

CDVS Test Model

14.0

Generated by Doxygen 1.8.6

Thu Jun 11 2015 14:54:16



# Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Requirements . . . . .	1
1.2.1	Requirements for building . . . . .	2
1.2.2	Requirements to run the code . . . . .	2
1.3	Building the code . . . . .	2
1.4	Running the CDVS experiments . . . . .	2
1.5	How to rebuild this documentation from source code . . . . .	3
1.6	Acknowledgements . . . . .	3
<b>2</b>	<b>Namespace Index</b>	<b>5</b>
2.1	Namespace List . . . . .	5
<b>3</b>	<b>Hierarchical Index</b>	<b>7</b>
3.1	Class Hierarchy . . . . .	7
<b>4</b>	<b>Data Structure Index</b>	<b>9</b>
4.1	Data Structures . . . . .	9
<b>5</b>	<b>File Index</b>	<b>13</b>
5.1	File List . . . . .	13
<b>6</b>	<b>Namespace Documentation</b>	<b>15</b>
6.1	mpeg7cdvs Namespace Reference . . . . .	15
6.1.1	Detailed Description . . . . .	17
6.1.2	Typedef Documentation . . . . .	17
6.1.2.1	CODE_VALUE . . . . .	17
6.1.2.2	element_type . . . . .	17
6.1.2.3	filter_type . . . . .	17
6.1.2.4	gs_type . . . . .	17
6.1.2.5	input_type . . . . .	18
6.1.2.6	ParameterSet . . . . .	18
6.1.2.7	recallGraph_t . . . . .	18

6.1.2.8	<a href="#">recallGraphNode_t</a>	18
6.1.3	<a href="#">Enumeration Type Documentation</a>	18
6.1.3.1	<a href="#">anonymous enum</a>	18
6.1.3.2	<a href="#">anonymous enum</a>	18
6.1.4	<a href="#">Variable Documentation</a>	18
6.1.4.1	<a href="#">CODE_VALUE_BITS</a>	18
6.1.4.2	<a href="#">FREQUENCY_BITS</a>	18
6.1.4.3	<a href="#">gama</a>	18
6.1.4.4	<a href="#">MAX_CODE</a>	18
6.1.4.5	<a href="#">MAX_FREQ</a>	18
6.1.4.6	<a href="#">NOT_FOUND</a>	18
6.1.4.7	<a href="#">num_bit_selection</a>	18
6.1.4.8	<a href="#">numberCentroids</a>	18
6.1.4.9	<a href="#">ONE_FOURTH</a>	18
6.1.4.10	<a href="#">ONE_HALF</a>	18
6.1.4.11	<a href="#">PCASiftLength</a>	19
6.1.4.12	<a href="#">PRECISION</a>	19
6.1.4.13	<a href="#">THREE_FOURTHS</a>	19
6.1.4.14	<a href="#">TOP_VALUE</a>	19
<b>7</b>	<b><a href="#">Data Structure Documentation</a></b>	<b>21</b>
7.1	<a href="#">mpeg7cdvs::AbstractDetector Class Reference</a>	21
7.1.1	<a href="#">Detailed Description</a>	22
7.1.2	<a href="#">Constructor &amp; Destructor Documentation</a>	22
7.1.2.1	<a href="#">AbstractDetector</a>	22
7.1.2.2	<a href="#">~AbstractDetector</a>	22
7.1.3	<a href="#">Member Function Documentation</a>	22
7.1.3.1	<a href="#">detect</a>	22
7.1.3.2	<a href="#">extract</a>	22
7.1.4	<a href="#">Field Documentation</a>	22
7.1.4.1	<a href="#">height</a>	22
7.1.4.2	<a href="#">originalHeight</a>	23
7.1.4.3	<a href="#">originalWidth</a>	23
7.1.4.4	<a href="#">width</a>	23
7.2	<a href="#">mpeg7cdvs::AC_decoder Class Reference</a>	23
7.2.1	<a href="#">Detailed Description</a>	23
7.2.2	<a href="#">Member Function Documentation</a>	23
7.2.2.1	<a href="#">bits</a>	23
7.2.2.2	<a href="#">decode_symbol</a>	24
7.2.2.3	<a href="#">done</a>	25

7.2.2.4	init	25
7.3	mpeg7cdvs::AC_encoder Class Reference	25
7.3.1	Detailed Description	25
7.3.2	Member Function Documentation	26
7.3.2.1	bits	26
7.3.2.2	done	26
7.3.2.3	encode_symbol	26
7.3.2.4	init	26
7.4	mpeg7cdvs::AC_model Class Reference	26
7.4.1	Detailed Description	27
7.4.2	Constructor & Destructor Documentation	27
7.4.2.1	AC_model	27
7.4.2.2	~AC_model	27
7.4.3	Member Function Documentation	27
7.4.3.1	adaptative	27
7.4.3.2	check	27
7.4.3.3	count	28
7.4.3.4	done	28
7.4.3.5	getSymbol	28
7.4.3.6	high	28
7.4.3.7	init	29
7.4.3.8	init	29
7.4.3.9	low	29
7.4.3.10	print	29
7.4.3.11	update	29
7.5	mpeg7cdvs::AlpDetector Class Reference	30
7.5.1	Detailed Description	31
7.5.2	Constructor & Destructor Documentation	31
7.5.2.1	AlpDetector	31
7.5.2.2	~AlpDetector	31
7.5.3	Member Function Documentation	31
7.5.3.1	detect	31
7.5.3.2	extract	31
7.5.3.3	sortPdf	32
7.5.3.4	sortPdf	32
7.6	mpeg7cdvs::AlpDetectorBF Class Reference	32
7.6.1	Detailed Description	33
7.6.2	Constructor & Destructor Documentation	33
7.6.2.1	AlpDetectorBF	33
7.6.3	Member Function Documentation	33

7.6.3.1	detect	33
7.6.3.2	extract	34
7.7	mpeg7cdvs::AlpDetectorLowMem Class Reference	34
7.7.1	Detailed Description	35
7.7.2	Constructor & Destructor Documentation	35
7.7.2.1	AlpDetectorLowMem	35
7.7.3	Member Function Documentation	35
7.7.3.1	detect	35
7.7.3.2	extract	36
7.8	mpeg7cdvs::AlpOctave Class Reference	36
7.8.1	Detailed Description	37
7.8.2	Constructor & Destructor Documentation	37
7.8.2.1	AlpOctave	37
7.8.2.2	~AlpOctave	37
7.8.3	Member Function Documentation	38
7.8.3.1	clear	38
7.8.3.2	computeDescriptor	38
7.8.3.3	computeDescriptor	38
7.8.3.4	computeDescriptor_fast_mode	38
7.8.3.5	detect	38
7.8.3.6	detectDuplicates	38
7.8.3.7	detectDuplicates	38
7.8.3.8	empty	39
7.8.3.9	getAlpKeypoints	39
7.8.3.10	getHeight	39
7.8.3.11	getKeypoints	39
7.8.3.12	getKeypoints_fast_mode	39
7.8.3.13	getOctave	40
7.8.3.14	getWidth	40
7.8.3.15	init	40
7.8.3.16	init	40
7.8.3.17	processFirst	41
7.8.3.18	processNext	41
7.8.4	Field Documentation	41
7.8.4.1	keypoints	41
7.9	mpeg7cdvs::AlpOctaveBF Class Reference	41
7.9.1	Detailed Description	43
7.9.2	Constructor & Destructor Documentation	43
7.9.2.1	AlpOctaveBF	43
7.9.2.2	~AlpOctaveBF	43

7.9.3	Member Function Documentation	43
7.9.3.1	Allocate	43
7.9.3.2	Clear	43
7.9.3.3	computeDescriptor	43
7.9.3.4	detect	43
7.9.3.5	detectDuplicates	44
7.9.3.6	empty	44
7.9.3.7	FrequencyFiltering	44
7.9.3.8	getBlockInput	44
7.9.3.9	getHeight	44
7.9.3.10	getOctave	44
7.9.3.11	getWidth	44
7.9.3.12	processOctave	44
7.9.3.13	SpatialFiltering	44
7.9.4	Field Documentation	45
7.9.4.1	block_buffer_len	45
7.9.4.2	block_col_n	45
7.9.4.3	block_col_remain_height	45
7.9.4.4	block_height	45
7.9.4.5	block_map	45
7.9.4.6	block_row_n	45
7.9.4.7	block_row_remain_width	45
7.9.4.8	block_width	45
7.9.4.9	cur_block_id	45
7.9.4.10	extrema_h	45
7.9.4.11	extrema_sx	45
7.9.4.12	extrema_sy	45
7.9.4.13	extrema_w	45
7.9.4.14	filters	45
7.9.4.15	h2	45
7.9.4.16	h_mode	45
7.9.4.17	keypoints	45
7.9.4.18	padWidth	45
7.9.4.19	sx	45
7.9.4.20	sy	45
7.9.4.21	v_mode	45
7.9.4.22	w2	45
7.9.4.23	windowSize	45
7.10	Ball2D Struct Reference	46
7.10.1	Constructor & Destructor Documentation	46

7.10.1.1	Ball2D	46
7.10.2	Field Documentation	46
7.10.2.1	center	46
7.10.2.2	radius	46
7.11	mpeg7cdvs::BitInputStream Class Reference	46
7.11.1	Detailed Description	47
7.11.2	Constructor & Destructor Documentation	47
7.11.2.1	BitInputStream	47
7.11.2.2	BitInputStream	47
7.11.2.3	~BitInputStream	48
7.11.3	Member Function Documentation	48
7.11.3.1	align	48
7.11.3.2	available	48
7.11.3.3	close	48
7.11.3.4	consumed	48
7.11.3.5	eof	48
7.11.3.6	getPointer	48
7.11.3.7	getSize	49
7.11.3.8	jumpTo	49
7.11.3.9	open	49
7.11.3.10	read	49
7.11.3.11	read	49
7.11.3.12	read	49
7.11.3.13	reset	50
7.11.3.14	skip	50
7.12	mpeg7cdvs::BitOutputStream Class Reference	50
7.12.1	Detailed Description	51
7.12.2	Constructor & Destructor Documentation	51
7.12.2.1	BitOutputStream	51
7.12.2.2	BitOutputStream	51
7.12.2.3	~BitOutputStream	51
7.12.3	Member Function Documentation	51
7.12.3.1	align	51
7.12.3.2	available	52
7.12.3.3	close	52
7.12.3.4	eof	52
7.12.3.5	flush	52
7.12.3.6	getPointer	52
7.12.3.7	getSize	52
7.12.3.8	jumpTo	52



7.12.3.9	open	53
7.12.3.10	produced	53
7.12.3.11	reset	53
7.12.3.12	skip	53
7.12.3.13	write	53
7.12.3.14	write	53
7.12.3.15	write	54
7.13	BoundingBox Class Reference	54
7.13.1	Detailed Description	54
7.13.2	Constructor & Destructor Documentation	54
7.13.2.1	BoundingBox	54
7.13.3	Member Function Documentation	54
7.13.3.1	cfind_overlap	55
7.13.3.2	find_overlap	55
7.13.3.3	getAddress	55
7.13.3.4	read	55
7.14	mpeg7cdvs::Buffer Class Reference	55
7.14.1	Detailed Description	56
7.14.2	Constructor & Destructor Documentation	57
7.14.2.1	Buffer	57
7.14.2.2	~Buffer	57
7.14.2.3	Buffer	57
7.14.2.4	Buffer	57
7.14.2.5	Buffer	57
7.14.3	Member Function Documentation	57
7.14.3.1	assign	57
7.14.3.2	clear	57
7.14.3.3	compare	57
7.14.3.4	data	57
7.14.3.5	data	57
7.14.3.6	empty	57
7.14.3.7	equals	58
7.14.3.8	fill	58
7.14.3.9	operator=	58
7.14.3.10	operator==	58
7.14.3.11	read	58
7.14.3.12	resize	58
7.14.3.13	size	58
7.14.3.14	swap	58
7.14.3.15	write	58

7.15	<a href="#">mpeg7cdvs::CdvsClient Class Reference</a>	58
7.15.1	<a href="#">Detailed Description</a>	59
7.15.2	<a href="#">Constructor &amp; Destructor Documentation</a>	59
7.15.2.1	<a href="#">~CdvsClient</a>	59
7.15.3	<a href="#">Member Function Documentation</a>	59
7.15.3.1	<a href="#">cdvsClientFactory</a>	59
7.15.3.2	<a href="#">encode</a>	60
7.16	<a href="#">mpeg7cdvs::CdvsClientBflog Class Reference</a>	60
7.16.1	<a href="#">Detailed Description</a>	61
7.16.2	<a href="#">Constructor &amp; Destructor Documentation</a>	61
7.16.2.1	<a href="#">CdvsClientBflog</a>	61
7.16.2.2	<a href="#">~CdvsClientBflog</a>	61
7.16.3	<a href="#">Member Function Documentation</a>	61
7.16.3.1	<a href="#">encode</a>	61
7.17	<a href="#">mpeg7cdvs::CdvsClientImpl Class Reference</a>	62
7.17.1	<a href="#">Detailed Description</a>	63
7.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	63
7.17.2.1	<a href="#">CdvsClientImpl</a>	63
7.17.2.2	<a href="#">~CdvsClientImpl</a>	63
7.17.3	<a href="#">Member Function Documentation</a>	63
7.17.3.1	<a href="#">encode</a>	63
7.17.4	<a href="#">Field Documentation</a>	64
7.17.4.1	<a href="#">g_factory</a>	64
7.17.4.2	<a href="#">modeld</a>	64
7.17.4.3	<a href="#">params</a>	64
7.18	<a href="#">mpeg7cdvs::CdvsClientLowMem Class Reference</a>	64
7.18.1	<a href="#">Detailed Description</a>	65
7.18.2	<a href="#">Constructor &amp; Destructor Documentation</a>	65
7.18.2.1	<a href="#">CdvsClientLowMem</a>	65
7.18.2.2	<a href="#">~CdvsClientLowMem</a>	65
7.18.3	<a href="#">Member Function Documentation</a>	65
7.18.3.1	<a href="#">encode</a>	65
7.19	<a href="#">mpeg7cdvs::CdvsConfiguration Class Reference</a>	66
7.19.1	<a href="#">Detailed Description</a>	67
7.19.2	<a href="#">Constructor &amp; Destructor Documentation</a>	67
7.19.2.1	<a href="#">~CdvsConfiguration</a>	67
7.19.3	<a href="#">Member Function Documentation</a>	67
7.19.3.1	<a href="#">cdvsConfigurationFactory</a>	67
7.19.3.2	<a href="#">getMode</a>	67
7.19.3.3	<a href="#">getParameters</a>	68

