
- nppiNormDiffInfGetBufferHostSize_16u_C3R
 - image_inf_normdiff, 337
- nppiNormDiffInfGetBufferHostSize_16u_C4R
 - image_inf_normdiff, 338
- nppiNormDiffInfGetBufferHostSize_32f_AC4R
 - image_inf_normdiff, 338
- nppiNormDiffInfGetBufferHostSize_32f_C1MR
 - image_inf_normdiff, 338
- nppiNormDiffInfGetBufferHostSize_32f_C1R
 - image_inf_normdiff, 339
- nppiNormDiffInfGetBufferHostSize_32f_C3CMR
 - image_inf_normdiff, 339
- nppiNormDiffInfGetBufferHostSize_32f_C3R
 - image_inf_normdiff, 339
- nppiNormDiffInfGetBufferHostSize_32f_C4R
 - image_inf_normdiff, 339
- nppiNormDiffInfGetBufferHostSize_8s_C1MR
 - image_inf_normdiff, 340
- nppiNormDiffInfGetBufferHostSize_8s_C3CMR
 - image_inf_normdiff, 340
- nppiNormDiffInfGetBufferHostSize_8u_AC4R
 - image_inf_normdiff, 340
- nppiNormDiffInfGetBufferHostSize_8u_C1MR
 - image_inf_normdiff, 341
- nppiNormDiffInfGetBufferHostSize_8u_C1R
 - image_inf_normdiff, 341
- nppiNormDiffInfGetBufferHostSize_8u_C3CMR
 - image_inf_normdiff, 341
- nppiNormDiffInfGetBufferHostSize_8u_C3R
 - image_inf_normdiff, 341
- nppiNormDiffInfGetBufferHostSize_8u_C4R
 - image_inf_normdiff, 342
- nppiNormDiffL1GetBufferHostSize_16s_AC4R
 - image_L1_normdiff, 358
- nppiNormDiffL1GetBufferHostSize_16s_C1R
 - image_L1_normdiff, 358
- nppiNormDiffL1GetBufferHostSize_16s_C3R
 - image_L1_normdiff, 358
- nppiNormDiffL1GetBufferHostSize_16s_C4R
 - image_L1_normdiff, 359
- nppiNormDiffL1GetBufferHostSize_16u_AC4R
 - image_L1_normdiff, 359
- nppiNormDiffL1GetBufferHostSize_16u_C1MR
 - image_L1_normdiff, 359
- nppiNormDiffL1GetBufferHostSize_16u_C1R
 - image_L1_normdiff, 360
- nppiNormDiffL1GetBufferHostSize_16u_C3CMR
 - image_L1_normdiff, 360
- nppiNormDiffL1GetBufferHostSize_16u_C3R
 - image_L1_normdiff, 360
- nppiNormDiffL1GetBufferHostSize_16u_C4R
 - image_L1_normdiff, 360
- nppiNormDiffL1GetBufferHostSize_32f_AC4R
 - image_L1_normdiff, 361
- nppiNormDiffL1GetBufferHostSize_32f_C1MR
 - image_L1_normdiff, 361
- nppiNormDiffL1GetBufferHostSize_32f_C1R
 - image_L1_normdiff, 361
- nppiNormDiffL1GetBufferHostSize_32f_C3CMR
 - image_L1_normdiff, 362
- nppiNormDiffL1GetBufferHostSize_32f_C3R
 - image_L1_normdiff, 362
- nppiNormDiffL1GetBufferHostSize_32f_C4R
 - image_L1_normdiff, 362
- nppiNormDiffL1GetBufferHostSize_8s_C1MR
 - image_L1_normdiff, 362
- nppiNormDiffL1GetBufferHostSize_8s_C3CMR
 - image_L1_normdiff, 363
- nppiNormDiffL1GetBufferHostSize_8u_AC4R
 - image_L1_normdiff, 363
- nppiNormDiffL1GetBufferHostSize_8u_C1MR
 - image_L1_normdiff, 363
- nppiNormDiffL1GetBufferHostSize_8u_C1R
 - image_L1_normdiff, 364
- nppiNormDiffL1GetBufferHostSize_8u_C3CMR
 - image_L1_normdiff, 364
- nppiNormDiffL1GetBufferHostSize_8u_C3R
 - image_L1_normdiff, 364

- image_L1_normdiff, 364
- nppiNormDiffL1GetBufferHostSize_8u_C4R
 - image_L1_normdiff, 364
- nppiNormDiffL2GetBufferHostSize_16s_AC4R
 - image_L2_normdiff, 381
- nppiNormDiffL2GetBufferHostSize_16s_C1R
 - image_L2_normdiff, 381
- nppiNormDiffL2GetBufferHostSize_16s_C3R
 - image_L2_normdiff, 381
- nppiNormDiffL2GetBufferHostSize_16s_C4R
 - image_L2_normdiff, 382
- nppiNormDiffL2GetBufferHostSize_16u_AC4R
 - image_L2_normdiff, 382
- nppiNormDiffL2GetBufferHostSize_16u_C1MR
 - image_L2_normdiff, 382
- nppiNormDiffL2GetBufferHostSize_16u_C1R
 - image_L2_normdiff, 383
- nppiNormDiffL2GetBufferHostSize_16u_C3CMR
 - image_L2_normdiff, 383
- nppiNormDiffL2GetBufferHostSize_16u_C3R
 - image_L2_normdiff, 383
- nppiNormDiffL2GetBufferHostSize_16u_C4R
 - image_L2_normdiff, 383
- nppiNormDiffL2GetBufferHostSize_32f_AC4R
 - image_L2_normdiff, 384
- nppiNormDiffL2GetBufferHostSize_32f_C1MR
 - image_L2_normdiff, 384
- nppiNormDiffL2GetBufferHostSize_32f_C1R
 - image_L2_normdiff, 384
- nppiNormDiffL2GetBufferHostSize_32f_C3CMR
 - image_L2_normdiff, 385
- nppiNormDiffL2GetBufferHostSize_32f_C3R
 - image_L2_normdiff, 385
- nppiNormDiffL2GetBufferHostSize_32f_C4R
 - image_L2_normdiff, 385
- nppiNormDiffL2GetBufferHostSize_8s_C1MR
 - image_L2_normdiff, 385
- nppiNormDiffL2GetBufferHostSize_8s_C3CMR
 - image_L2_normdiff, 386
- nppiNormDiffL2GetBufferHostSize_8u_AC4R
 - image_L2_normdiff, 386
- nppiNormDiffL2GetBufferHostSize_8u_C1MR
 - image_L2_normdiff, 386
- nppiNormDiffL2GetBufferHostSize_8u_C1R
 - image_L2_normdiff, 387
- nppiNormDiffL2GetBufferHostSize_8u_C3CMR
 - image_L2_normdiff, 387
- nppiNormDiffL2GetBufferHostSize_8u_C3R
 - image_L2_normdiff, 387
- nppiNormDiffL2GetBufferHostSize_8u_C4R
 - image_L2_normdiff, 387
- nppiNormInf
 - typedefs_npp, 42
- nppiNormInfGetBufferHostSize_16s_AC4R
- image_inf_norm, 269
- nppiNormInfGetBufferHostSize_16s_C1R
 - image_inf_norm, 270
- nppiNormInfGetBufferHostSize_16s_C3R
 - image_inf_norm, 270
- nppiNormInfGetBufferHostSize_16s_C4R
 - image_inf_norm, 270
- nppiNormInfGetBufferHostSize_16u_AC4R
 - image_inf_norm, 271
- nppiNormInfGetBufferHostSize_16u_C1MR
 - image_inf_norm, 271
- nppiNormInfGetBufferHostSize_16u_C1R
 - image_inf_norm, 271
- nppiNormInfGetBufferHostSize_16u_C3CMR
 - image_inf_norm, 271
- nppiNormInfGetBufferHostSize_16u_C3R
 - image_inf_norm, 272
- nppiNormInfGetBufferHostSize_16u_C4R
 - image_inf_norm, 272
- nppiNormInfGetBufferHostSize_32f_AC4R
 - image_inf_norm, 272
- nppiNormInfGetBufferHostSize_32f_C1MR
 - image_inf_norm, 273
- nppiNormInfGetBufferHostSize_32f_C1R
 - image_inf_norm, 273
- nppiNormInfGetBufferHostSize_32f_C3CMR
 - image_inf_norm, 273
- nppiNormInfGetBufferHostSize_32f_C3R
 - image_inf_norm, 273
- nppiNormInfGetBufferHostSize_32f_C4R
 - image_inf_norm, 274
- nppiNormInfGetBufferHostSize_32s_C1R
 - image_inf_norm, 274
- nppiNormInfGetBufferHostSize_8s_C1MR
 - image_inf_norm, 274
- nppiNormInfGetBufferHostSize_8s_C3CMR
 - image_inf_norm, 275
- nppiNormInfGetBufferHostSize_8u_AC4R
 - image_inf_norm, 275
- nppiNormInfGetBufferHostSize_8u_C1MR
 - image_inf_norm, 275
- nppiNormInfGetBufferHostSize_8u_C1R
 - image_inf_norm, 275
- nppiNormInfGetBufferHostSize_8u_C3CMR
 - image_inf_norm, 276
- nppiNormInfGetBufferHostSize_8u_C3R
 - image_inf_norm, 276
- nppiNormInfGetBufferHostSize_8u_C4R
 - image_inf_norm, 276
- nppiNormL1
 - typedefs_npp, 42
- nppiNormL1GetBufferHostSize_16s_AC4R
 - image_L1_norm, 291
- nppiNormL1GetBufferHostSize_16s_C1R