7.19.3.4	setParameters	69
7.20	mpeg7cdvs::CdvsConfigurationImpl Class Reference	69
7.20.1	Detailed Description	70
7.20.2	Constructor & Destructor Documentation	70
7.20.2.1	CdvsConfigurationImpl	70
7.20.2.2	~CdvsConfigurationImpl	70
7.20.3	Member Function Documentation	70
7.20.3.1	getParameters	70
7.20.3.2	setParameters	71
7.21	mpeg7cdvs::CdvsDescriptor Class Reference	71
7.21.1	Detailed Description	73
7.21.2	Constructor & Destructor Documentation	73
7.21.2.1	CdvsDescriptor	73
7.21.2.2	~CdvsDescriptor	73
7.21.3	Member Function Documentation	73
7.21.3.1	check	73
7.21.3.2	clear	73
7.21.3.3	decode	73
7.21.3.4	encode	74
7.21.3.5	getGlobalHasBitSelection	74
7.21.3.6	getGlobalHasVariance	74
7.21.3.7	getHistogramCountSize	74
7.21.3.8	getHistogramMapSizeX	74
7.21.3.9	getHistogramMapSizeY	74
7.21.3.10	getModelID	74
7.21.3.11	getNumberOfLocalDescriptors	74
7.21.3.12	getOriginalImageXResolution	74
7.21.3.13	getOriginalImageYResolution	75
7.21.3.14	getRelevanceBitsPresent	75
7.21.3.15	getVersionID	75
7.21.3.16	print	75
7.21.3.17	setGlobalHasBitSelection	75
7.21.3.18	setGlobalHasVariance	75
7.21.3.19	setHistogramCountSize	75
7.21.3.20	setHistogramMapSizeX	75
7.21.3.21	setHistogramMapSizeY	75
7.21.3.22	setModelID	75
7.21.3.23	setNumberOfLocalDescriptors	75
7.21.3.24	setOriginalImageXResolution	76
7.21.3.25	setOriginalImageYResolution	76

7.21.3.26 setRelevanceBitsPresent . . . . .	76
7.21.3.27 setVersionID . . . . .	76
7.21.4 Field Documentation . . . . .	76
7.21.4.1 buffer . . . . .	76
7.21.4.2 featurelist . . . . .	76
7.21.4.3 scfvSignature . . . . .	76
7.22 mpeg7cdvs::CdvsException Class Reference . . . . .	76
7.22.1 Detailed Description . . . . .	77
7.22.2 Constructor & Destructor Documentation . . . . .	77
7.22.2.1 CdvsException . . . . .	77
7.22.2.2 ~CdvsException . . . . .	78
7.22.3 Member Function Documentation . . . . .	78
7.22.3.1 what . . . . .	78
7.23 mpeg7cdvs::CDVSPPOINT Struct Reference . . . . .	78
7.23.1 Detailed Description . . . . .	78
7.23.2 Field Documentation . . . . .	78
7.23.2.1 x . . . . .	78
7.23.2.2 y . . . . .	78
7.24 mpeg7cdvs::CdvsServer Class Reference . . . . .	78
7.24.1 Detailed Description . . . . .	80
7.24.2 Constructor & Destructor Documentation . . . . .	80
7.24.2.1 ~CdvsServer . . . . .	80
7.24.3 Member Function Documentation . . . . .	80
7.24.3.1 addDescriptorToDB . . . . .	80
7.24.3.2 cdvsServerFactory . . . . .	80
7.24.3.3 clearDB . . . . .	81
7.24.3.4 commitDB . . . . .	81
7.24.3.5 createDB . . . . .	81
7.24.3.6 decode . . . . .	81
7.24.3.7 decode . . . . .	81
7.24.3.8 getImageld . . . . .	82
7.24.3.9 isDescriptorInDB . . . . .	82
7.24.3.10 loadDB . . . . .	82
7.24.3.11 match . . . . .	82
7.24.3.12 match . . . . .	83
7.24.3.13 replaceDescriptorInDB . . . . .	83
7.24.3.14 retrieve . . . . .	84
7.24.3.15 sizeofDB . . . . .	84
7.24.3.16 storeDB . . . . .	84
7.25 mpeg7cdvs::CdvsServerImpl Class Reference . . . . .	85

7.25.1	Detailed Description	86
7.25.2	Constructor & Destructor Documentation	86
7.25.2.1	CdvsServerImpl	86
7.25.2.2	~CdvsServerImpl	86
7.25.3	Member Function Documentation	86
7.25.3.1	addDescriptorToDB	86
7.25.3.2	clearDB	87
7.25.3.3	commitDB	87
7.25.3.4	createDB	87
7.25.3.5	decode	87
7.25.3.6	decode	87
7.25.3.7	getImageld	88
7.25.3.8	isDescriptorInDB	88
7.25.3.9	loadDB	88
7.25.3.10	match	88
7.25.3.11	match	89
7.25.3.12	replaceDescriptorInDB	89
7.25.3.13	retrieve	90
7.25.3.14	sizeofDB	90
7.25.3.15	storeDB	90
7.26	mpeg7cdvs::CsscCoordinateCoding::CircularSumContext Struct Reference	91
7.26.1	Detailed Description	91
7.26.2	Field Documentation	91
7.26.2.1	vCount	91
7.26.2.2	vInitialMap	91
7.26.2.3	vMap	91
7.27	mpeg7cdvs::CompressedFeatureList Class Reference	91
7.27.1	Detailed Description	93
7.27.2	Constructor & Destructor Documentation	93
7.27.2.1	CompressedFeatureList	93
7.27.2.2	CompressedFeatureList	93
7.27.2.3	CompressedFeatureList	93
7.27.2.4	~CompressedFeatureList	93
7.27.2.5	CompressedFeatureList	93
7.27.3	Member Function Documentation	93
7.27.3.1	descrBytes	93
7.27.3.2	getDistance	94
7.27.3.3	matchDescriptors_oneWay	95
7.27.3.4	matchDescriptors_twoWay	95
7.27.3.5	nFeatures	95

7.27.3.6	<a href="#">operator=</a>	95
7.27.3.7	<a href="#">print</a>	96
7.27.3.8	<a href="#">read</a>	96
7.27.3.9	<a href="#">readFromFile</a>	96
7.27.3.10	<a href="#">setFilename</a>	96
7.27.3.11	<a href="#">swap</a>	96
7.27.3.12	<a href="#">write</a>	96
7.27.3.13	<a href="#">writeToFile</a>	97
7.27.4	<a href="#">Field Documentation</a>	97
7.27.4.1	<a href="#">features</a>	97
7.27.4.2	<a href="#">imagefile</a>	97
7.27.4.3	<a href="#">imageHeight</a>	97
7.27.4.4	<a href="#">imageWidth</a>	97
7.27.4.5	<a href="#">nDescLength</a>	97
7.27.4.6	<a href="#">numFeatures</a>	97
7.27.4.7	<a href="#">originalHeight</a>	97
7.27.4.8	<a href="#">originalWidth</a>	98
7.27.4.9	<a href="#">Xcoord</a>	98
7.27.4.10	<a href="#">Ycoord</a>	98
7.28	<a href="#">mpeg7cdvs::CsscCoordinateCoding Class Reference</a>	98
7.28.1	<a href="#">Detailed Description</a>	99
7.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	99
7.28.2.1	<a href="#">CsscCoordinateCoding</a>	99
7.28.2.2	<a href="#">~CsscCoordinateCoding</a>	99
7.28.3	<a href="#">Member Function Documentation</a>	99
7.28.3.1	<a href="#">AddImageSample</a>	99
7.28.3.2	<a href="#">compare</a>	99
7.28.3.3	<a href="#">EndTrainingMode</a>	100
7.28.3.4	<a href="#">exportVars</a>	100
7.28.3.5	<a href="#">fromBinary</a>	101
7.28.3.6	<a href="#">generateFeatureList</a>	101
7.28.3.7	<a href="#">generateHistogramMap</a>	101
7.28.3.8	<a href="#">readSeparateContext</a>	101
7.28.3.9	<a href="#">readSeparateContext</a>	101
7.28.3.10	<a href="#">StartTrainingMode</a>	101
7.28.3.11	<a href="#">toBinary</a>	101
7.28.3.12	<a href="#">writeSeparateContext</a>	101
7.28.4	<a href="#">Field Documentation</a>	101
7.28.4.1	<a href="#">CONTEXT_RANGE</a>	102
7.28.4.2	<a href="#">MAXIMUM_SUM_CONTEXT</a>	102

7.28.4.3	SUM_HIST_COUNT_SIZE	102
7.29	mpeg7cdvs::Database Class Reference	102
7.29.1	Detailed Description	103
7.29.2	Constructor & Destructor Documentation	103
7.29.2.1	Database	103
7.29.2.2	~Database	103
7.29.3	Member Function Documentation	103
7.29.3.1	addImage	103
7.29.3.2	clear	104
7.29.3.3	copyImageName	104
7.29.3.4	find	104
7.29.3.5	getImageName	104
7.29.3.6	getMode	104
7.29.3.7	getRecallGraph	105
7.29.3.8	hasRecallGraph	105
7.29.3.9	matchCompressedDescriptors_oneWay	105
7.29.3.10	matchCompressedDescriptors_twoWay	105
7.29.3.11	merge	106
7.29.3.12	read	106
7.29.3.13	readFromFile	106
7.29.3.14	readHeader	106
7.29.3.15	replaceImage	106
7.29.3.16	size	107
7.29.3.17	write	107
7.29.3.18	writeToFile	107
7.29.4	Field Documentation	107
7.29.4.1	images	107
7.29.4.2	modeld	107
7.29.4.3	recallGraph	108
7.30	DistratEigen Class Reference	108
7.30.1	Detailed Description	108
7.30.2	Constructor & Destructor Documentation	108
7.30.2.1	~DistratEigen	108
7.30.2.2	DistratEigen	108
7.30.3	Member Function Documentation	109
7.30.3.1	estimateInliers	109
7.30.4	Field Documentation	109
7.30.4.1	m_bFitsGood	109
7.30.4.2	m_c	109
7.30.4.3	m_GoFThreshold	109

7.31	<a href="#">mpeg7cdvs::Feature Class Reference</a>	109
7.31.1	<a href="#">Detailed Description</a>	110
7.31.2	<a href="#">Constructor &amp; Destructor Documentation</a>	111
7.31.2.1	<a href="#">Feature</a>	111
7.31.3	<a href="#">Member Function Documentation</a>	111
7.31.3.1	<a href="#">fromFile</a>	111
7.31.3.2	<a href="#">toFile</a>	111
7.31.4	<a href="#">Field Documentation</a>	111
7.31.4.1	<a href="#">curvRatio</a>	111
7.31.4.2	<a href="#">curvSigma</a>	111
7.31.4.3	<a href="#">descr</a>	111
7.31.4.4	<a href="#">descrLength</a>	111
7.31.4.5	<a href="#">iscale</a>	111
7.31.4.6	<a href="#">octave</a>	111
7.31.4.7	<a href="#">orientation</a>	111
7.31.4.8	<a href="#">pdf</a>	112
7.31.4.9	<a href="#">peak</a>	112
7.31.4.10	<a href="#">qdescr</a>	112
7.31.4.11	<a href="#">relevance</a>	112
7.31.4.12	<a href="#">scale</a>	112
7.31.4.13	<a href="#">spatialIndex</a>	112
7.31.4.14	<a href="#">x</a>	112
7.31.4.15	<a href="#">y</a>	112
7.32	<a href="#">mpeg7cdvs::FeatureAlp Class Reference</a>	112
7.32.1	<a href="#">Detailed Description</a>	114
7.32.2	<a href="#">Member Function Documentation</a>	114
7.32.2.1	<a href="#">distL1</a>	114
7.32.3	<a href="#">Friends And Related Function Documentation</a>	114
7.32.3.1	<a href="#">operator&lt;&lt;</a>	114
7.32.4	<a href="#">Field Documentation</a>	114
7.32.4.1	<a href="#">curvRatio</a>	114
7.32.4.2	<a href="#">curvSigma</a>	114
7.32.4.3	<a href="#">iscale</a>	114
7.32.4.4	<a href="#">ix</a>	114
7.32.4.5	<a href="#">iy</a>	114
7.32.4.6	<a href="#">octave</a>	115
7.32.4.7	<a href="#">orientation</a>	115
7.32.4.8	<a href="#">pdf</a>	115
7.32.4.9	<a href="#">peak</a>	115
7.32.4.10	<a href="#">sigma</a>	115



7.32.4.11 spatialIndex . . . . .	115
7.32.4.12 x . . . . .	115
7.32.4.13 y . . . . .	115
7.33 mpeg7cdvs::FeatureAlpBF Class Reference . . . . .	115
7.33.1 Detailed Description . . . . .	116
7.33.2 Field Documentation . . . . .	116
7.33.2.1 block_id . . . . .	116
7.33.2.2 mode . . . . .	117
7.34 mpeg7cdvs::FeatureList Class Reference . . . . .	117
7.34.1 Detailed Description . . . . .	118
7.34.2 Constructor & Destructor Documentation . . . . .	119
7.34.2.1 FeatureList . . . . .	119
7.34.3 Member Function Documentation . . . . .	119
7.34.3.1 addFeature . . . . .	119
7.34.3.2 clear . . . . .	119
7.34.3.3 compareCoordinates . . . . .	119
7.34.3.4 compareDescriptors . . . . .	119
7.34.3.5 compareKeypoints . . . . .	119
7.34.3.6 compress . . . . .	121
7.34.3.7 computeMaxPoints . . . . .	121
7.34.3.8 fromBinary . . . . .	121
7.34.3.9 fromFile . . . . .	121
7.34.3.10 fromFile . . . . .	121
7.34.3.11 getRelevantPoints . . . . .	122
7.34.3.12 nFeatures . . . . .	122
7.34.3.13 print . . . . .	122
7.34.3.14 select . . . . .	122
7.34.3.15 selectFirst . . . . .	122
7.34.3.16 selectFromTo . . . . .	122
7.34.3.17 setRelevantPoints . . . . .	122
7.34.3.18 setResolution . . . . .	123
7.34.3.19 sortRelevance . . . . .	123
7.34.3.20 sortSpatialIndex . . . . .	123
7.34.3.21 toBinary . . . . .	123
7.34.3.22 toFile . . . . .	123
7.34.3.23 toFile . . . . .	123
7.34.4 Friends And Related Function Documentation . . . . .	123
7.34.4.1 CompressedFeatureList . . . . .	123
7.34.5 Field Documentation . . . . .	124
7.34.5.1 features . . . . .	124