- image_L1_norm, 291
- nppiNormL1GetBufferHostSize_16s_C3R
 - image_L1_norm, 291
- nppiNormL1GetBufferHostSize_16s_C4R
 - image_L1_norm, 292
- nppiNormL1GetBufferHostSize_16u_AC4R
 - image_L1_norm, 292
- nppiNormL1GetBufferHostSize_16u_C1MR
 - image_L1_norm, 292
- nppiNormL1GetBufferHostSize_16u_C1R
 - image_L1_norm, 293
- nppiNormL1GetBufferHostSize_16u_C3CMR
 - image_L1_norm, 293
- nppiNormL1GetBufferHostSize_16u_C3R
 - image_L1_norm, 293
- nppiNormL1GetBufferHostSize_16u_C4R
 - image_L1_norm, 293
- nppiNormL1GetBufferHostSize_32f_AC4R
 - image_L1_norm, 294
- nppiNormL1GetBufferHostSize_32f_C1MR
 - image_L1_norm, 294
- nppiNormL1GetBufferHostSize_32f_C1R
 - image_L1_norm, 294
- nppiNormL1GetBufferHostSize_32f_C3CMR
 - image_L1_norm, 295
- nppiNormL1GetBufferHostSize_32f_C3R
 - image_L1_norm, 295
- nppiNormL1GetBufferHostSize_32f_C4R
 - image_L1_norm, 295
- nppiNormL1GetBufferHostSize_8s_C1MR
 - image_L1_norm, 295
- nppiNormL1GetBufferHostSize_8s_C3CMR
 - image_L1_norm, 296
- nppiNormL1GetBufferHostSize_8u_AC4R
 - image_L1_norm, 296
- nppiNormL1GetBufferHostSize_8u_C1MR
 - image_L1_norm, 296
- nppiNormL1GetBufferHostSize_8u_C1R
 - image_L1_norm, 297
- nppiNormL1GetBufferHostSize_8u_C3CMR
 - image_L1_norm, 297
- nppiNormL1GetBufferHostSize_8u_C3R
 - image_L1_norm, 297
- nppiNormL1GetBufferHostSize_8u_C4R
 - image_L1_norm, 297
- nppiNormL2
 - typedefs_npp, 42
- nppiNormL2GetBufferHostSize_16s_AC4R
 - image_L2_norm, 312
- nppiNormL2GetBufferHostSize_16s_C1R
 - image_L2_norm, 312
- nppiNormL2GetBufferHostSize_16s_C3R
 - image_L2_norm, 312
- nppiNormL2GetBufferHostSize_16s_C4R
 - image_L2_norm, 313
- nppiNormL2GetBufferHostSize_16u_AC4R
 - image_L2_norm, 313
- nppiNormL2GetBufferHostSize_16u_C1MR
 - image_L2_norm, 313
- nppiNormL2GetBufferHostSize_16u_C1R
 - image_L2_norm, 314
- nppiNormL2GetBufferHostSize_16u_C3CMR
 - image_L2_norm, 314
- nppiNormL2GetBufferHostSize_16u_C3R
 - image_L2_norm, 314
- nppiNormL2GetBufferHostSize_16u_C4R
 - image_L2_norm, 314
- nppiNormL2GetBufferHostSize_32f_AC4R
 - image_L2_norm, 315
- nppiNormL2GetBufferHostSize_32f_C1MR
 - image_L2_norm, 315
- nppiNormL2GetBufferHostSize_32f_C1R
 - image_L2_norm, 315
- nppiNormL2GetBufferHostSize_32f_C3CMR
 - image_L2_norm, 316
- nppiNormL2GetBufferHostSize_32f_C3R
 - image_L2_norm, 316
- nppiNormL2GetBufferHostSize_32f_C4R
 - image_L2_norm, 316
- nppiNormL2GetBufferHostSize_8s_C1MR
 - image_L2_norm, 316
- nppiNormL2GetBufferHostSize_8s_C3CMR
 - image_L2_norm, 317
- nppiNormL2GetBufferHostSize_8u_AC4R
 - image_L2_norm, 317
- nppiNormL2GetBufferHostSize_8u_C1MR
 - image_L2_norm, 317
- nppiNormL2GetBufferHostSize_8u_C1R
 - image_L2_norm, 318
- nppiNormL2GetBufferHostSize_8u_C3CMR
 - image_L2_norm, 318
- nppiNormL2GetBufferHostSize_8u_C3R
 - image_L2_norm, 318
- nppiNormL2GetBufferHostSize_8u_C4R
 - image_L2_norm, 318
- nppiNormRel_Inf_16s_AC4R
 - image_inf_normrel, 393
- nppiNormRel_Inf_16s_C1R
 - image_inf_normrel, 393
- nppiNormRel_Inf_16s_C3R
 - image_inf_normrel, 394
- nppiNormRel_Inf_16s_C4R
 - image_inf_normrel, 394
- nppiNormRel_Inf_16u_AC4R
 - image_inf_normrel, 395
- nppiNormRel_Inf_16u_C1MR
 - image_inf_normrel, 395
- nppiNormRel_Inf_16u_C1R
 - image_inf_normrel, 395