7.34.5.2	imageHeight	124
7.34.5.3	imageWidth	124
7.34.5.4	MAX_NUM_FEATURES	124
7.34.5.5	originalHeight	124
7.34.5.6	originalWidth	124
7.34.5.7	qdescr_size	124
7.35	FileManager Class Reference	124
7.35.1	Detailed Description	125
7.35.2	Constructor & Destructor Documentation	125
7.35.2.1	FileManager	125
7.35.2.2	~FileManager	125
7.35.3	Member Function Documentation	125
7.35.3.1	getAbsolutePathname	125
7.35.3.2	getQueryName	125
7.35.3.3	getReferenceName	126
7.35.3.4	getRelativePathname	126
7.35.3.5	readAnnotation	126
7.35.3.6	replaceExt	126
7.35.3.7	setAnnotationPath	127
7.35.3.8	setDatasetPath	127
7.36	mpeg7cdvs::Filter Class Reference	127
7.36.1	Detailed Description	128
7.36.2	Constructor & Destructor Documentation	128
7.36.2.1	Filter	128
7.36.3	Member Function Documentation	128
7.36.3.1	print	128
7.36.4	Field Documentation	128
7.36.4.1	kernel	128
7.36.4.2	maxsize	128
7.36.4.3	ntaps	128
7.36.4.4	sigma	128
7.37	mpeg7cdvs::FrequencyFilter Class Reference	128
7.37.1	Constructor & Destructor Documentation	129
7.37.1.1	FrequencyFilter	129
7.37.1.2	~FrequencyFilter	129
7.37.2	Member Function Documentation	129
7.37.2.1	close_lib	129
7.37.2.2	Convolution	129
7.37.2.3	open_lib	129
7.37.3	Field Documentation	129

7.37.3.1	<a href="#">fblock</a>	129
7.37.3.2	<a href="#">fblock_flog</a>	129
7.37.3.3	<a href="#">fftplan</a>	130
7.37.3.4	<a href="#">ifftplan</a>	130
7.37.3.5	<a href="#">inmat</a>	130
7.38	<a href="#">HiResTimer Class Reference</a>	130
7.38.1	<a href="#">Detailed Description</a>	130
7.38.2	<a href="#">Member Function Documentation</a>	130
7.38.2.1	<a href="#">elapsed</a>	130
7.38.2.2	<a href="#">start</a>	131
7.38.2.3	<a href="#">stop</a>	131
7.39	<a href="#">mpeg7cdvs::ImageBuffer Class Reference</a>	132
7.39.1	<a href="#">Detailed Description</a>	133
7.39.2	<a href="#">Constructor &amp; Destructor Documentation</a>	134
7.39.2.1	<a href="#">ImageBuffer</a>	134
7.39.2.2	<a href="#">~ImageBuffer</a>	134
7.39.3	<a href="#">Member Function Documentation</a>	134
7.39.3.1	<a href="#">fastInterpolate</a>	134
7.39.3.2	<a href="#">fastScalarQuantize</a>	134
7.39.3.3	<a href="#">print</a>	134
7.39.3.4	<a href="#">printDescr</a>	134
7.39.3.5	<a href="#">printHeader</a>	135
7.39.3.6	<a href="#">read</a>	135
7.39.3.7	<a href="#">resample</a>	135
7.39.3.8	<a href="#">resample</a>	135
7.39.3.9	<a href="#">resampleIfGreater</a>	135
7.39.3.10	<a href="#">resize</a>	136
7.39.3.11	<a href="#">scalarQuantize</a>	136
7.39.3.12	<a href="#">sortPdfPredicate</a>	136
7.39.3.13	<a href="#">sortPredicate</a>	136
7.39.3.14	<a href="#">swap</a>	137
7.39.3.15	<a href="#">writeBMP</a>	137
7.39.3.16	<a href="#">writeRawData</a>	137
7.39.4	<a href="#">Field Documentation</a>	137
7.39.4.1	<a href="#">buffer</a>	137
7.40	<a href="#">mpeg7cdvs::LookUpTable Class Reference</a>	137
7.40.1	<a href="#">Detailed Description</a>	138
7.40.2	<a href="#">Constructor &amp; Destructor Documentation</a>	138
7.40.2.1	<a href="#">LookUpTable</a>	138
7.40.3	<a href="#">Field Documentation</a>	138

7.40.3.1	f	138
7.41	Match Class Reference	138
7.41.1	Detailed Description	138
7.41.2	Member Function Documentation	139
7.41.2.1	sortMatchByWeight	139
7.41.2.2	sortMatchPredicate	139
7.41.3	Field Documentation	139
7.41.3.1	featureInd	139
7.41.3.2	otherFeatureInd	139
7.41.3.3	weight	139
7.42	mpeg7cdvs::Parameters Class Reference	140
7.42.1	Detailed Description	141
7.42.2	Constructor & Destructor Documentation	142
7.42.2.1	Parameters	142
7.42.2.2	~Parameters	142
7.42.3	Member Function Documentation	142
7.42.3.1	getModelID	142
7.42.3.2	readAll	142
7.42.3.3	readAll	142
7.42.3.4	readParameters	142
7.42.3.5	readParameters	143
7.42.4	Field Documentation	143
7.42.4.1	blockWidth	143
7.42.4.2	chiSquarePercentile	143
7.42.4.3	ctxTableIdx	143
7.42.4.4	debugLevel	143
7.42.4.5	descLength	143
7.42.4.6	gdThreshold	143
7.42.4.7	gdThresholdMixed	143
7.42.4.8	hasBitSelection	143
7.42.4.9	hasVar	143
7.42.4.10	locationBits	144
7.42.4.11	minNumInliers	144
7.42.4.12	modeExt	144
7.42.4.13	nBits	144
7.42.4.14	nModes	144
7.42.4.15	numberOfElementGroups	144
7.42.4.16	numRelevantPoints	144
7.42.4.17	queryExpansionLoops	144
7.42.4.18	ransacNumTests	144

7.42.4.19	<a href="#">ransacThreshold</a>	144
7.42.4.20	<a href="#">ratioThreshold</a>	144
7.42.4.21	<a href="#">resizeMaxSize</a>	144
7.42.4.22	<a href="#">retrievalLoops</a>	145
7.42.4.23	<a href="#">retrievalMaxPoints</a>	145
7.42.4.24	<a href="#">scfvThreshold</a>	145
7.42.4.25	<a href="#">selectMaxPoints</a>	145
7.42.4.26	<a href="#">wmMixed</a>	145
7.42.4.27	<a href="#">wmMixed2Way</a>	145
7.42.4.28	<a href="#">wmRetrieval</a>	145
7.42.4.29	<a href="#">wmRetrieval2Way</a>	145
7.42.4.30	<a href="#">wmThreshold</a>	145
7.42.4.31	<a href="#">wmThreshold2Way</a>	145
7.43	<a href="#">mpeg7cdvs::PointPairs Class Reference</a>	145
7.43.1	<a href="#">Detailed Description</a>	147
7.43.2	<a href="#">Constructor &amp; Destructor Documentation</a>	147
7.43.2.1	<a href="#">PointPairs</a>	147
7.43.2.2	<a href="#">PointPairs</a>	147
7.43.2.3	<a href="#">PointPairs</a>	147
7.43.2.4	<a href="#">~PointPairs</a>	147
7.43.3	<a href="#">Member Function Documentation</a>	147
7.43.3.1	<a href="#">addPair</a>	147
7.43.3.2	<a href="#">getInlierWeight</a>	148
7.43.3.3	<a href="#">getTotalWeight</a>	148
7.43.3.4	<a href="#">hasLocalizationInliers</a>	148
7.43.3.5	<a href="#">operator=</a>	148
7.43.3.6	<a href="#">toFullResolution</a>	148
7.43.4	<a href="#">Field Documentation</a>	148
7.43.4.1	<a href="#">global_score</a>	148
7.43.4.2	<a href="#">global_threshold</a>	148
7.43.4.3	<a href="#">inlierIndexes</a>	149
7.43.4.4	<a href="#">local_score</a>	149
7.43.4.5	<a href="#">local_threshold</a>	149
7.43.4.6	<a href="#">match_dirs</a>	149
7.43.4.7	<a href="#">nInliers</a>	149
7.43.4.8	<a href="#">nMatched</a>	149
7.43.4.9	<a href="#">score</a>	149
7.43.4.10	<a href="#">size</a>	149
7.43.4.11	<a href="#">weights</a>	149
7.43.4.12	<a href="#">x1</a>	149

7.43.4.13 x2	149
7.43.4.14 y1	149
7.43.4.15 y2	150
7.44 Polygon Class Reference	150
7.44.1 Detailed Description	150
7.44.2 Constructor & Destructor Documentation	150
7.44.2.1 Polygon	150
7.44.3 Member Function Documentation	150
7.44.3.1 inter	150
7.45 Projective2D Class Reference	150
7.45.1 Detailed Description	151
7.45.2 Constructor & Destructor Documentation	151
7.45.2.1 Projective2D	151
7.45.3 Member Function Documentation	151
7.45.3.1 isIdentity	151
7.45.3.2 makeHomography	152
7.45.3.3 makeHomography	152
7.45.3.4 modelFit	152
7.45.3.5 moveByHomography	152
7.45.3.6 moveByHomography	152
7.45.3.7 ransac	153
7.45.3.8 ransac	153
7.46 QRA Struct Reference	153
7.46.1 Detailed Description	153
7.46.2 Field Documentation	154
7.46.2.1 matches	154
7.46.2.2 n_matches	154
7.46.2.3 query	154
7.47 mpeg7cdvs::RetrievalData Struct Reference	154
7.47.1 Detailed Description	154
7.47.2 Field Documentation	154
7.47.2.1 fScore	154
7.47.2.2 gScore	154
7.47.2.3 index	155
7.47.2.4 nInliers	155
7.47.2.5 nMatched	155
7.48 mpeg7cdvs::SCFVFactory Class Reference	155
7.48.1 Detailed Description	155
7.48.2 Constructor & Destructor Documentation	155
7.48.2.1 SCFVFactory	155

7.48.3	Member Function Documentation	155
7.48.3.1	generateSCFV	155
7.48.3.2	hasBitSelection	156
7.48.3.3	hasVariance	156
7.48.3.4	init	156
7.49	mpeg7cdvs::SCFVIndex Class Reference	156
7.49.1	Detailed Description	157
7.49.2	Constructor & Destructor Documentation	157
7.49.2.1	SCFVIndex	157
7.49.3	Member Function Documentation	157
7.49.3.1	append	157
7.49.3.2	clear	157
7.49.3.3	generateWeight	157
7.49.3.4	getImage	158
7.49.3.5	loadHammingWeight	158
7.49.3.6	matchImages	158
7.49.3.7	matchImages_bitselection	158
7.49.3.8	numberImages	159
7.49.3.9	query	159
7.49.3.10	query_bitselection	159
7.49.3.11	read	159
7.49.3.12	replace	159
7.49.3.13	reserve	159
7.49.3.14	resize	160
7.49.3.15	write	160
7.50	mpeg7cdvs::SCFVSignature Class Reference	160
7.50.1	Detailed Description	161
7.50.2	Constructor & Destructor Documentation	161
7.50.2.1	SCFVSignature	161
7.50.3	Member Function Documentation	161
7.50.3.1	clear	161
7.50.3.2	compare	161
7.50.3.3	compressedNumBits	162
7.50.3.4	fromFile	162
7.50.3.5	getNorm	162
7.50.3.6	getVisited	162
7.50.3.7	hasBitSelection	162
7.50.3.8	hasBitSelection	162
7.50.3.9	hasVar	162
7.50.3.10	hasVar	162

7.50.3.11 print . . . . .	162
7.50.3.12 read . . . . .	162
7.50.3.13 setNorm . . . . .	162
7.50.3.14 setVisited . . . . .	162
7.50.3.15 size . . . . .	163
7.50.3.16 toFile . . . . .	163
7.50.3.17 write . . . . .	163
7.50.4 Field Documentation . . . . .	163
7.50.4.1 m_vWordBlock . . . . .	163
7.50.4.2 m_vWordVarBlock . . . . .	163
7.50.4.3 table_bit_selection . . . . .	163
7.51 TM_COUNTER Union Reference . . . . .	163
7.51.1 Detailed Description . . . . .	164
7.51.2 Field Documentation . . . . .	164
7.51.2.1 counter . . . . .	164
7.51.2.2 osxtmspec . . . . .	164
7.51.2.3 tmspec . . . . .	164
7.51.2.4 tv_nsec . . . . .	164
7.51.2.5 tv_sec . . . . .	164
7.51.2.6 tv_usec . . . . .	164
7.52 TM_STATE Struct Reference . . . . .	164
7.52.1 Detailed Description . . . . .	164
7.52.2 Field Documentation . . . . .	164
7.52.2.1 freq . . . . .	164
7.53 TraceManager Class Reference . . . . .	164
7.53.1 Detailed Description . . . . .	165
7.53.2 Constructor & Destructor Documentation . . . . .	165
7.53.2.1 TraceManager . . . . .	165
7.53.2.2 ~TraceManager . . . . .	165
7.53.3 Member Function Documentation . . . . .	165
7.53.3.1 isEnabled . . . . .	165
7.53.3.2 matchPair . . . . .	165
7.53.3.3 matchResults . . . . .	166
7.53.3.4 openTxt . . . . .	166
7.53.3.5 openXml . . . . .	166
7.53.3.6 start . . . . .	166
7.53.3.7 stop . . . . .	166
<b>8 File Documentation</b>	<b>167</b>
8.1 AbstractDetector.h File Reference . . . . .	167



8.2	AlpDetector.h File Reference	168
8.3	AlpDetectorBF.h File Reference	168
8.4	AlpDetectorLowMem.h File Reference	169
8.5	AlpOctave.h File Reference	170
8.6	AlpOctaveBF.h File Reference	171
8.6.1	Macro Definition Documentation	173
8.6.1.1	BLOCK_SIZE	173
8.6.1.2	BLOCK_WIDTH	173
8.6.1.3	BPO_P_PI2	173
8.6.1.4	EPSILON_D	173
8.6.1.5	EPSILON_F	173
8.6.1.6	FBLOCK_WIDTH	173
8.6.1.7	FHWIDTH	173
8.6.1.8	FILTER_GAUSSIAN	173
8.6.1.9	FILTER_LOG	173
8.6.1.10	HALF_PI	173
8.6.1.11	LOG_OF_2	173
8.6.1.12	MAX	173
8.6.1.13	MAX_FILTER_WIDTH	173
8.6.1.14	MIN	173
8.6.1.15	MODE_BOTTOM	173
8.6.1.16	MODE_FINISH	173
8.6.1.17	MODE_LEFT	173
8.6.1.18	MODE_RIGHT	173
8.6.1.19	MODE_TOP	173
8.6.1.20	PAD_WIDTH	173
8.6.1.21	PADDING_REPEAT	173
8.6.1.22	PI	173
8.6.1.23	PI2	173
8.6.1.24	QUTER_PI	173
8.6.1.25	SHIFT_LEFT	173
8.6.1.26	THR_QUTER_PI	173
8.7	ArithmeticCoding.h File Reference	174
8.8	BitInputStream.h File Reference	175
8.9	BitOutputStream.h File Reference	176
8.10	BoundingBox.h File Reference	177
8.10.1	Typedef Documentation	177
8.10.1.1	point	177
8.10.1.2	real_t	177
8.11	Buffer.h File Reference	177

8.12 CdvsClientBflog.h File Reference . . . . .	178
8.12.1 Function Documentation . . . . .	179
8.12.1.1 libcdvsbflog_is_present . . . . .	179
8.13 CdvsClientImpl.h File Reference . . . . .	179
8.14 CdvsClientLowMem.h File Reference . . . . .	180
8.15 CdvsConfigurationImpl.h File Reference . . . . .	181
8.16 CdvsDescriptor.h File Reference . . . . .	182
8.17 CdvsException.h File Reference . . . . .	183
8.18 CdvsInterface.h File Reference . . . . .	184
8.19 CdvsPoint.h File Reference . . . . .	186
8.20 CdvsServerImpl.h File Reference . . . . .	186
8.21 CsscCoordinateCoding.h File Reference . . . . .	187
8.22 Database.h File Reference . . . . .	188
8.23 DistratEigen.h File Reference . . . . .	189
8.24 Feature.h File Reference . . . . .	189
8.25 FeatureList.h File Reference . . . . .	191
8.26 FileManager.h File Reference . . . . .	192
8.27 HiResTimer.h File Reference . . . . .	192
8.28 ImageBuffer.h File Reference . . . . .	193
8.29 main.main File Reference . . . . .	193
8.30 map.h File Reference . . . . .	194
8.30.1 Macro Definition Documentation . . . . .	194
8.30.1.1 MAX_FILENAME_LENGTH . . . . .	194
8.30.1.2 MAX_MATCHES . . . . .	194
8.30.2 Typedef Documentation . . . . .	194
8.30.2.1 FILENAME . . . . .	194
8.30.3 Function Documentation . . . . .	194
8.30.3.1 alloc_qra_list . . . . .	194
8.30.3.2 free_qra_list . . . . .	194
8.30.3.3 mean_average_precision . . . . .	194
8.30.3.4 read_qra_list . . . . .	194
8.30.3.5 success_rate_for_top_match . . . . .	194
8.30.3.6 success_rate_for_top_match_2 . . . . .	194
8.30.3.7 write_qra_list . . . . .	194
8.31 Match.h File Reference . . . . .	195
8.32 Parameters.h File Reference . . . . .	195
8.33 platform.h File Reference . . . . .	196
8.33.1 Macro Definition Documentation . . . . .	196
8.33.1.1 assert . . . . .	196
8.34 platform.h File Reference . . . . .	197

8.35 platform.h File Reference . . . . .	197
8.35.1 Macro Definition Documentation . . . . .	197
8.35.1.1 assert . . . . .	197
8.36 PointPairs.h File Reference . . . . .	198
8.37 Points.h File Reference . . . . .	198
8.37.1 Typedef Documentation . . . . .	200
8.37.1.1 HomPoint2D . . . . .	200
8.37.1.2 HomPoint2DArray . . . . .	200
8.37.1.3 Point2D . . . . .	200
8.37.1.4 Point2DArray . . . . .	200
8.37.2 Function Documentation . . . . .	200
8.37.2.1 baricenter . . . . .	200
8.37.2.2 homogeneousPoints . . . . .	200
8.37.2.3 homogeneousPoints . . . . .	200
8.37.2.4 inhomogeneousPoints . . . . .	200
8.37.2.5 inhomogeneousPoints . . . . .	200
8.37.2.6 operator>> . . . . .	201
8.37.2.7 preconditionTransform . . . . .	201
8.37.2.8 preconditionTransform . . . . .	201
8.37.2.9 randomSubset . . . . .	201
8.37.2.10 read . . . . .	201
8.37.2.11 spread . . . . .	201
8.38 Projective2D.h File Reference . . . . .	201
8.39 SCFVIndex.h File Reference . . . . .	202
8.40 timer.h File Reference . . . . .	204
8.40.1 Function Documentation . . . . .	204
8.40.1.1 elapsed_time . . . . .	204
8.40.1.2 start_timer . . . . .	205
8.40.1.3 stop_timer . . . . .	205
8.41 TraceManager.h File Reference . . . . .	205
<b>Index</b>	<b>206</b>



# Chapter 1

## Documentation

### 1.1 Introduction

This is the documentation of the Test Model (TM) for MPEG Compact Descriptors for Visual Search.

The relevant MPEG documents related to CDVS are the following:

- N12201: Compact Descriptors for Visual Search: Call for Proposals
- N12202: Compact Descriptors for Visual Search: Evaluation Framework
- N11529: Compact Descriptors for Visual Search: Applications and Use Scenarios
- N11530: Compact Descriptors for Visual Search: Context and Objectives
- N11531: Compact Descriptors for Visual Search: Requirements

The software implements Architecture B for visual search applications, as described in N11529. It is composed by a suite of five executables:

- binaries for extraction and pairwise comparison:
  - `extract.exe`: extract CDVS descriptor from a set of images;
  - `match.exe`: match query and reference descriptors producing a similarity score;
- binaries for retrieval experiments:
  - `makeIndex.exe`: produces indices from a set of reference descriptors;
  - `joinIndices.exe`: joins a list of indices by merging their values into a single big index;
  - `retrieve.exe`: uses the global index to match a query image with the most similar images in the database;

All executables are built as 64-bit applications, which allows them to process a vast amount of data (please note that at least 8 GB of memory are needed to run the CDVS retrieval experiments). Moreover, the `extract` and `retrieve` executables are implemented as multi-threading processes in order to speed-up the execution of CDVS experiments.

In the following section, we describe the requirements for building and running the code (see [Requirements](#)), how to build the code (see [Building the code](#)), and finally how to run the CDVS experiments (see [Running the CDVS experiments](#)).

### 1.2 Requirements

### 1.2.1 Requirements for building

The CDVS code is entirely written in standard C and C++, and has been compiled on Windows 7 Enterprise 64-bit using Visual C++ 2010 (64 bit). The code can also be compiled on Linux 64-bit, as described in the Linux package files (see the README instructions in CDVS\_evaluation\_framework); also in this case the executables are all compiled as 64-bit applications, and provide comparable results (but not exactly the same results): differences in the order of 0.1% may be noted in pairwise matching and retrieval.

### 1.2.2 Requirements to run the code

All applications must run on 64-bit Windows or Linux, and running the retrieval binaries (makeIndex, joinIndices, retrieve) require at least 8 GB of physical memory installed on the processing machine. The other binaries require much less memory.

## 1.3 Building the code

Open the CDVS\_evaluation\_framework/deployment/all\_projects/all\_projects.sln solution using Visual Studio 2010 (64-bit); To build all applications, set:

- Solution Configurations: Release;
- Solution Platforms: x64;

and press F7 ("Build Solution"). This will compile all x64 applications and store the binaries in the "bin" directory.

The output results of the build process are the following:

- the binary executables for local testing in CDVS\_evaluation\_framework/bin/:
  - extract.exe
  - match.exe
  - makeIndex.exe
  - joinIndices.exe
  - retrieve.exe

## 1.4 Running the CDVS experiments

In order to run the MPEG CDVS experiments, first obtain the data set and the distractor set from the MPEG CDVS ad hoc group ([CDVS-TNT@listserv.uni-hannover.de](mailto:CDVS-TNT@listserv.uni-hannover.de)), which is composed by a set of one million JPEG images (about a Terabyte of data). Then, it is possible to run the experiments installing the data and code on a 64-bit workstation and using the scripts provided in CDVS\_evaluation\_framework/run/

In brief, the following steps must be performed:

- install the data package (data set and distractor set);
- install the annotation files
- install perl (this is needed in order to run the perl scripts)
- install the CDVS binaries into C:\bin
- modify the conf.pl Perl file in CDVS\_evaluation\_framework/run/ in order to match the location of your data:
  - edit \$datasetPath and set the correct data set directory;
  - edit \$annotationPath and set the correct annotation directory;

- run the full set of experiments
  - execute run-all-perl.bat
- get the results from the produced log files (the cdvsLogReader.jar tool can be used to produce a summary of results).

## 1.5 How to rebuild this documentation from source code

If you want to rebuild this documentation from source code, you have to install the following applications in C:\bin

- Doxygen from <http://www.stack.nl/~dimitri/doxygen/>
- Graphviz from <http://www.graphviz.org/>

Then go to CDVS\_evaluation\_framework/Doxygen/ and click on "createDoc.bat", it will create the html files in a subdirectory named "html". Locate the file "index.html" and open it in a browser to see the produced documentation.

## 1.6 Acknowledgements

The MPEG CDVS working group would like to thank all those developers providing the following great libraries in open source to the scientific community, who allowed the group to produce this CDVS implementation:

- jpeg-8c (<http://www.ijg.org/>) to read JPEG images;
- imageresampler 1.0.7 (<http://code.google.com/p/imageresampler/>) to scale down the images to a reasonable resolution;
- vlfeat 0.9.14 (<http://www.vlfeat.org>) to extract the keypoints and the SIFT descriptors from scaled images;
- ari 2014 (<http://marknelson.us/2014/10/19/data-compression-with-arithmetic-coding>) to perform adaptive arithmetic coding for the compression of coordinates;
- eigen 3.2.0 (<http://eigen.tuxfamily.org/index.php>) to perform some linear algebra for the localization.

In particular, we would like to thank Andrea Vedaldi from the Oxford Visual Geometry Group who kindly agreed to release his VLFeat library under the very liberal terms of the BSD license (<http://www.vlfeat.org/license.html>).





## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[mpeg7cdvs](#)

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS

Library headers (in particular [CdvslInterface.h](#)) are included . . . . . 15



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

mpeg7cdvs::AbstractDetector . . . . .	21
mpeg7cdvs::ImageBuffer . . . . .	132
mpeg7cdvs::AlpDetector . . . . .	30
mpeg7cdvs::AlpDetectorBF . . . . .	32
mpeg7cdvs::AlpDetectorLowMem . . . . .	34
mpeg7cdvs::AC_decoder . . . . .	23
mpeg7cdvs::AC_encoder . . . . .	25
mpeg7cdvs::AC_model . . . . .	26
mpeg7cdvs::AlpOctave . . . . .	36
mpeg7cdvs::AlpOctaveBF . . . . .	41
Ball2D . . . . .	46
mpeg7cdvs::BitInputStream . . . . .	46
mpeg7cdvs::BitOutputStream . . . . .	50
BoundingBox . . . . .	54
mpeg7cdvs::Buffer . . . . .	55
mpeg7cdvs::CdvsClient . . . . .	58
mpeg7cdvs::CdvsClientImpl . . . . .	62
mpeg7cdvs::CdvsClientBflog . . . . .	60
mpeg7cdvs::CdvsClientLowMem . . . . .	64
mpeg7cdvs::CdvsConfiguration . . . . .	66
mpeg7cdvs::CdvsConfigurationImpl . . . . .	69
mpeg7cdvs::CdvsDescriptor . . . . .	71
mpeg7cdvs::CDVSPPOINT . . . . .	78
mpeg7cdvs::CdvsServer . . . . .	78
mpeg7cdvs::CdvsServerImpl . . . . .	85
mpeg7cdvs::CsscCoordinateCoding::CircularSumContext . . . . .	91
mpeg7cdvs::CompressedFeatureList . . . . .	91
mpeg7cdvs::CsscCoordinateCoding . . . . .	98
mpeg7cdvs::Database . . . . .	102
DistratEigen . . . . .	108
exception . . . . .	
mpeg7cdvs::CdvsException . . . . .	76
mpeg7cdvs::Feature . . . . .	109
mpeg7cdvs::FeatureAlp . . . . .	112
mpeg7cdvs::FeatureAlpBF . . . . .	115
mpeg7cdvs::FeatureList . . . . .	117