- image_inf_normrel, 396
- nppiNormRel_Inf_16u_C3CMR
 - image_inf_normrel, 396
- nppiNormRel_Inf_16u_C3R
 - image_inf_normrel, 397
- nppiNormRel_Inf_16u_C4R
 - image_inf_normrel, 397
- nppiNormRel_Inf_32f_AC4R
 - image_inf_normrel, 397
- nppiNormRel_Inf_32f_C1MR
 - image_inf_normrel, 398
- nppiNormRel_Inf_32f_C1R
 - image_inf_normrel, 398
- nppiNormRel_Inf_32f_C3CMR
 - image_inf_normrel, 399
- nppiNormRel_Inf_32f_C3R
 - image_inf_normrel, 399
- nppiNormRel_Inf_32f_C4R
 - image_inf_normrel, 400
- nppiNormRel_Inf_8s_C1MR
 - image_inf_normrel, 400
- nppiNormRel_Inf_8s_C3CMR
 - image_inf_normrel, 401
- nppiNormRel_Inf_8u_AC4R
 - image_inf_normrel, 401
- nppiNormRel_Inf_8u_C1MR
 - image_inf_normrel, 402
- nppiNormRel_Inf_8u_C1R
 - image_inf_normrel, 402
- nppiNormRel_Inf_8u_C3CMR
 - image_inf_normrel, 403
- nppiNormRel_Inf_8u_C3R
 - image_inf_normrel, 403
- nppiNormRel_Inf_8u_C4R
 - image_inf_normrel, 404
- nppiNormRel_L1_16s_AC4R
 - image_L1_normrel, 416
- nppiNormRel_L1_16s_C1R
 - image_L1_normrel, 416
- nppiNormRel_L1_16s_C3R
 - image_L1_normrel, 417
- nppiNormRel_L1_16s_C4R
 - image_L1_normrel, 417
- nppiNormRel_L1_16u_AC4R
 - image_L1_normrel, 418
- nppiNormRel_L1_16u_C1MR
 - image_L1_normrel, 418
- nppiNormRel_L1_16u_C1R
 - image_L1_normrel, 419
- nppiNormRel_L1_16u_C3CMR
 - image_L1_normrel, 419
- nppiNormRel_L1_16u_C3R
 - image_L1_normrel, 419
- nppiNormRel_L1_16u_C4R
 - image_L1_normrel, 420
- nppiNormRel_L1_32f_AC4R
 - image_L1_normrel, 420
- nppiNormRel_L1_32f_C1MR
 - image_L1_normrel, 421
- nppiNormRel_L1_32f_C1R
 - image_L1_normrel, 421
- nppiNormRel_L1_32f_C3CMR
 - image_L1_normrel, 422
- nppiNormRel_L1_32f_C3R
 - image_L1_normrel, 422
- nppiNormRel_L1_32f_C4R
 - image_L1_normrel, 423
- nppiNormRel_L1_8s_C1MR
 - image_L1_normrel, 423
- nppiNormRel_L1_8s_C3CMR
 - image_L1_normrel, 424
- nppiNormRel_L1_8u_AC4R
 - image_L1_normrel, 424
- nppiNormRel_L1_8u_C1MR
 - image_L1_normrel, 425
- nppiNormRel_L1_8u_C1R
 - image_L1_normrel, 425
- nppiNormRel_L1_8u_C3CMR
 - image_L1_normrel, 426
- nppiNormRel_L1_8u_C3R
 - image_L1_normrel, 426
- nppiNormRel_L1_8u_C4R
 - image_L1_normrel, 427
- nppiNormRel_L2_16s_AC4R
 - image_L2_normrel, 439
- nppiNormRel_L2_16s_C1R
 - image_L2_normrel, 439
- nppiNormRel_L2_16s_C3R
 - image_L2_normrel, 440
- nppiNormRel_L2_16s_C4R
 - image_L2_normrel, 440
- nppiNormRel_L2_16u_AC4R
 - image_L2_normrel, 441
- nppiNormRel_L2_16u_C1MR
 - image_L2_normrel, 441
- nppiNormRel_L2_16u_C1R
 - image_L2_normrel, 442
- nppiNormRel_L2_16u_C3CMR
 - image_L2_normrel, 442
- nppiNormRel_L2_16u_C3R
 - image_L2_normrel, 442
- nppiNormRel_L2_16u_C4R
 - image_L2_normrel, 443
- nppiNormRel_L2_32f_AC4R
 - image_L2_normrel, 443
- nppiNormRel_L2_32f_C1MR
 - image_L2_normrel, 444
- nppiNormRel_L2_32f_C1R

- image_L2_normrel, 444
- nppiNormRel_L2_32f_C3CMR
 - image_L2_normrel, 445
- nppiNormRel_L2_32f_C3R
 - image_L2_normrel, 445
- nppiNormRel_L2_32f_C4R
 - image_L2_normrel, 446
- nppiNormRel_L2_8s_C1MR
 - image_L2_normrel, 446
- nppiNormRel_L2_8s_C3CMR
 - image_L2_normrel, 447
- nppiNormRel_L2_8u_AC4R
 - image_L2_normrel, 447
- nppiNormRel_L2_8u_C1MR
 - image_L2_normrel, 448
- nppiNormRel_L2_8u_C1R
 - image_L2_normrel, 448
- nppiNormRel_L2_8u_C3CMR
 - image_L2_normrel, 449
- nppiNormRel_L2_8u_C3R
 - image_L2_normrel, 449
- nppiNormRel_L2_8u_C4R
 - image_L2_normrel, 450
- nppiNormRelInfGetBufferHostSize_16s_AC4R
 - image_inf_normrel, 404
- nppiNormRelInfGetBufferHostSize_16s_C1R
 - image_inf_normrel, 405
- nppiNormRelInfGetBufferHostSize_16s_C3R
 - image_inf_normrel, 405
- nppiNormRelInfGetBufferHostSize_16s_C4R
 - image_inf_normrel, 405
- nppiNormRelInfGetBufferHostSize_16u_AC4R
 - image_inf_normrel, 405
- nppiNormRelInfGetBufferHostSize_16u_C1MR
 - image_inf_normrel, 406
- nppiNormRelInfGetBufferHostSize_16u_C1R
 - image_inf_normrel, 406
- nppiNormRelInfGetBufferHostSize_16u_C3CMR
 - image_inf_normrel, 406
- nppiNormRelInfGetBufferHostSize_16u_C3R
 - image_inf_normrel, 407
- nppiNormRelInfGetBufferHostSize_16u_C4R
 - image_inf_normrel, 407
- nppiNormRelInfGetBufferHostSize_32f_AC4R
 - image_inf_normrel, 407
- nppiNormRelInfGetBufferHostSize_32f_C1MR
 - image_inf_normrel, 407
- nppiNormRelInfGetBufferHostSize_32f_C1R
 - image_inf_normrel, 408
- nppiNormRelInfGetBufferHostSize_32f_C3CMR
 - image_inf_normrel, 408
- nppiNormRelInfGetBufferHostSize_32f_C3R
 - image_inf_normrel, 408
- nppiNormRelInfGetBufferHostSize_32f_C4R
 - image_inf_normrel, 408
- image_inf_normrel, 409
- nppiNormRelInfGetBufferHostSize_32s_C1R
 - image_inf_normrel, 409
- nppiNormRelInfGetBufferHostSize_8s_C1MR
 - image_inf_normrel, 409
- nppiNormRelInfGetBufferHostSize_8s_C3CMR
 - image_inf_normrel, 409
- nppiNormRelInfGetBufferHostSize_8u_AC4R
 - image_inf_normrel, 410
- nppiNormRelInfGetBufferHostSize_8u_C1MR
 - image_inf_normrel, 410
- nppiNormRelInfGetBufferHostSize_8u_C1R
 - image_inf_normrel, 410
- nppiNormRelInfGetBufferHostSize_8u_C3CMR
 - image_inf_normrel, 411
- nppiNormRelInfGetBufferHostSize_8u_C3R
 - image_inf_normrel, 411
- nppiNormRelInfGetBufferHostSize_8u_C4R
 - image_inf_normrel, 411
- nppiNormRelL1GetBufferHostSize_16s_AC4R
 - image_L1_normrel, 427
- nppiNormRelL1GetBufferHostSize_16s_C1R
 - image_L1_normrel, 427
- nppiNormRelL1GetBufferHostSize_16s_C3R
 - image_L1_normrel, 428
- nppiNormRelL1GetBufferHostSize_16s_C4R
 - image_L1_normrel, 428
- nppiNormRelL1GetBufferHostSize_16u_AC4R
 - image_L1_normrel, 428
- nppiNormRelL1GetBufferHostSize_16u_C1MR
 - image_L1_normrel, 429
- nppiNormRelL1GetBufferHostSize_16u_C1R
 - image_L1_normrel, 429
- nppiNormRelL1GetBufferHostSize_16u_C3CMR
 - image_L1_normrel, 429
- nppiNormRelL1GetBufferHostSize_16u_C3R
 - image_L1_normrel, 429
- nppiNormRelL1GetBufferHostSize_16u_C4R
 - image_L1_normrel, 430
- nppiNormRelL1GetBufferHostSize_32f_AC4R
 - image_L1_normrel, 430
- nppiNormRelL1GetBufferHostSize_32f_C1MR
 - image_L1_normrel, 430
- nppiNormRelL1GetBufferHostSize_32f_C1R
 - image_L1_normrel, 431
- nppiNormRelL1GetBufferHostSize_32f_C3CMR
 - image_L1_normrel, 431
- nppiNormRelL1GetBufferHostSize_32f_C3R
 - image_L1_normrel, 431
- nppiNormRelL1GetBufferHostSize_32f_C4R
 - image_L1_normrel, 431
- nppiNormRelL1GetBufferHostSize_8s_C1MR
 - image_L1_normrel, 432
- nppiNormRelL1GetBufferHostSize_8s_C3CMR