FileManager . . . . .	124
mpeg7cdvs::Filter . . . . .	127
mpeg7cdvs::FrequencyFilter . . . . .	128
HiResTimer . . . . .	130
mpeg7cdvs::LookUpTable . . . . .	137
Match . . . . .	138
mpeg7cdvs::Parameters . . . . .	140
mpeg7cdvs::PointPairs . . . . .	145
Polygon . . . . .	150
Projective2D . . . . .	150
QRA . . . . .	153
mpeg7cdvs::RetrievalData . . . . .	154
mpeg7cdvs::SCFVFactory . . . . .	155
mpeg7cdvs::SCFVIndex . . . . .	156
mpeg7cdvs::SCFVSignature . . . . .	160
TM_COUNTER . . . . .	163
TM_STATE . . . . .	164
TraceManager . . . . .	164

## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">mpeg7cdvs::AbstractDetector</a>	21
Base class for keypoint detectors . . . . .	
<a href="#">mpeg7cdvs::AC_decoder</a>	23
The decoder using the Arithmetic Coder Model . . . . .	
<a href="#">mpeg7cdvs::AC_encoder</a>	25
The encoder using the Arithmetic Coding model . . . . .	
<a href="#">mpeg7cdvs::AC_model</a>	26
Arithmetic Coding model to be used when encoding or decoding a symbol . . . . .	
<a href="#">mpeg7cdvs::AlpDetector</a>	30
Implementation of the ALP keypoint detector . . . . .	
<a href="#">mpeg7cdvs::AlpDetectorBF</a>	32
Implementation of the ALP keypoint detector by employing block-based processing and frequency domain filtering . . . . .	
<a href="#">mpeg7cdvs::AlpDetectorLowMem</a>	34
Low memory implementation of the ALP keypoint detector . . . . .	
<a href="#">mpeg7cdvs::AlpOctave</a>	36
A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points . . . . .	
<a href="#">mpeg7cdvs::AlpOctaveBF</a>	41
A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points . . . . .	
<a href="#">Ball2D</a>	46
<a href="#">mpeg7cdvs::BitInputStream</a>	46
This class represents an input stream of bits . . . . .	
<a href="#">mpeg7cdvs::BitOutputStream</a>	50
This class represents an output stream of bits . . . . .	
<a href="#">BoundingBox</a>	54
A class containing four points which identify the object in the image . . . . .	
<a href="#">mpeg7cdvs::Buffer</a>	55
A container class for a byte array, intended to replace all malloc() and new() instructions in the main code . . . . .	
<a href="#">mpeg7cdvs::CdvsClient</a>	58
Interface to the client-side functionality of the CDVS Library . . . . .	
<a href="#">mpeg7cdvs::CdvsClientBflog</a>	60
Block-based frequency domain Laplacian of Gaussian implementation of the high level interface to the client-side functionality of the CDVS Library . . . . .	
<a href="#">mpeg7cdvs::CdvsClientImpl</a>	62
Main implementation of the high level interface to the client-side functionality of the CDVS Library . . . . .	

<a href="#">mpeg7cdvs::CdvsClientLowMem</a>	
Low Memory implementation of the high level interface to the client-side functionality of the CDVS Library . . . . .	64
<a href="#">mpeg7cdvs::CdvsConfiguration</a>	
Interface to all configuration parameters for clients and servers . . . . .	66
<a href="#">mpeg7cdvs::CdvsConfigurationImpl</a>	
Interface to all configuration parameters for clients and servers . . . . .	69
<a href="#">mpeg7cdvs::CdvsDescriptor</a>	
Helper class to read/write/check CDVS descriptors according to the syntax defined in ISO/IEC 15938-13 . . . . .	71
<a href="#">mpeg7cdvs::CdvsException</a>	
Class defining a specific exception for CDVS . . . . .	76
<a href="#">mpeg7cdvs::CDVSPPOINT</a>	
A structure containing the x and y coordinate of a point in the image . . . . .	78
<a href="#">mpeg7cdvs::CdvsServer</a>	
Interface to the server-side functionality of the CDVS Library . . . . .	78
<a href="#">mpeg7cdvs::CdvsServerImpl</a>	
Implementation of the high level interface to the server-side functionality of the CDVS Library . . . . .	85
<a href="#">mpeg7cdvs::CsscCoordinateCoding::CircularSumContext</a>	
Basic structure for Cssc . . . . .	91
<a href="#">mpeg7cdvs::CompressedFeatureList</a>	
Container class for all compressed features of an image . . . . .	91
<a href="#">mpeg7cdvs::CsscCoordinateCoding</a>	
Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa . . . . .	98
<a href="#">mpeg7cdvs::Database</a>	
The image database implementation containing helper methods for image retrieval . . . . .	102
<a href="#">DistratEigen</a>	
Simplified and faster version of DISTRAT, based on the Eigen library . . . . .	108
<a href="#">mpeg7cdvs::Feature</a>	
Container class for the features of a single point (storing coordinates, scale, orientation, peak and descriptor of a point) . . . . .	109
<a href="#">mpeg7cdvs::FeatureAlp</a>	
Definition of keypoint for Alp . . . . .	112
<a href="#">mpeg7cdvs::FeatureAlpBF</a>	
Inherits all member variables declared in <a href="#">FeatureAlp</a> and adds two new member variables . . . . .	115
<a href="#">mpeg7cdvs::FeatureList</a>	
Container class for all features of an image . . . . .	117
<a href="#">FileManager</a>	
Helper class to manage lists of file names . . . . .	124
<a href="#">mpeg7cdvs::Filter</a>	
A class containing a separable Gaussian filter kernel . . . . .	127
<a href="#">mpeg7cdvs::FrequencyFilter</a>	
. . . . .	128
<a href="#">HiResTimer</a>	
C++ wrapper class for the C functions implementing the high resolution timer . . . . .	130
<a href="#">mpeg7cdvs::ImageBuffer</a>	
A container class for a bidimensional image; it's the base class of all keypoint detector classes . . . . .	132
<a href="#">mpeg7cdvs::LookUpTable</a>	
A simple look up table implementation, to perform a bit count very quickly . . . . .	137
<a href="#">Match</a>	
Helper class to sort images according to a pair of values . . . . .	138
<a href="#">mpeg7cdvs::Parameters</a>	
Container for all encoding/decoding parameters associated to each target bitrate defined by M-PEG CDVS . . . . .	140
<a href="#">mpeg7cdvs::PointPairs</a>	
Parameter class, used to pass around matched point coordinates . . . . .	145
<a href="#">Polygon</a>	
A base class for all polygons; . . . . .	150

<a href="#">Projective2D</a>	Computes and performs homographies between two images . . . . .	150
<a href="#">QRA</a>	Structure containing the name of a query image and the list of names of matching reference pictures . . . . .	153
<a href="#">mpeg7cdvs::RetrievalData</a>	A structure containing the output of a retrieval operation . . . . .	154
<a href="#">mpeg7cdvs::SCFVFactory</a>	A class to produce SCFV signatures . . . . .	155
<a href="#">mpeg7cdvs::SCFVIndex</a>	A class to manage an indexed list of SCFV signatures . . . . .	156
<a href="#">mpeg7cdvs::SCFVSignature</a>	Container class for a Scalable Fisher Vector binary signature; allows reading/writing from/to a bitstream, fetching/storing from/into a file, and comparing a signature with another . . . . .	160
<a href="#">TM_COUNTER</a>	Structure used by the high precision timer . . . . .	163
<a href="#">TM_STATE</a>	Structure used by the high precision timer . . . . .	164
<a href="#">TraceManager</a>	Helper class to manage trace files in text or XML format . . . . .	164





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">AbstractDetector.h</a>	167
<a href="#">AlpDetector.h</a>	168
<a href="#">AlpDetectorBF.h</a>	168
<a href="#">AlpDetectorLowMem.h</a>	169
<a href="#">AlpOctave.h</a>	170
<a href="#">AlpOctaveBF.h</a>	171
<a href="#">ArithmeticCoding.h</a>	174
<a href="#">BitInputStream.h</a>	175
<a href="#">BitOutputStream.h</a>	176
<a href="#">BoundingBox.h</a>	177
<a href="#">Buffer.h</a>	177
<a href="#">CdvsClientBflog.h</a>	178
<a href="#">CdvsClientImpl.h</a>	179
<a href="#">CdvsClientLowMem.h</a>	180
<a href="#">CdvsConfigurationImpl.h</a>	181
<a href="#">CdvsDescriptor.h</a>	182
<a href="#">CdvsException.h</a>	183
<a href="#">CdvsInterface.h</a>	184
<a href="#">CdvsPoint.h</a>	186
<a href="#">CdvsServerImpl.h</a>	186
<a href="#">CsscCoordinateCoding.h</a>	187
<a href="#">Database.h</a>	188
<a href="#">DistratEigen.h</a>	189
<a href="#">Feature.h</a>	189
<a href="#">FeatureList.h</a>	191
<a href="#">FileManager.h</a>	192
<a href="#">HiResTimer.h</a>	192
<a href="#">ImageBuffer.h</a>	193
<a href="#">main.main</a>	193
<a href="#">map.h</a>	194
<a href="#">Match.h</a>	195
<a href="#">Parameters.h</a>	195
<a href="#">Symbian/platform.h</a>	196
<a href="#">w2000/platform.h</a>	197
<a href="#">WCE/platform.h</a>	197
<a href="#">PointPairs.h</a>	198
<a href="#">Points.h</a>	198
<a href="#">Projective2D.h</a>	201

SCFVIndex.h . . . . .	202
timer.h . . . . .	204
TraceManager.h . . . . .	205

## Chapter 6

# Namespace Documentation

### 6.1 mpeg7cdvs Namespace Reference

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.

#### Data Structures

- class [AbstractDetector](#)  
*Base class for keypoint detectors.*
- class [AlpDetector](#)  
*Implementation of the ALP keypoint detector.*
- class [AlpDetectorBF](#)  
*implementation of the ALP keypoint detector by employing block-based processing and frequency domain filtering.*
- class [AlpDetectorLowMem](#)  
*Low memory implementation of the ALP keypoint detector.*
- class [FeatureAlp](#)  
*Definition of keypoint for Alp.*
- class [Filter](#)  
*A class containing a separable Gaussian filter kernel.*
- class [AlpOctave](#)  
*A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.*
- class [FrequencyFilter](#)
- class [FeatureAlpBF](#)  
*Inherits all member variables declared in [FeatureAlp](#) and adds two new member variables.*
- class [AlpOctaveBF](#)  
*A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.*
- class [AC\\_model](#)  
*Arithmetic Coding model to be used when encoding or decoding a symbol.*
- class [AC\\_encoder](#)  
*The encoder using the Arithmetic Coding model.*
- class [AC\\_decoder](#)  
*The decoder using the Arithmetic Coder Model.*
- class [Buffer](#)  
*A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.*
- class [CdvDescriptor](#)  
*Helper class to read/write/check CDVS descriptors according to the syntax defined in ISO/IEC 15938-13.*

- class [CdvsException](#)  
*Class defining a specific exception for CDVS.*
- struct [CDVSPPOINT](#)  
*A structure containing the x and y coordinate of a point in the image.*
- struct [RetrievalData](#)  
*A structure containing the output of a retrieval operation.*
- class [CsscCoordinateCoding](#)  
*Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.*
- class [Database](#)  
*The image database implementation containing helper methods for image retrieval.*
- class [Feature](#)  
*Container class for the features of a single point (storing coordinates, scale, orientation, peak and descriptor of a point).*
- class [FeatureList](#)  
*Container class for all features of an image.*
- class [CompressedFeatureList](#)  
*Container class for all compressed features of an image.*
- class [ImageBuffer](#)  
*A container class for a bidimensional image; it's the base class of all keypoint detector classes.*
- class [Parameters](#)  
*Container for all encoding/decoding parameters associated to each target bitrate defined by MPEG CDVS.*
- class [PointPairs](#)  
*Parameter class, used to pass around matched point coordinates.*
- class [LookUpTable](#)  
*A simple look up table implementation, to perform a bit count very quickly.*
- class [SCFVSignature](#)  
*Container class for a Scalable Fisher Vector binary signature; allows reading/writing from/to a bitstream, fetching/storing from/into a file, and comparing a signature with another.*
- class [SCFVIndex](#)  
*A class to manage an indexed list of SCFV signatures.*
- class [SCFVFactory](#)  
*A class to produce SCFV signatures.*
- class [CdvsClientBflog](#)  
*Block-based frequency domain Laplacian of Gaussian implementation of the high level interface to the client-side functionality of the CDVS Library.*
- class [CdvsClientImpl](#)  
*Main implementation of the high level interface to the client-side functionality of the CDVS Library.*
- class [CdvsClientLowMem](#)  
*Low Memory implementation of the high level interface to the client-side functionality of the CDVS Library.*
- class [CdvsConfigurationImpl](#)  
*Interface to all configuration parameters for clients and servers.*
- class [CdvsConfiguration](#)  
*Interface to all configuration parameters for clients and servers.*
- class [CdvsClient](#)  
*Interface to the client-side functionality of the CDVS Library.*
- class [CdvsServer](#)  
*Interface to the server-side functionality of the CDVS Library.*
- class [CdvsServerImpl](#)  
*Implementation of the high level interface to the server-side functionality of the CDVS Library.*
- class [BitInputStream](#)  
*This class represents an input stream of bits.*
- class [BitOutputStream](#)  
*This class represents an output stream of bits.*

## Typedefs

- typedef float [input\\_type](#)
- typedef float [gs\\_type](#)
- typedef float [element\\_type](#)
- typedef float [filter\\_type](#)
- typedef unsigned long long [CODE\\_VALUE](#)
- typedef std::vector< unsigned int > [recallGraphNode\\_t](#)
- typedef std::vector  
    < [recallGraphNode\\_t](#) > [recallGraph\\_t](#)
- typedef [Parameters](#) [ParameterSet](#) [[Parameters::nModes](#)]

## Enumerations

- enum { [MATCH\\_TYPE\\_DEFAULT](#) = 0, [MATCH\\_TYPE\\_BOTH](#) = 1, [MATCH\\_TYPE\\_LOCAL](#) = 2, [MATCH\\_TYPE\\_GLOBAL](#) = 3 }
- Type of matching.*
- enum { [match\\_2way\\_INTERSECTION](#) = 0, [match\\_2way\\_DISJOINT1](#) = 1, [match\\_2way\\_DISJOINT2](#) = 2 }

## Variables

- static const int [PRECISION](#) = 64
- static const int [CODE\\_VALUE\\_BITS](#) = 18
- static const int [FREQUENCY\\_BITS](#) = [CODE\\_VALUE\\_BITS](#) - 2
- static const [CODE\\_VALUE](#) [MAX\\_CODE](#) = (1 << [CODE\\_VALUE\\_BITS](#)) - 1
- static const [CODE\\_VALUE](#) [MAX\\_FREQ](#) = (1 << [FREQUENCY\\_BITS](#)) - 1
- static const [CODE\\_VALUE](#) [TOP\\_VALUE](#) = [MAX\\_CODE](#)
- static const [CODE\\_VALUE](#) [ONE\\_FOURTH](#) = 1 << ([CODE\\_VALUE\\_BITS](#) - 2)
- static const [CODE\\_VALUE](#) [ONE\\_HALF](#) = 2 \* [ONE\\_FOURTH](#)
- static const [CODE\\_VALUE](#) [THREE\\_FOURTHS](#) = 3 \* [ONE\\_FOURTH](#)
- static const size\_t [NOT\\_FOUND](#) = std::numeric\_limits<size\_t>::max()
- static const float [gama](#) = 0.3f
- static const int [num\\_bit\\_selection](#) = 24
- static const int [PCASiftLength](#) = 32
- number of principal components in the centroid space*
- static const int [numberCentroids](#) = 512
- number of centroids of the codebook*

### 6.1.1 Detailed Description

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.

### 6.1.2 Typedef Documentation

6.1.2.1 typedef unsigned long long mpeg7cdvs::CODE\_VALUE

6.1.2.2 typedef float mpeg7cdvs::element\_type

6.1.2.3 typedef float mpeg7cdvs::filter\_type

6.1.2.4 typedef float mpeg7cdvs::gs\_type

6.1.2.5 `typedef float mpeg7cdvs::input_type`

6.1.2.6 `typedef Parameters mpeg7cdvs::ParameterSet[Parameters::nModes]`

6.1.2.7 `typedef std::vector<recallGraphNode_t> mpeg7cdvs::recallGraph_t`

6.1.2.8 `typedef std::vector<unsigned int> mpeg7cdvs::recallGraphNode_t`

### 6.1.3 Enumeration Type Documentation

6.1.3.1 anonymous enum

Type of matching.

Enumerator

***MATCH\_TYPE\_DEFAULT*** ignore global if local match

***MATCH\_TYPE\_BOTH*** compute both local and global matching scores

***MATCH\_TYPE\_LOCAL*** compute only local matching score

***MATCH\_TYPE\_GLOBAL*** compute only global matching score

6.1.3.2 anonymous enum

Enumerator

***match\_2way\_INTERSECTION*** indicates 2-direction matching pair

***match\_2way\_DISJOINT1*** indicates 1-direction matching pair: direction A=>B

***match\_2way\_DISJOINT2*** indicates 1-direction matching pair: direction B=>A

### 6.1.4 Variable Documentation

6.1.4.1 `const int mpeg7cdvs::CODE_VALUE_BITS = 18` [static]

6.1.4.2 `const int mpeg7cdvs::FREQUENCY_BITS = CODE_VALUE_BITS - 2` [static]

6.1.4.3 `const float mpeg7cdvs::gama = 0.3f` [static]

6.1.4.4 `const CODE_VALUE mpeg7cdvs::MAX_CODE = (1 << CODE_VALUE_BITS) - 1` [static]

6.1.4.5 `const CODE_VALUE mpeg7cdvs::MAX_FREQ = (1 << FREQUENCY_BITS) - 1` [static]

6.1.4.6 `const size_t mpeg7cdvs::NOT_FOUND = std::numeric_limits<size_t>::max()` [static]

6.1.4.7 `const int mpeg7cdvs::num_bit_selection = 24` [static]

6.1.4.8 `const int mpeg7cdvs::numberCentroids = 512` [static]

number of centroids of the codebook

6.1.4.9 `const CODE_VALUE mpeg7cdvs::ONE_FOURTH = 1 << (CODE_VALUE_BITS - 2)` [static]

6.1.4.10 `const CODE_VALUE mpeg7cdvs::ONE_HALF = 2 * ONE_FOURTH` [static]

6.1.4.11 `const int mpeg7cdvs::PCASiftLength = 32` `[static]`

number of principal components in the centroid space

6.1.4.12 `const int mpeg7cdvs::PRECISION = 64` `[static]`

6.1.4.13 `const CODE_VALUE mpeg7cdvs::THREE_FOURTHS = 3 * ONE_FOURTH` `[static]`

6.1.4.14 `const CODE_VALUE mpeg7cdvs::TOP_VALUE = MAX_CODE` `[static]`





## Chapter 7

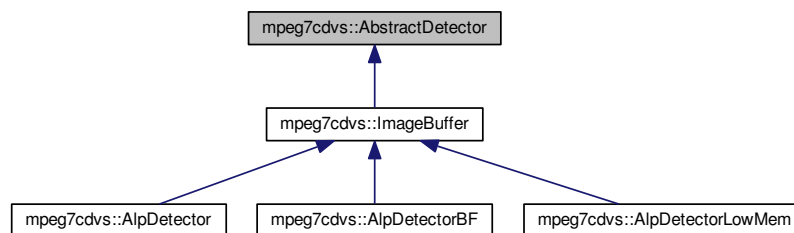
# Data Structure Documentation

### 7.1 mpeg7cdvs::AbstractDetector Class Reference

Base class for keypoint detectors.

```
#include <AbstractDetector.h>
```

Inheritance diagram for mpeg7cdvs::AbstractDetector:



#### Public Member Functions

- `AbstractDetector ()`
- `virtual ~AbstractDetector ()`
- `virtual void detect (FeatureList &featurelist, const Parameters &params)=0`  
*Detect all keypoints from this image.*
- `virtual void extract (FeatureList &featurelist, size_t num) const =0`  
*Extract the SIFT descriptor of each keypoint and store it back in featurelist.*

#### Data Fields

- `int height`  
*height of the image*
- `int width`  
*width of the image*
- `int originalWidth`  
*original width of the image (before any resample operation)*
- `int originalHeight`

*original height of the image (before any resample operation)*

### 7.1.1 Detailed Description

Base class for keypoint detectors.

#### Author

Massimo Balestri

#### Date

2014

### 7.1.2 Constructor & Destructor Documentation

7.1.2.1 `mpeg7cdvs::AbstractDetector::AbstractDetector ( )` `[inline]`

7.1.2.2 `virtual mpeg7cdvs::AbstractDetector::~~AbstractDetector ( )` `[inline]`, `[virtual]`

### 7.1.3 Member Function Documentation

7.1.3.1 `virtual void mpeg7cdvs::AbstractDetector::detect ( FeatureList & featurelist, const Parameters & params )`  
`[pure virtual]`

Detect all keypoints from this image.

#### Parameters

<i>featurelist</i>	the ouput list of keypoints with their associated features, in descending order of importance.
<i>params</i>	the running parameters.

#### Exceptions

<a href="#"><i>CdvsException</i></a>	in case of error
--------------------------------------	------------------

Implemented in [mpeg7cdvs::AlpDetector](#), [mpeg7cdvs::AlpDetectorBF](#), and [mpeg7cdvs::AlpDetectorLowMem](#).

7.1.3.2 `virtual void mpeg7cdvs::AbstractDetector::extract ( FeatureList & featurelist, size_t num ) const` `[pure virtual]`

Extract the SIFT descriptor of each keypoint and store it back in featurelist.

#### Parameters

<i>featurelist</i>	the detected keypoints
<i>num</i>	the absolute maximum number of features to be extracted from this image

Implemented in [mpeg7cdvs::AlpDetector](#), [mpeg7cdvs::AlpDetectorBF](#), and [mpeg7cdvs::AlpDetectorLowMem](#).

### 7.1.4 Field Documentation

7.1.4.1 `int mpeg7cdvs::AbstractDetector::height`

height of the image

## 7.1.4.2 int mpeg7cdvs::AbstractDetector::originalHeight

original height of the image (before any resample operation)

## 7.1.4.3 int mpeg7cdvs::AbstractDetector::originalWidth

original width of the image (before any resample operation)

## 7.1.4.4 int mpeg7cdvs::AbstractDetector::width

width of the image

The documentation for this class was generated from the following file:

- [AbstractDetector.h](#)

## 7.2 mpeg7cdvs::AC\_decoder Class Reference

The decoder using the Arithmetic Coder Model.

```
#include <ArithmeticCoding.h>
```

### Public Member Functions

- void [init](#) ([BitInputStream](#) &reader)  
*Initialize the decoder using the given input buffer reader.*
- int [decode\\_symbol](#) ([AC\\_model](#) &model)  
*Decode a symbol using the given Arithmetic Coding model.*
- long [bits](#) () const  
*Get the number of bits currently decoded.*
- void [done](#) ()  
*Terminate reading from the input buffer.*

### 7.2.1 Detailed Description

The decoder using the Arithmetic Coder Model.

#### Author

Giovanni Cordara, Massimo Balestri

#### Date

2012

### 7.2.2 Member Function Documentation

## 7.2.2.1 long mpeg7cdvs::AC\_decoder::bits ( ) const

Get the number of bits currently decoded.

#### Returns

the number of bits read from the bitstream.

7.2.2.2 `int mpeg7cdvs::AC_decoder::decode_symbol ( AC_model & model )`

Decode a symbol using the given Arithmetic Coding model.

## Parameters

<i>model</i>	the decoding model to be used.
--------------	--------------------------------

## Returns

the decoded symbol.

## 7.2.2.3 void mpeg7cdvs::AC\_decoder::done ( )

Terminate reading from the input buffer.

## 7.2.2.4 void mpeg7cdvs::AC\_decoder::init ( BitInputStream &amp; reader )

Initialize the decoder using the given input buffer reader.

## Parameters

<i>reader</i>	the input buffer reader
---------------	-------------------------

The documentation for this class was generated from the following file:

- [ArithmeticCoding.h](#)

## 7.3 mpeg7cdvs::AC\_encoder Class Reference

The encoder using the Arithmetic Coding model.

```
#include <ArithmeticCoding.h>
```

### Public Member Functions

- void [init](#) ([BitOutputStream](#) &writer)  
*Initialize the decoder using the given output buffer writer.*
- void [encode\\_symbol](#) ([AC\\_model](#) &model, int symbol)  
*Encode a symbol using the given Arithmetic Coding model.*
- long [bits](#) () const  
*Get the number of bits currently encoded.*
- void [done](#) ()  
*Terminate writing into the output buffer.*

### 7.3.1 Detailed Description

The encoder using the Arithmetic Coding model.

## Author

Giovanni Cordara, Massimo Balestri

## Date

2012

### 7.3.2 Member Function Documentation

#### 7.3.2.1 `long mpeg7cdvs::AC_encoder::bits ( ) const`

Get the number of bits currently encoded.

##### Returns

the number of bits written into the bitstream.

#### 7.3.2.2 `void mpeg7cdvs::AC_encoder::done ( )`

Terminate writing into the output buffer.

#### 7.3.2.3 `void mpeg7cdvs::AC_encoder::encode_symbol ( AC_model & model, int symbol )`

Encode a symbol using the given Arithmetic Coding model.

##### Parameters

<i>model</i>	the decoding model to be used.
<i>symbol</i>	the symbol to encode.

#### 7.3.2.4 `void mpeg7cdvs::AC_encoder::init ( BitOutputStream & writer )`

Initialize the decoder using the given output buffer writer.

##### Parameters

<i>writer</i>	the output buffer writer.
---------------	---------------------------

The documentation for this class was generated from the following file:

- [ArithmeticCoding.h](#)

## 7.4 mpeg7cdvs::AC\_model Class Reference

Arithmetic Coding model to be used when encoding or decoding a symbol.

```
#include <ArithmeticCoding.h>
```

### Public Member Functions

- [AC\\_model](#) ()
- virtual [~AC\\_model](#) ()
- void [init](#) (int nsym, long \*cfreq)  
*Initialize an Arithmetic Coder model using the given accumulated frequency values.*
- void [init](#) (int nsym, int \*ifreq, int adapt)  
*Initialize an Arithmetic Coder model using the given input frequency values.*
- void [update](#) (int symbol)  
*update model with the given symbol*
- void [done](#) ()  
*Terminate using this Arithmetic Coder model.*

- void `print` ()  
*Print the model constants.*
- void `check` (int symbol) const  
*Check if the symbol belongs to the symbol range of this model.*
- int `high` (int symbol) const  
*Get the higher accumulated frequency of this symbol.*
- int `low` (int symbol) const  
*Get the lower accumulated frequency of this symbol.*
- int `count` () const  
*Get the total accumulated frequency.*
- int `getSymbol` (CODE\_VALUE cum) const  
*Get the symbol corresponding to a specific code value.*
- bool `adaptative` () const  
*Get the current mode.*

### 7.4.1 Detailed Description

Arithmetic Coding model to be used when encoding or decoding a symbol.

#### Author

Giovanni Cordara, Massimo Balestri

#### Date

2012

### 7.4.2 Constructor & Destructor Documentation

7.4.2.1 `mpeg7cdvs::AC_model::AC_model ( )`

7.4.2.2 `virtual mpeg7cdvs::AC_model::~~AC_model ( )` `[virtual]`

### 7.4.3 Member Function Documentation

7.4.3.1 `bool mpeg7cdvs::AC_model::adaptative ( ) const` `[inline]`

Get the current mode.

#### Returns

true if currently in adaptative mode

7.4.3.2 `void mpeg7cdvs::AC_model::check ( int symbol ) const`

Check if the symbol belongs to the symbol range of this model.

#### Parameters

<i>symbol</i>	the symbol to check
---------------	---------------------

## Exceptions

<a href="#">CdvsException</a>	
-------------------------------	--

**7.4.3.3** `int mpeg7cdvs::AC_model::count ( ) const` `[inline]`

Get the total accumulated frequency.