- image_L1_normrel, 432
- nppiNormRelL1GetBufferHostSize_8u_AC4R
 - image_L1_normrel, 432
- nppiNormRelL1GetBufferHostSize_8u_C1MR
 - image_L1_normrel, 433
- nppiNormRelL1GetBufferHostSize_8u_C1R
 - image_L1_normrel, 433
- nppiNormRelL1GetBufferHostSize_8u_C3CMR
 - image_L1_normrel, 433
- nppiNormRelL1GetBufferHostSize_8u_C3R
 - image_L1_normrel, 433
- nppiNormRelL1GetBufferHostSize_8u_C4R
 - image_L1_normrel, 434
- nppiNormRelL2GetBufferHostSize_16s_AC4R
 - image_L2_normrel, 450
- nppiNormRelL2GetBufferHostSize_16s_C1R
 - image_L2_normrel, 450
- nppiNormRelL2GetBufferHostSize_16s_C3R
 - image_L2_normrel, 451
- nppiNormRelL2GetBufferHostSize_16s_C4R
 - image_L2_normrel, 451
- nppiNormRelL2GetBufferHostSize_16u_AC4R
 - image_L2_normrel, 451
- nppiNormRelL2GetBufferHostSize_16u_C1MR
 - image_L2_normrel, 452
- nppiNormRelL2GetBufferHostSize_16u_C1R
 - image_L2_normrel, 452
- nppiNormRelL2GetBufferHostSize_16u_C3CMR
 - image_L2_normrel, 452
- nppiNormRelL2GetBufferHostSize_16u_C3R
 - image_L2_normrel, 452
- nppiNormRelL2GetBufferHostSize_16u_C4R
 - image_L2_normrel, 453
- nppiNormRelL2GetBufferHostSize_32f_AC4R
 - image_L2_normrel, 453
- nppiNormRelL2GetBufferHostSize_32f_C1MR
 - image_L2_normrel, 453
- nppiNormRelL2GetBufferHostSize_32f_C1R
 - image_L2_normrel, 454
- nppiNormRelL2GetBufferHostSize_32f_C3CMR
 - image_L2_normrel, 454
- nppiNormRelL2GetBufferHostSize_32f_C3R
 - image_L2_normrel, 454
- nppiNormRelL2GetBufferHostSize_32f_C4R
 - image_L2_normrel, 454
- nppiNormRelL2GetBufferHostSize_8s_C1MR
 - image_L2_normrel, 455
- nppiNormRelL2GetBufferHostSize_8s_C3CMR
 - image_L2_normrel, 455
- nppiNormRelL2GetBufferHostSize_8u_AC4R
 - image_L2_normrel, 455
- nppiNormRelL2GetBufferHostSize_8u_C1MR
 - image_L2_normrel, 456
- nppiNormRelL2GetBufferHostSize_8u_C1R
 - image_L2_normrel, 456
- nppiNormRelL2GetBufferHostSize_8u_C3CMR
 - image_L2_normrel, 456
- nppiNormRelL2GetBufferHostSize_8u_C3R
 - image_L2_normrel, 456
- nppiNormRelL2GetBufferHostSize_8u_C4R
 - image_L2_normrel, 457
- NppiPoint, 785
 - x, 785
 - y, 785
- nppiQualityIndex_16u32f_AC4R
 - image_quality_index, 674
- nppiQualityIndex_16u32f_C1R
 - image_quality_index, 674
- nppiQualityIndex_16u32f_C3R
 - image_quality_index, 675
- nppiQualityIndex_32f_AC4R
 - image_quality_index, 675
- nppiQualityIndex_32f_C1R
 - image_quality_index, 676
- nppiQualityIndex_32f_C3R
 - image_quality_index, 676
- nppiQualityIndex_8u32f_AC4R
 - image_quality_index, 676
- nppiQualityIndex_8u32f_C1R
 - image_quality_index, 677
- nppiQualityIndex_8u32f_C3R
 - image_quality_index, 677
- nppiQualityIndexGetBufferHostSize_16u32f_-AC4R
 - image_quality_index, 678
- nppiQualityIndexGetBufferHostSize_16u32f_C1R
 - image_quality_index, 678
- nppiQualityIndexGetBufferHostSize_16u32f_C3R
 - image_quality_index, 678
- nppiQualityIndexGetBufferHostSize_32f_AC4R
 - image_quality_index, 679
- nppiQualityIndexGetBufferHostSize_32f_C1R
 - image_quality_index, 679
- nppiQualityIndexGetBufferHostSize_32f_C3R
 - image_quality_index, 679
- nppiQualityIndexGetBufferHostSize_8u32f_AC4R
 - image_quality_index, 680
- nppiQualityIndexGetBufferHostSize_8u32f_C1R
 - image_quality_index, 680
- nppiQualityIndexGetBufferHostSize_8u32f_C3R
 - image_quality_index, 680
- NppiRect, 786
 - height, 786
 - width, 786
 - x, 786
 - y, 786
- nppiRectStdDev_32f_C1R
 - image_rectstddev, 508