## Returns

the total accumulated frequency

**7.4.3.4** `void mpeg7cdvs::AC_model::done ( )`

Terminate using this Arithmetic Coder model.

**7.4.3.5** `int mpeg7cdvs::AC_model::getSymbol ( CODE_VALUE cum ) const` `[inline]`

Get the symbol corresponding to a specific code value.

## Parameters

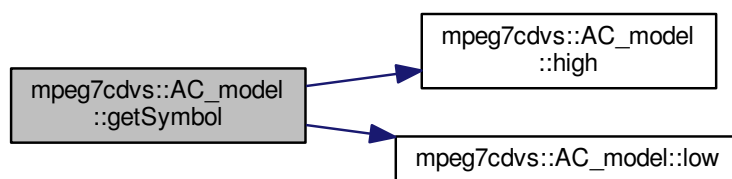
<i>cum</i>	the accumulated code value
------------	----------------------------

## Returns

the symbol

References `high()`, and `low()`.

Here is the call graph for this function:



**7.4.3.6** `int mpeg7cdvs::AC_model::high ( int symbol ) const` `[inline]`

Get the higher accumulated frequency of this symbol.

## Parameters



<i>symbol</i>	the input symbol
---------------	------------------

**Returns**

the higher value of the symbol

Referenced by `getSymbol()`.

**7.4.3.7 void mpeg7cdvs::AC\_model::init ( int *nsym*, long \* *cfreq* )**

Initialize an Arithmetic Coder model using the given accumulated frequency values.

Note that adaptation is disabled in this case.

**Parameters**

<i>nsym</i>	number of symbols to be encoded/decoded
<i>cfreq</i>	the cumulative frequency of all symbols

**7.4.3.8 void mpeg7cdvs::AC\_model::init ( int *nsym*, int \* *ifreq*, int *adapt* )**

Initialize an Arithmetic Coder model using the given input frequency values.

**Parameters**

<i>nsym</i>	number of symbols to be encoded/decoded;
<i>ifreq</i>	the input frequency table, if NULL a frequency of one for all elements will be assumed;
<i>adapt</i>	flag indicating if dynamic adaptation must be performed during encoding/decoding

**7.4.3.9 int mpeg7cdvs::AC\_model::low ( int *symbol* ) const [inline]**

Get the lower accumulated frequency of this symbol.

**Parameters**

<i>symbol</i>	the input symbol
---------------	------------------

**Returns**

the lower value of the symbol

Referenced by `getSymbol()`.

**7.4.3.10 void mpeg7cdvs::AC\_model::print ( )**

Print the model constants.

**7.4.3.11 void mpeg7cdvs::AC\_model::update ( int *symbol* )**

update model with the given symbol

## Parameters

<i>symbol</i>	the symbol to update
---------------	----------------------

The documentation for this class was generated from the following file:

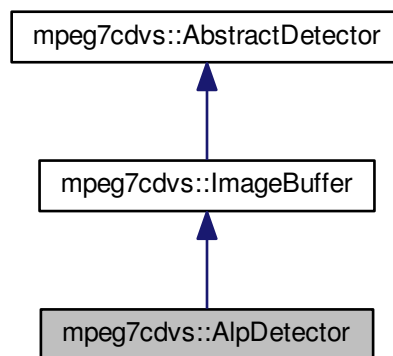
- [ArithmeticCoding.h](#)

## 7.5 mpeg7cdvs::AlpDetector Class Reference

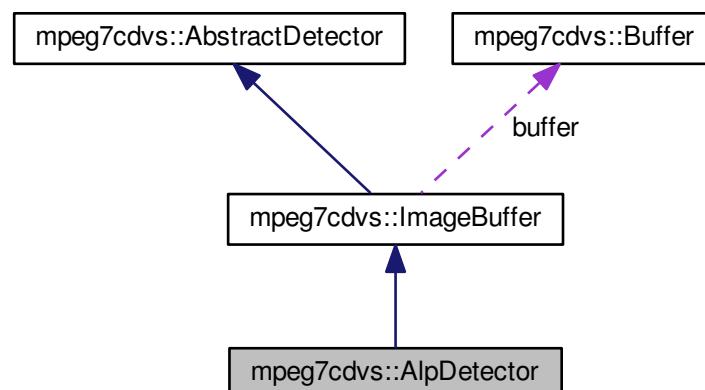
Implementation of the ALP keypoint detector.

```
#include <AlpDetector.h>
```

Inheritance diagram for mpeg7cdvs::AlpDetector:



Collaboration diagram for mpeg7cdvs::AlpDetector:



## Public Member Functions

- [AlpDetector](#) ()
- virtual [~AlpDetector](#) ()
- void [detect](#) ([FeatureList](#) &featurelist, const [Parameters](#) &params)  
*Detect all ALP keypoints from this image.*
- void [extract](#) ([FeatureList](#) &featurelist, size\_t num) const  
*Extract the SIFT descriptor of each keypoint and store it back in featurelist.*

## Static Public Member Functions

- static void [sortPdf](#) ([FeatureList](#) &featurelist)  
*Sort the given ALP features in descending order of importance.*
- static void [sortPdf](#) (std::vector< [FeatureAlp](#) > &alpFeatures, float imageWidth, float imageHeight)  
*Sort the given ALP features in descending order of importance.*

## Additional Inherited Members

### 7.5.1 Detailed Description

Implementation of the ALP keypoint detector.

#### Author

Gianluca Francini, Massimo Balestri

#### Date

2013

### 7.5.2 Constructor & Destructor Documentation

7.5.2.1 [mpeg7cdvs::AlpDetector::AlpDetector](#) ( )

7.5.2.2 virtual [mpeg7cdvs::AlpDetector::~~AlpDetector](#) ( ) `[virtual]`

### 7.5.3 Member Function Documentation

7.5.3.1 void [mpeg7cdvs::AlpDetector::detect](#) ( [FeatureList](#) & *featurelist*, const [Parameters](#) & *params* ) `[virtual]`

Detect all ALP keypoints from this image.

#### Parameters

<i>featurelist</i>	the ouput list of keypoints with their associated features.
<i>params</i>	the running parameters.

#### Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

Implements [mpeg7cdvs::AbstractDetector](#).

7.5.3.2 void [mpeg7cdvs::AlpDetector::extract](#) ( [FeatureList](#) & *featurelist*, size\_t *num* ) const `[virtual]`

Extract the SIFT descriptor of each keypoint and store it back in featurelist.

## Parameters

<i>featurelist</i>	the detected keypoints
<i>num</i>	the absolute maximum number of features to be extracted from this image

Implements [mpeg7cdvs::AbstractDetector](#).

7.5.3.3 `static void mpeg7cdvs::AlpDetector::sortPdf ( FeatureList & featurelist ) [static]`

Sort the given ALP features in descending order of importance.

## Parameters

<i>featurelist</i>	the detected keypoints
--------------------	------------------------

7.5.3.4 `static void mpeg7cdvs::AlpDetector::sortPdf ( std::vector< FeatureAlp > & alpFeatures, float imageWidth, float imageHeight ) [static]`

Sort the given ALP features in descending order of importance.

## Parameters

<i>alpFeatures</i>	the detected keypoints (without descriptor information and/or storage)
<i>imageWidth</i>	the width of the (possibly resampled) image
<i>imageHeight</i>	the height of the (possibly resampled) image

The documentation for this class was generated from the following file:

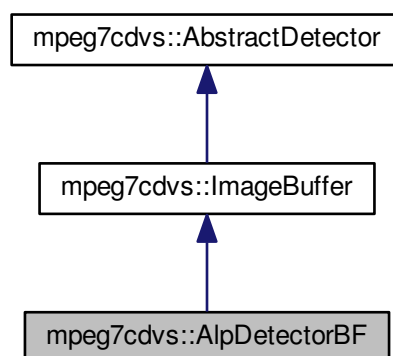
- [AlpDetector.h](#)

## 7.6 mpeg7cdvs::AlpDetectorBF Class Reference

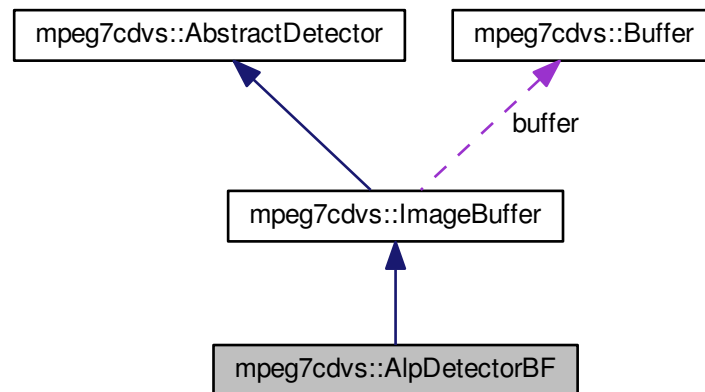
implementation of the ALP keypoint detector by employing block-based processing and frequency domain filtering.

```
#include <AlpDetectorBF.h>
```

Inheritance diagram for mpeg7cdvs::AlpDetectorBF:



Collaboration diagram for mpeg7cdvs::AlpDetectorBF:



## Public Member Functions

- [AlpDetectorBF](#) ()
- void [detect](#) ([FeatureList](#) &featurelist, const [Parameters](#) &params)  
*Detect all ALP keypoints from this image.*
- void [extract](#) ([FeatureList](#) &featurelist, size\_t num) const  
*Extract the SIFT descriptor of each keypoint and store it back in featurelist.*

## Additional Inherited Members

### 7.6.1 Detailed Description

implementation of the ALP keypoint detector by employing block-based processing and frequency domain filtering.

#### Author

Jie Chen

#### Date

2014

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 mpeg7cdvs::AlpDetectorBF::AlpDetectorBF ( )

### 7.6.3 Member Function Documentation

#### 7.6.3.1 void mpeg7cdvs::AlpDetectorBF::detect ( [FeatureList](#) & *featurelist*, const [Parameters](#) & *params* ) [virtual]

Detect all ALP keypoints from this image.

## Parameters

<i>featurelist</i>	the ouput list of keypoints with their associated features.
<i>params</i>	the running parameters.

## Exceptions

<a href="#"><i>CdvsException</i></a>	in case of error
--------------------------------------	------------------

Implements [mpeg7cdvs::AbstractDetector](#).

7.6.3.2 void mpeg7cdvs::AlpDetectorBF::extract ( FeatureList & *featurelist*, size\_t *num* ) const [virtual]

Extract the SIFT descriptor of each keypoint and store it back in featurelist.

## Parameters

<i>featurelist</i>	the detected keypoints
<i>num</i>	the absolute maximum number of features to be extracted from this image

Implements [mpeg7cdvs::AbstractDetector](#).

The documentation for this class was generated from the following file:

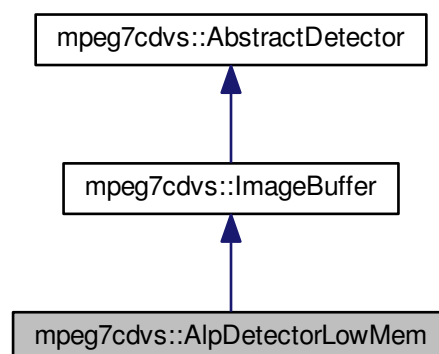
- [AlpDetectorBF.h](#)

## 7.7 mpeg7cdvs::AlpDetectorLowMem Class Reference

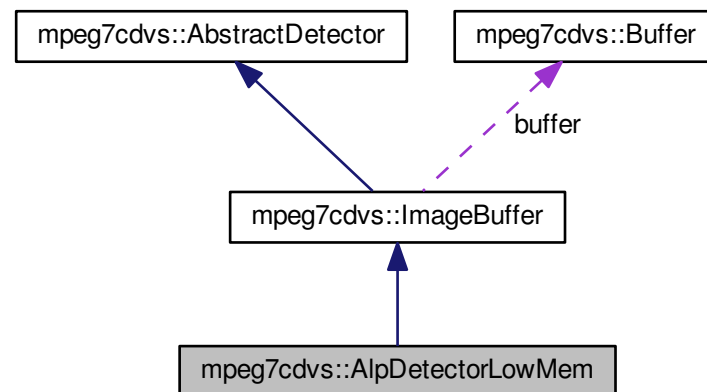
Low memory implementation of the ALP keypoint detector.

```
#include <AlpDetectorLowMem.h>
```

Inheritance diagram for mpeg7cdvs::AlpDetectorLowMem:



Collaboration diagram for mpeg7cdvs::AlpDetectorLowMem:



## Public Member Functions

- [AlpDetectorLowMem](#) ()
- void [detect](#) ([FeatureList](#) &featurelist, const [Parameters](#) &params)  
*Detect all ALP keypoints from this image.*
- void [extract](#) ([FeatureList](#) &featurelist, size\_t num) const  
*Extract the SIFT descriptor of each keypoint and store it back in featurelist.*

## Additional Inherited Members

### 7.7.1 Detailed Description

Low memory implementation of the ALP keypoint detector.

#### Author

Gianluca Francini, Massimo Balestri

#### Date

2013

### 7.7.2 Constructor & Destructor Documentation

#### 7.7.2.1 mpeg7cdvs::AlpDetectorLowMem::AlpDetectorLowMem ( )

### 7.7.3 Member Function Documentation

#### 7.7.3.1 void mpeg7cdvs::AlpDetectorLowMem::detect ( [FeatureList](#) & *featurelist*, const [Parameters](#) & *params* ) [virtual]

Detect all ALP keypoints from this image.

## Parameters

<i>featurelist</i>	the ouput list of keypoints with their associated features.
<i>params</i>	the running parameters.

## Exceptions

<a href="#"><i>CdvsException</i></a>	in case of error
--------------------------------------	------------------

Implements [mpeg7cdvs::AbstractDetector](#).

**7.7.3.2** void [mpeg7cdvs::AlpDetectorLowMem::extract](#) ( [FeatureList](#) & *featurelist*, [size\\_t](#) *num* ) const [virtual]

Extract the SIFT descriptor of each keypoint and store it back in featurelist.

## Parameters

<i>featurelist</i>	the detected keypoints
<i>num</i>	the absolute maximum number of features to be extracted from this image

Implements [mpeg7cdvs::AbstractDetector](#).