- nppiRectStdDev_32s32f_C1R
image_rectstddev, 509
- nppiRectStdDev_32s_C1RSfs
image_rectstddev, 509
- nppiSameNormLevelGetBufferHostSize_16u32f_-
AC4R
crosscorrsmenormlevel, 645
- nppiSameNormLevelGetBufferHostSize_16u32f_-
C1R
crosscorrsmenormlevel, 646
- nppiSameNormLevelGetBufferHostSize_16u32f_-
C3R
crosscorrsmenormlevel, 646
- nppiSameNormLevelGetBufferHostSize_16u32f_-
C4R
crosscorrsmenormlevel, 646
- nppiSameNormLevelGetBufferHostSize_32f_-
AC4R
crosscorrsmenormlevel, 647
- nppiSameNormLevelGetBufferHostSize_32f_C1R
crosscorrsmenormlevel, 647
- nppiSameNormLevelGetBufferHostSize_32f_C3R
crosscorrsmenormlevel, 647
- nppiSameNormLevelGetBufferHostSize_32f_C4R
crosscorrsmenormlevel, 647
- nppiSameNormLevelGetBufferHostSize_8s32f_-
AC4R
crosscorrsmenormlevel, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_-
C1R
crosscorrsmenormlevel, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_-
C3R
crosscorrsmenormlevel, 648
- nppiSameNormLevelGetBufferHostSize_8s32f_-
C4R
crosscorrsmenormlevel, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_-
AC4R
crosscorrsmenormlevel, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_-
C1R
crosscorrsmenormlevel, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_-
C3R
crosscorrsmenormlevel, 649
- nppiSameNormLevelGetBufferHostSize_8u32f_-
C4R
crosscorrsmenormlevel, 650
- nppiSameNormLevelGetBufferHostSize_8u_-
AC4RSfs
crosscorrsmenormlevel, 650
- nppiSameNormLevelGetBufferHostSize_8u_-
C1RSfs
crosscorrsmenormlevel, 650
- nppiSameNormLevelGetBufferHostSize_8u_-
C3RSfs
crosscorrsmenormlevel, 651
- nppiSameNormLevelGetBufferHostSize_8u_-
C4RSfs
crosscorrsmenormlevel, 651
- NppiSize, 787
height, 787
width, 787
- nppiSqrDistanceFull_Norm_16u32f_AC4R
sqrdistancefullnorm, 545
- nppiSqrDistanceFull_Norm_16u32f_C1R
sqrdistancefullnorm, 545
- nppiSqrDistanceFull_Norm_16u32f_C3R
sqrdistancefullnorm, 545
- nppiSqrDistanceFull_Norm_16u32f_C4R
sqrdistancefullnorm, 546
- nppiSqrDistanceFull_Norm_32f_AC4R
sqrdistancefullnorm, 546
- nppiSqrDistanceFull_Norm_32f_C1R
sqrdistancefullnorm, 547
- nppiSqrDistanceFull_Norm_32f_C3R
sqrdistancefullnorm, 547
- nppiSqrDistanceFull_Norm_32f_C4R
sqrdistancefullnorm, 548
- nppiSqrDistanceFull_Norm_8s32f_AC4R
sqrdistancefullnorm, 548
- nppiSqrDistanceFull_Norm_8s32f_C1R
sqrdistancefullnorm, 548
- nppiSqrDistanceFull_Norm_8s32f_C3R
sqrdistancefullnorm, 549
- nppiSqrDistanceFull_Norm_8s32f_C4R
sqrdistancefullnorm, 549
- nppiSqrDistanceFull_Norm_8u32f_AC4R
sqrdistancefullnorm, 550
- nppiSqrDistanceFull_Norm_8u32f_C1R
sqrdistancefullnorm, 550
- nppiSqrDistanceFull_Norm_8u32f_C3R
sqrdistancefullnorm, 551
- nppiSqrDistanceFull_Norm_8u32f_C4R
sqrdistancefullnorm, 551
- nppiSqrDistanceFull_Norm_8u_AC4RSfs
sqrdistancefullnorm, 551
- nppiSqrDistanceFull_Norm_8u_C1RSfs
sqrdistancefullnorm, 552
- nppiSqrDistanceFull_Norm_8u_C3RSfs
sqrdistancefullnorm, 552
- nppiSqrDistanceFull_Norm_8u_C4RSfs
sqrdistancefullnorm, 553
- nppiSqrDistanceSame_Norm_16u32f_AC4R
sqrdistancesamenorm, 556
- nppiSqrDistanceSame_Norm_16u32f_C1R
sqrdistancesamenorm, 556

- nppiSqrDistanceSame_Norm_16u32f_C3R
 sqrdistancesamenorm, [557](#)
 nppiSqrDistanceSame_Norm_16u32f_C4R
 sqrdistancesamenorm, [557](#)
 nppiSqrDistanceSame_Norm_32f_AC4R
 sqrdistancesamenorm, [557](#)
 nppiSqrDistanceSame_Norm_32f_C1R
 sqrdistancesamenorm, [558](#)
 nppiSqrDistanceSame_Norm_32f_C3R
 sqrdistancesamenorm, [558](#)
 nppiSqrDistanceSame_Norm_32f_C4R
 sqrdistancesamenorm, [559](#)
 nppiSqrDistanceSame_Norm_8s32f_AC4R
 sqrdistancesamenorm, [559](#)
 nppiSqrDistanceSame_Norm_8s32f_C1R
 sqrdistancesamenorm, [560](#)
 nppiSqrDistanceSame_Norm_8s32f_C3R
 sqrdistancesamenorm, [560](#)
 nppiSqrDistanceSame_Norm_8s32f_C4R
 sqrdistancesamenorm, [560](#)
 nppiSqrDistanceSame_Norm_8u32f_AC4R
 sqrdistancesamenorm, [561](#)
 nppiSqrDistanceSame_Norm_8u32f_C1R
 sqrdistancesamenorm, [561](#)
 nppiSqrDistanceSame_Norm_8u32f_C3R
 sqrdistancesamenorm, [562](#)
 nppiSqrDistanceSame_Norm_8u32f_C4R
 sqrdistancesamenorm, [562](#)
 nppiSqrDistanceSame_Norm_8u_AC4RSfs
 sqrdistancesamenorm, [563](#)
 nppiSqrDistanceSame_Norm_8u_C1RSfs
 sqrdistancesamenorm, [563](#)
 nppiSqrDistanceSame_Norm_8u_C3RSfs
 sqrdistancesamenorm, [564](#)
 nppiSqrDistanceSame_Norm_8u_C4RSfs
 sqrdistancesamenorm, [564](#)
 nppiSqrDistanceValid_Norm_16u32f_AC4R
 sqrdistancevalidnorm, [567](#)
 nppiSqrDistanceValid_Norm_16u32f_C1R
 sqrdistancevalidnorm, [567](#)
 nppiSqrDistanceValid_Norm_16u32f_C3R
 sqrdistancevalidnorm, [568](#)
 nppiSqrDistanceValid_Norm_16u32f_C4R
 sqrdistancevalidnorm, [568](#)
 nppiSqrDistanceValid_Norm_32f_AC4R
 sqrdistancevalidnorm, [568](#)
 nppiSqrDistanceValid_Norm_32f_C1R
 sqrdistancevalidnorm, [569](#)
 nppiSqrDistanceValid_Norm_32f_C3R
 sqrdistancevalidnorm, [569](#)
 nppiSqrDistanceValid_Norm_32f_C4R
 sqrdistancevalidnorm, [570](#)
 nppiSqrDistanceValid_Norm_8s32f_AC4R
 sqrdistancevalidnorm, [570](#)
 nppiSqrDistanceValid_Norm_8s32f_C1R
 sqrdistancevalidnorm, [571](#)
 nppiSqrDistanceValid_Norm_8s32f_C3R
 sqrdistancevalidnorm, [571](#)
 nppiSqrDistanceValid_Norm_8s32f_C4R
 sqrdistancevalidnorm, [571](#)
 nppiSqrDistanceValid_Norm_8u32f_AC4R
 sqrdistancevalidnorm, [572](#)
 nppiSqrDistanceValid_Norm_8u32f_C1R
 sqrdistancevalidnorm, [572](#)
 nppiSqrDistanceValid_Norm_8u32f_C3R
 sqrdistancevalidnorm, [573](#)
 nppiSqrDistanceValid_Norm_8u32f_C4R
 sqrdistancevalidnorm, [573](#)
 nppiSqrDistanceValid_Norm_8u_AC4RSfs
 sqrdistancevalidnorm, [574](#)
 nppiSqrDistanceValid_Norm_8u_C1RSfs
 sqrdistancevalidnorm, [574](#)
 nppiSqrDistanceValid_Norm_8u_C3RSfs
 sqrdistancevalidnorm, [575](#)
 nppiSqrDistanceValid_Norm_8u_C4RSfs
 sqrdistancevalidnorm, [575](#)
 nppiSqrIntegral_8u32f64f_C1R
 image_sqrintegral, [505](#)
 nppiSqrIntegral_8u32s64f_C1R
 image_sqrintegral, [506](#)
 nppiSqrIntegral_8u32s_C1R
 image_sqrintegral, [506](#)
 nppiSum_16s_AC4R
 image_sum, [120](#)
 nppiSum_16s_C1R
 image_sum, [120](#)
 nppiSum_16s_C3R
 image_sum, [120](#)
 nppiSum_16s_C4R
 image_sum, [121](#)
 nppiSum_16u_AC4R
 image_sum, [121](#)
 nppiSum_16u_C1R
 image_sum, [121](#)
 nppiSum_16u_C3R
 image_sum, [122](#)
 nppiSum_16u_C4R
 image_sum, [122](#)
 nppiSum_32f_AC4R
 image_sum, [122](#)
 nppiSum_32f_C1R
 image_sum, [123](#)
 nppiSum_32f_C3R
 image_sum, [123](#)
 nppiSum_32f_C4R
 image_sum, [123](#)
 nppiSum_8u64s_C1R
 image_sum, [124](#)

- nppiSum_8u64s_C4R
 - image_sum, [124](#)
- nppiSum_8u_AC4R
 - image_sum, [125](#)
- nppiSum_8u_C1R
 - image_sum, [125](#)
- nppiSum_8u_C3R
 - image_sum, [125](#)
- nppiSum_8u_C4R
 - image_sum, [126](#)
- nppiSumGetBufferHostSize_16s_AC4R
 - image_sum, [126](#)
- nppiSumGetBufferHostSize_16s_C1R
 - image_sum, [126](#)
- nppiSumGetBufferHostSize_16s_C3R
 - image_sum, [127](#)
- nppiSumGetBufferHostSize_16s_C4R
 - image_sum, [127](#)
- nppiSumGetBufferHostSize_16u_AC4R
 - image_sum, [127](#)
- nppiSumGetBufferHostSize_16u_C1R
 - image_sum, [127](#)
- nppiSumGetBufferHostSize_16u_C3R
 - image_sum, [128](#)
- nppiSumGetBufferHostSize_16u_C4R
 - image_sum, [128](#)
- nppiSumGetBufferHostSize_32f_AC4R
 - image_sum, [128](#)
- nppiSumGetBufferHostSize_32f_C1R
 - image_sum, [129](#)
- nppiSumGetBufferHostSize_32f_C3R
 - image_sum, [129](#)
- nppiSumGetBufferHostSize_32f_C4R
 - image_sum, [129](#)
- nppiSumGetBufferHostSize_8u64s_C1R
 - image_sum, [129](#)
- nppiSumGetBufferHostSize_8u64s_C4R
 - image_sum, [130](#)
- nppiSumGetBufferHostSize_8u_AC4R
 - image_sum, [130](#)
- nppiSumGetBufferHostSize_8u_C1R
 - image_sum, [130](#)
- nppiSumGetBufferHostSize_8u_C3R
 - image_sum, [131](#)
- nppiSumGetBufferHostSize_8u_C4R
 - image_sum, [131](#)
- nppiValidNormLevelGetBufferHostSize_16u32f_-AC4R
 - crosscorrvalidnormlevel, [665](#)
- nppiValidNormLevelGetBufferHostSize_16u32f_-C1R
 - crosscorrvalidnormlevel, [666](#)
- nppiValidNormLevelGetBufferHostSize_16u32f_-C3R
 - crosscorrvalidnormlevel, [666](#)
- nppiValidNormLevelGetBufferHostSize_16u32f_-C4R
 - crosscorrvalidnormlevel, [666](#)
- nppiValidNormLevelGetBufferHostSize_32f_-AC4R
 - crosscorrvalidnormlevel, [667](#)
- nppiValidNormLevelGetBufferHostSize_32f_C1R
 - crosscorrvalidnormlevel, [667](#)
- nppiValidNormLevelGetBufferHostSize_32f_C3R
 - crosscorrvalidnormlevel, [667](#)
- nppiValidNormLevelGetBufferHostSize_32f_C4R
 - crosscorrvalidnormlevel, [667](#)
- nppiValidNormLevelGetBufferHostSize_8s32f_-AC4R
 - crosscorrvalidnormlevel, [668](#)
- nppiValidNormLevelGetBufferHostSize_8s32f_-C1R
 - crosscorrvalidnormlevel, [668](#)
- nppiValidNormLevelGetBufferHostSize_8s32f_-C3R
 - crosscorrvalidnormlevel, [668](#)
- nppiValidNormLevelGetBufferHostSize_8s32f_-C4R
 - crosscorrvalidnormlevel, [669](#)
- nppiValidNormLevelGetBufferHostSize_8u32f_-AC4R
 - crosscorrvalidnormlevel, [669](#)
- nppiValidNormLevelGetBufferHostSize_8u32f_-C1R
 - crosscorrvalidnormlevel, [669](#)
- nppiValidNormLevelGetBufferHostSize_8u32f_-C3R
 - crosscorrvalidnormlevel, [669](#)
- nppiValidNormLevelGetBufferHostSize_8u32f_-C4R
 - crosscorrvalidnormlevel, [670](#)
- nppiValidNormLevelGetBufferHostSize_8u_-AC4RSfs
 - crosscorrvalidnormlevel, [670](#)
- nppiValidNormLevelGetBufferHostSize_8u_-C1RSfs
 - crosscorrvalidnormlevel, [670](#)
- nppiValidNormLevelGetBufferHostSize_8u_-C3RSfs
 - crosscorrvalidnormlevel, [671](#)
- nppiValidNormLevelGetBufferHostSize_8u_-C4RSfs
 - crosscorrvalidnormlevel, [671](#)
- NppLibraryVersion, [788](#)
 - build, [788](#)
 - major, [788](#)
 - minor, [788](#)
- NppRoundMode

- typedefs_npp, 42
- nppSetStream
 - core_npp, 29
- NppStatus
 - typedefs_npp, 42
- NppsZCType
 - typedefs_npp, 44
- nppZCC
 - typedefs_npp, 45
- nppZCR
 - typedefs_npp, 45
- nppZCxor
 - typedefs_npp, 45
- numClassifiers
 - NppiHaarClassifier_32f, 784
- re
 - NPP_ALIGN_16, 780
 - NPP_ALIGN_8, 781, 782
- RectStdDev, 508
- SqrDistanceFull_Norm, 543
- sqrdistancefullnorm
 - nppiSqrDistanceFull_Norm_16u32f_AC4R, 545
 - nppiSqrDistanceFull_Norm_16u32f_C1R, 545
 - nppiSqrDistanceFull_Norm_16u32f_C3R, 545
 - nppiSqrDistanceFull_Norm_16u32f_C4R, 546
 - nppiSqrDistanceFull_Norm_32f_AC4R, 546
 - nppiSqrDistanceFull_Norm_32f_C1R, 547
 - nppiSqrDistanceFull_Norm_32f_C3R, 547
 - nppiSqrDistanceFull_Norm_32f_C4R, 548
 - nppiSqrDistanceFull_Norm_8s32f_AC4R, 548
 - nppiSqrDistanceFull_Norm_8s32f_C1R, 548
 - nppiSqrDistanceFull_Norm_8s32f_C3R, 549
 - nppiSqrDistanceFull_Norm_8s32f_C4R, 549
 - nppiSqrDistanceFull_Norm_8u32f_AC4R, 550
 - nppiSqrDistanceFull_Norm_8u32f_C1R, 550
 - nppiSqrDistanceFull_Norm_8u32f_C3R, 551
 - nppiSqrDistanceFull_Norm_8u32f_C4R, 551
 - nppiSqrDistanceFull_Norm_8u_AC4RSfs, 551
 - nppiSqrDistanceFull_Norm_8u_C1RSfs, 552
 - nppiSqrDistanceFull_Norm_8u_C3RSfs, 552
 - nppiSqrDistanceFull_Norm_8u_C4RSfs, 553
- SqrDistanceSame_Norm, 554
- sqrdistancesamenorm
 - nppiSqrDistanceSame_Norm_16u32f_AC4R, 556
 - nppiSqrDistanceSame_Norm_16u32f_C1R, 556
 - nppiSqrDistanceSame_Norm_16u32f_C3R, 557
 - nppiSqrDistanceSame_Norm_16u32f_C4R, 557
 - nppiSqrDistanceSame_Norm_32f_AC4R, 557
 - nppiSqrDistanceSame_Norm_32f_C1R, 558
 - nppiSqrDistanceSame_Norm_32f_C3R, 558
 - nppiSqrDistanceSame_Norm_32f_C4R, 559
 - nppiSqrDistanceSame_Norm_8s32f_AC4R, 559
 - nppiSqrDistanceSame_Norm_8s32f_C1R, 560
 - nppiSqrDistanceSame_Norm_8s32f_C3R, 560
 - nppiSqrDistanceSame_Norm_8s32f_C4R, 560
 - nppiSqrDistanceSame_Norm_8u32f_AC4R, 561
 - nppiSqrDistanceSame_Norm_8u32f_C1R, 561
 - nppiSqrDistanceSame_Norm_8u32f_C3R, 562
 - nppiSqrDistanceSame_Norm_8u32f_C4R, 562
 - nppiSqrDistanceSame_Norm_8u_AC4RSfs, 563
 - nppiSqrDistanceSame_Norm_8u_C1RSfs, 563
 - nppiSqrDistanceSame_Norm_8u_C3RSfs, 564
 - nppiSqrDistanceSame_Norm_8u_C4RSfs, 564
- SqrDistanceValid_Norm, 565
- sqrdistancevalidnorm
 - nppiSqrDistanceValid_Norm_16u32f_AC4R, 567
 - nppiSqrDistanceValid_Norm_16u32f_C1R, 567
 - nppiSqrDistanceValid_Norm_16u32f_C3R, 568
 - nppiSqrDistanceValid_Norm_16u32f_C4R, 568
 - nppiSqrDistanceValid_Norm_32f_AC4R, 568
 - nppiSqrDistanceValid_Norm_32f_C1R, 569
 - nppiSqrDistanceValid_Norm_32f_C3R, 569
 - nppiSqrDistanceValid_Norm_32f_C4R, 570
 - nppiSqrDistanceValid_Norm_8s32f_AC4R, 570
 - nppiSqrDistanceValid_Norm_8s32f_C1R, 571
 - nppiSqrDistanceValid_Norm_8s32f_C3R, 571
 - nppiSqrDistanceValid_Norm_8s32f_C4R, 571
 - nppiSqrDistanceValid_Norm_8u32f_AC4R, 572
 - nppiSqrDistanceValid_Norm_8u32f_C1R, 572
 - nppiSqrDistanceValid_Norm_8u32f_C3R, 573
 - nppiSqrDistanceValid_Norm_8u32f_C4R, 573

- nppiSqrDistanceValid_Norm_8u_AC4RSfs, 574
- nppiSqrDistanceValid_Norm_8u_C1RSfs, 574
- nppiSqrDistanceValid_Norm_8u_C3RSfs, 575
- nppiSqrDistanceValid_Norm_8u_C4RSfs, 575
- SqrIntegral, 505
- Statistical Operations, 50
- Sum, 117
- typedefs_npp
 - NPP_AFFINE_QUAD_INCORRECT_WARNING, 44
 - NPP_ALG_HINT_ACCURATE, 39
 - NPP_ALG_HINT_FAST, 39
 - NPP_ALG_HINT_NONE, 39
 - NPP_ALIGNMENT_ERROR, 43
 - NPP_ANCHOR_ERROR, 43
 - NPP_BAD_ARGUMENT_ERROR, 44
 - NPP_BORDER_CONSTANT, 40
 - NPP_BORDER_MIRROR, 40
 - NPP_BORDER_NONE, 40
 - NPP_BORDER_REPLICATE, 40
 - NPP_BORDER_UNDEFINED, 40
 - NPP_BORDER_WRAP, 40
 - NPP_BOTH_AXIS, 40
 - NPP_CHANNEL_ERROR, 43
 - NPP_CHANNEL_ORDER_ERROR, 43
 - NPP_CMP_EQ, 39
 - NPP_CMP_GREATER, 39
 - NPP_CMP_GREATER_EQ, 39
 - NPP_CMP_LESS, 38
 - NPP_CMP_LESS_EQ, 38
 - NPP_COEFFICIENT_ERROR, 43
 - NPP_COI_ERROR, 43
 - NPP_CONTEXT_MATCH_ERROR, 44
 - NPP_CORRUPTED_DATA_ERROR, 43
 - NPP_CUDA_1_0, 39
 - NPP_CUDA_1_1, 39
 - NPP_CUDA_1_2, 39
 - NPP_CUDA_1_3, 39
 - NPP_CUDA_2_0, 39
 - NPP_CUDA_2_1, 39
 - NPP_CUDA_3_0, 39
 - NPP_CUDA_3_2, 39
 - NPP_CUDA_3_5, 39
 - NPP_CUDA_3_7, 39
 - NPP_CUDA_5_0, 39
 - NPP_CUDA_5_2, 39
 - NPP_CUDA_5_3, 39
 - NPP_CUDA_6_0, 39
 - NPP_CUDA_KERNEL_EXECUTION_ERROR, 43
 - NPP_CUDA_NOT_CAPABLE, 39
 - NPP_CUDA_UNKNOWN_VERSION, 39
 - NPP_DATA_TYPE_ERROR, 44
 - NPP_DIVIDE_BY_ZERO_ERROR, 44
 - NPP_DIVIDE_BY_ZERO_WARNING, 44
 - NPP_DIVISOR_ERROR, 43
 - NPP_DOUBLE_SIZE_WARNING, 44
 - NPP_ERROR, 44
 - NPP_ERROR_RESERVED, 44
 - NPP_FFT_FLAG_ERROR, 44
 - NPP_FFT_ORDER_ERROR, 44
 - NPP_FILTER_SCHARR, 40
 - NPP_FILTER_SOBEL, 40
 - NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR, 43
 - NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR, 43
 - NPP_HORIZONTAL_AXIS, 40
 - NPP_INTERPOLATION_ERROR, 44
 - NPP_INVALID_DEVICE_POINTER_ERROR, 43
 - NPP_INVALID_HOST_POINTER_ERROR, 43
 - NPP_LUT_NUMBER_OF_LEVELS_ERROR, 43
 - NPP_LUT_PALETTE_BITSIZE_ERROR, 43
 - NPP_MASK_SIZE_11_X_11, 41
 - NPP_MASK_SIZE_13_X_13, 41
 - NPP_MASK_SIZE_15_X_15, 41
 - NPP_MASK_SIZE_1_X_3, 41
 - NPP_MASK_SIZE_1_X_5, 41
 - NPP_MASK_SIZE_3_X_1, 41
 - NPP_MASK_SIZE_3_X_3, 41
 - NPP_MASK_SIZE_5_X_1, 41
 - NPP_MASK_SIZE_5_X_5, 41
 - NPP_MASK_SIZE_7_X_7, 41
 - NPP_MASK_SIZE_9_X_9, 41
 - NPP_MASK_SIZE_ERROR, 43
 - NPP_MEMCPY_ERROR, 43
 - NPP_MEMFREE_ERROR, 43
 - NPP_MEMORY_ALLOCATION_ERR, 44
 - NPP_MEMSET_ERROR, 43
 - NPP_MIRROR_FLIP_ERROR, 44
 - NPP_MISALIGNED_DST_ROI_WARNING, 44
 - NPP_MOMENT_00_ZERO_ERROR, 44
 - NPP_NO_ERROR, 44
 - NPP_NO_MEMORY_ERROR, 44
 - NPP_NO_OPERATION_WARNING, 44
 - NPP_NOT_EVEN_STEP_ERROR, 43
 - NPP_NOT_IMPLEMENTED_ERROR, 44
 - NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY, 43
 - NPP_NOT_SUPPORTED_MODE_ERROR, 43
 - NPP_NULL_POINTER_ERROR, 44

- NPP_NUMBER_OF_CHANNELS_ERROR, 43
- NPP_OUT_OFF_RANGE_ERROR, 44
- NPP_OVERFLOW_ERROR, 43
- NPP_QUADRANGLE_ERROR, 43
- NPP_QUALITY_INDEX_ERROR, 43
- NPP_RANGE_ERROR, 44
- NPP_RECTANGLE_ERROR, 43
- NPP_RESIZE_FACTOR_ERROR, 44
- NPP_RESIZE_NO_OPERATION_ERROR, 43
- NPP_RND_FINANCIAL, 42
- NPP_RND_NEAR, 42
- NPP_RND_ZERO, 42
- NPP_ROUND_MODE_NOT_SUPPORTED_ERROR, 43
- NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO, 42
- NPP_ROUND_NEAREST_TIES_TO_EVEN, 42
- NPP_ROUND_TOWARD_ZERO, 42
- NPP_SCALE_RANGE_ERROR, 44
- NPP_SIZE_ERROR, 44
- NPP_STEP_ERROR, 44
- NPP_STRIDE_ERROR, 43
- NPP_SUCCESS, 44
- NPP_TEXTURE_BIND_ERROR, 43
- NPP_THRESHOLD_ERROR, 44
- NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR, 44
- NPP_VERTICAL_AXIS, 40
- NPP_WRONG_INTERSECTION_QUAD_WARNING, 44
- NPP_WRONG_INTERSECTION_ROI_ERROR, 43
- NPP_WRONG_INTERSECTION_ROI_WARNING, 44
- NPP_ZC_MODE_NOT_SUPPORTED_ERROR, 43
- NPP_ZERO_MASK_VALUE_ERROR, 43
- NPPI_BAYER_BGGR, 40
- NPPI_BAYER_GBRG, 40
- NPPI_BAYER_GRBG, 40
- NPPI_BAYER_RRGB, 40
- NPPI_INTER_CUBIC, 41
- NPPI_INTER_CUBIC2P_B05C03, 41
- NPPI_INTER_CUBIC2P_BSPLINE, 41
- NPPI_INTER_CUBIC2P_CATMULLROM, 41
- NPPI_INTER_LANCZOS, 41
- NPPI_INTER_LANCZOS3_ADVANCED, 41
- NPPI_INTER_LINEAR, 41
- NPPI_INTER_NN, 41
- NPPI_INTER_SUPER, 41
- NPPI_INTER_UNDEFINED, 41
- NPPI_OP_ALPHA_ATOP, 39
- NPPI_OP_ALPHA_ATOP_PREMUL, 40
- NPPI_OP_ALPHA_IN, 39
- NPPI_OP_ALPHA_IN_PREMUL, 40
- NPPI_OP_ALPHA_OUT, 39
- NPPI_OP_ALPHA_OUT_PREMUL, 40
- NPPI_OP_ALPHA_OVER, 39
- NPPI_OP_ALPHA_OVER_PREMUL, 40
- NPPI_OP_ALPHA_PLUS, 39
- NPPI_OP_ALPHA_PLUS_PREMUL, 40
- NPPI_OP_ALPHA_PREMUL, 40
- NPPI_OP_ALPHA_XOR, 39
- NPPI_OP_ALPHA_XOR_PREMUL, 40
- NPPI_SMOOTH_EDGE, 41
- nppiACTable, 41
- nppiDCTable, 41
- nppiNormInf, 42
- nppiNormL1, 42
- nppiNormL2, 42
- nppZCC, 45
- nppZCR, 45
- nppZCXor, 45
- typedefs_npp
 - NPP_MAX_16S, 37
 - NPP_MAX_16U, 37
 - NPP_MAX_32S, 37
 - NPP_MAX_32U, 37
 - NPP_MAX_64S, 37
 - NPP_MAX_64U, 37
 - NPP_MAX_8S, 37
 - NPP_MAX_8U, 37
 - NPP_MAXABS_32F, 37
 - NPP_MAXABS_64F, 37
 - NPP_MIN_16S, 37
 - NPP_MIN_16U, 38
 - NPP_MIN_32S, 38
 - NPP_MIN_32U, 38
 - NPP_MIN_64S, 38
 - NPP_MIN_64U, 38
 - NPP_MIN_8S, 38
 - NPP_MIN_8U, 38
 - NPP_MINABS_32F, 38
 - NPP_MINABS_64F, 38
 - NppCmpOp, 38
 - NppGpuComputeCapability, 39
 - NppHintAlgorithm, 39
 - NppiAlphaOp, 39
 - NppiAxis, 40
 - NppiBayerGridPosition, 40
 - NppiBorderType, 40
 - NppiDifferentialKernel, 40
 - NppiHuffmanTableType, 40
 - NppiInterpolationMode, 41

NppiMaskSize, [41](#)
NppiNorm, [41](#)
NppRoundMode, [42](#)
NppStatus, [42](#)
NppsZCType, [44](#)

width

NppiRect, [786](#)
NppiSize, [787](#)

x

NppiPoint, [785](#)
NppiRect, [786](#)

y

NppiPoint, [785](#)
NppiRect, [786](#)