The documentation for this class was generated from the following file:

- [AlpDetectorLowMem.h](#)

## 7.8 mpeg7cdvs::AlpOctave Class Reference

A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.

```
#include <AlpOctave.h>
```

### Public Member Functions

- [AlpOctave](#) ()  
*constructor*
- virtual [~AlpOctave](#) ()  
*destructor*
- bool [init](#) (unsigned char \*data, int width, int height, int extra\_top\_lines=0, int extra\_bottom\_lines=0, bool fast=false)  
*initialize the octave using the given image.*
- bool [init](#) (const [AlpOctave](#) &previous)  
*initialize the octave using the previous octave.*
- bool [empty](#) () const  
*return true if the [AlpOctave](#) is empty*
- void [clear](#) ()  
*clear the [AlpOctave](#)*
- void [detect](#) ()  
*Detect all ALP keypoints from this octave.*
- void [detectDuplicates](#) ([AlpOctave](#) &previous)  
*Drop duplicate points between this and the previous octave.*
- void [detectDuplicates](#) ([FeatureList](#) &featurelist)  
*Low-memory version of detect duplicates.*
- void [getAlpKeypoints](#) (std::vector< [FeatureAlp](#) > &outKeypoints, bool rescale=true)  
*Get Alp keypoints.*



- void `getKeypoints` (std::vector< [Feature](#) > &outKeypoints, bool calcDescriptor=false, bool rescale=true, int base\_Y=0)  
*Get the final keypoints.*
- void `getKeypoints_fast_mode` (std::vector< [Feature](#) > &outKeypoints, float minpdf=0.0, bool calcDescriptor=false, bool rescale=true, int base\_Y=0)  
*Get the final keypoints (fast mode).*
- void `computeDescriptor` ([Feature](#) &keypoint, bool rescale) const  
*Compute the descriptor of the given keypoint.*
- void `computeDescriptor` ([FeatureList](#) &featurelist, const [FeatureAlp](#) &keypoint, bool rescale, size\_t missing) const  
*Compute orientation and descriptor of the given ALP keypoint and store it in featurelist.*
- void `computeDescriptor_fast_mode` ([Feature](#) &keypoint, bool rescale) const  
*This function was added by ETRI which is fast mode of computeDescriptor Partial gradient computation was adopted in this function.*
- bool `processFirst` (unsigned char \*data, int width, int height, [FeatureList](#) &featurelist)  
*Process the first octave of the given image (low memory version).*
- bool `processNext` ([FeatureList](#) &featurelist)  
*Process the next octave (low memory version).*
- int `getHeight` () const  
*Export the height value (read only)*
- int `getWidth` () const  
*Export the width value (read only)*
- int `getOctave` () const  
*Export the octave value (read only), the meaning of which is defined in the vl\_sift library: By convention, the octave of index 0 starts with the image full resolution.*

## Data Fields

- std::vector< [FeatureAlp](#) > `keypoints`  
*raw key points detected in this octave (without orientation) // Keundong Lee, moved to public*

### 7.8.1 Detailed Description

A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.

#### Author

Massimo Balestri

#### Date

2013

### 7.8.2 Constructor & Destructor Documentation

#### 7.8.2.1 mpeg7cdvs::AlpOctave::AlpOctave ( )

constructor

#### 7.8.2.2 virtual mpeg7cdvs::AlpOctave::~~AlpOctave ( ) [virtual]

destructor

### 7.8.3 Member Function Documentation

#### 7.8.3.1 void mpeg7cdvs::AlpOctave::clear ( )

clear the [AlpOctave](#)

#### 7.8.3.2 void mpeg7cdvs::AlpOctave::computeDescriptor ( Feature & *keypoint*, bool *rescale* ) const

Compute the descriptor of the given keypoint.

Parameters

<i>keypoint</i>	the input/output keypoint instance
<i>rescale</i>	if true, the keypoint x and y coordinates are scaled to match the octave of the keypoint

#### 7.8.3.3 void mpeg7cdvs::AlpOctave::computeDescriptor ( FeatureList & *featurelist*, const FeatureAlp & *keypoint*, bool *rescale*, size\_t *missing* ) const

Compute orientation and descriptor of the given ALP keypoint and store it in featurelist.

Each ALP keypoint can produce up to four output keypoints in featurelist, having different orientation.

Parameters

<i>featurelist</i>	the output list of keypoints
<i>keypoint</i>	the input ALP keypoint instance
<i>rescale</i>	if true, the keypoint x and y coordinates are scaled to match the octave of the keypoint
<i>missing</i>	how many missing points are currently required to be computed

#### 7.8.3.4 void mpeg7cdvs::AlpOctave::computeDescriptor\_fast\_mode ( Feature & *keypoint*, bool *rescale* ) const

This function was added by ETRI which is fast mode of computeDescriptor Partial gradient computation was adopted in this function.

#### 7.8.3.5 void mpeg7cdvs::AlpOctave::detect ( )

Detect all ALP keypoints from this octave.

Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

#### 7.8.3.6 void mpeg7cdvs::AlpOctave::detectDuplicates ( AlpOctave & *previous* )

Drop duplicate points between this and the previous octave.

This method assumes that the octave given as parameter is the previous octave.

Parameters

<i>previous</i>	the previous octave.
-----------------	----------------------

#### 7.8.3.7 void mpeg7cdvs::AlpOctave::detectDuplicates ( FeatureList & *featurelist* )

Low-memory version of detect duplicates.

## Parameters

<i>featurelist</i>	the list of keypoints already detected in the previous octaves.
--------------------	---

## 7.8.3.8 bool mpeg7cdvs::AlpOctave::empty ( ) const

return true if the [AlpOctave](#) is empty

7.8.3.9 void mpeg7cdvs::AlpOctave::getAlpKeypoints ( std::vector< [FeatureAlp](#) > & outKeypoints, bool rescale = true )

Get Alp keypoints.

These do not include orientation information and descriptor data. This must be done after [init\(\)](#), [detect\(\)](#) and [detectDuplicates\(\)](#) because it reuses the same memory buffers.

## Parameters

<i>outKeypoints</i>	the output vector of keypoints
<i>rescale</i>	rescale the keypoints to their absolute value

7.8.3.10 int mpeg7cdvs::AlpOctave::getHeight ( ) const [\[inline\]](#)

Export the height value (read only)

## Returns

the height of this octave

7.8.3.11 void mpeg7cdvs::AlpOctave::getKeypoints ( std::vector< [Feature](#) > & outKeypoints, bool calcDescriptor = false, bool rescale = true, int base\_Y = 0 )

Get the final keypoints.

This must be done after [init\(\)](#), [detect\(\)](#) and [detectDuplicates\(\)](#) because it reuses the same memory buffers.

## Parameters

<i>outKeypoints</i>	the output vector of keypoints
<i>calcDescriptor</i>	optionally compute the descriptor of each keypoint
<i>rescale</i>	rescale the keypoints to their absolute value
<i>base_Y</i>	displace all Y coordinates using the given base Y

7.8.3.12 void mpeg7cdvs::AlpOctave::getKeypoints\_fast\_mode ( std::vector< [Feature](#) > & outKeypoints, float minpdf = 0.0, bool calcDescriptor = false, bool rescale = true, int base\_Y = 0 )

Get the final keypoints (fast mode).

This function was added by ETRI. This function computes orientation only for keypoints having a pdf greater than minpdf. Partial gradient computation was adopted in this function.

## Parameters

<i>outKeypoints</i>	the output vector of keypoints
<i>minpdf</i>	is added for preliminary feature selection
<i>calcDescriptor</i>	optionally compute the descriptor of each keypoint
<i>rescale</i>	rescale the keypoints to their absolute value
<i>base_Y</i>	displace all Y coordinates using the given base Y

#### 7.8.3.13 `int mpeg7cdvs::AlpOctave::getOctave ( ) const [inline]`

Export the octave value (read only), the meaning of which is defined in the vl\_sift library: By convention, the octave of index 0 starts with the image full resolution.

Specifying an index greater than 0 starts the scale space at a lower resolution (e.g. 1 halves the resolution).

##### Returns

the octave number

#### 7.8.3.14 `int mpeg7cdvs::AlpOctave::getWidth ( ) const [inline]`

Export the width value (read only)

##### Returns

the width of this octave

#### 7.8.3.15 `bool mpeg7cdvs::AlpOctave::init ( unsigned char * data, int width, int height, int extra_top_lines = 0, int extra_bottom_lines = 0, bool fast = false )`

initialize the octave using the given image.

If the image size is above a given limit, it can be split horizontally into n parts to reduce memory allocation.

##### Parameters

<i>data</i>	the image data
<i>width</i>	the image width
<i>height</i>	the image height
<i>extra_top_lines</i>	extra top lines of the split image (images are split to reduce memory usage if needed)
<i>extra_bottom_lines</i>	extra bottom lines of the split image (images are split to reduce memory usage if needed)
<i>fast</i>	the boolean parameter indicating whether partial gradient computation will be used or not (Added by ETRI)

##### Returns

true if successful

#### 7.8.3.16 `bool mpeg7cdvs::AlpOctave::init ( const AlpOctave & previous )`

initialize the octave using the previous octave.

## Parameters

<i>previous</i>	the previous octave
-----------------	---------------------

## Returns

true if successful

### 7.8.3.17 bool mpeg7cdvs::AlpOctave::processFirst ( unsigned char \* *data*, int *width*, int *height*, FeatureList & *featurelist* )

Process the first octave of the given image (low memory version).

If the image size is above a given limit, it can be split horizontally into n parts (slices) to reduce memory allocation. Moreover, this implementation extract all keypoints before applying the feature selection. It must be followed by multiple [processNext\(\)](#) calls.

## Parameters

<i>data</i>	the image data
<i>width</i>	the image width
<i>height</i>	the image height
<i>featurelist</i>	the output list of features

## Returns

true if successful

### 7.8.3.18 bool mpeg7cdvs::AlpOctave::processNext ( FeatureList & *featurelist* )

Process the next octave (low memory version).

This implementation extract all keypoints before applying the feature selection.

## Parameters

<i>featurelist</i>	the output list of features
--------------------	-----------------------------

## Returns

true if successful, false if the next octave is too small.

## 7.8.4 Field Documentation

### 7.8.4.1 std::vector<FeatureAlp> mpeg7cdvs::AlpOctave::keypoints

raw key points detected in this octave (without orientation) // Keundong Lee, moved to public

The documentation for this class was generated from the following file:

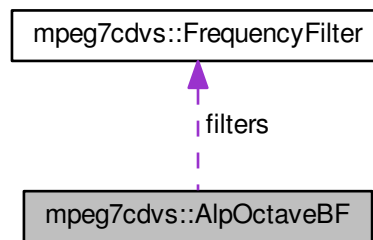
- [AlpOctave.h](#)

## 7.9 mpeg7cdvs::AlpOctaveBF Class Reference

A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.

```
#include <AlpOctaveBF.h>
```

Collaboration diagram for `mpeg7cdvs::AlpOctaveBF`:



## Public Member Functions

- `AlpOctaveBF ()`  
*constructor*
- `~AlpOctaveBF ()`  
*destructor*
- `bool empty () const`  
*return true if the AlpOctave is empty*
- `void detect ()`  
*Detect all ALP keypoints from this octave.*
- `void detectDuplicates (FeatureList &featurelist)`  
*Low-memory version of detect duplicates.*
- `int getHeight () const`  
*Export the height value (read only)*
- `int getWidth () const`  
*Export the width value (read only)*
- `int getOctave () const`  
*Export the octave value (read only), the meaning of which is defined in the vl\_sift library: By convention, the octave of index 0 starts with the image full resolution.*
- `void Allocate (int Image_width, int Image_height, int Block_width, int Block_height)`
- `void Clear ()`
- `void SpatialFiltering ()`
- `void FrequencyFiltering ()`
- `void getBlockInput (unsigned char *data=NULL)`
- `bool processOctave (FeatureList &featurelist, unsigned char *data=NULL, int width=0, int height=0)`
- `void computeDescriptor (FeatureList &featurelist, float xper)`

## Data Fields

- `std::vector< FeatureAlpBF > keypoints`  
*raw key points detected in this octave (without orientation) // Keundong Lee, moved to public*
- `int block_row_n`
- `int block_col_n`
- `int block_buffer_len`
- `int block_row_remain_width`

- int [block\\_col\\_remain\\_height](#)
- int [block\\_map](#) [64]
- [FrequencyFilter](#) \* [filters](#)
- int [block\\_width](#)
- int [block\\_height](#)
- int [extrema\\_w](#)
- int [extrema\\_h](#)
- int [extrema\\_sx](#)
- int [extrema\\_sy](#)
- int [padWidth](#)
- int [cur\\_block\\_id](#)
- int [h\\_mode](#)
- int [v\\_mode](#)
- int [windowSize](#)
- int [sx](#)
- int [sy](#)
- int [w2](#)
- int [h2](#)

### 7.9.1 Detailed Description

A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.

Author

Massimo Balestri

Date

2013

### 7.9.2 Constructor & Destructor Documentation

#### 7.9.2.1 mpeg7cdvs::AlpOctaveBF::AlpOctaveBF ( )

constructor

#### 7.9.2.2 mpeg7cdvs::AlpOctaveBF::~~AlpOctaveBF ( )

destructor

### 7.9.3 Member Function Documentation

#### 7.9.3.1 void mpeg7cdvs::AlpOctaveBF::Allocate ( int *Image\_width*, int *Image\_height*, int *Block\_width*, int *Block\_height* )

#### 7.9.3.2 void mpeg7cdvs::AlpOctaveBF::Clear ( )

#### 7.9.3.3 void mpeg7cdvs::AlpOctaveBF::computeDescriptor ( *FeatureList* & *featurelist*, float *xper* )

#### 7.9.3.4 void mpeg7cdvs::AlpOctaveBF::detect ( )

Detect all ALP keypoints from this octave.

## Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

7.9.3.5 void `mpeg7cdvs::AlpOctaveBF::detectDuplicates ( FeatureList & featurelist )`

Low-memory version of detect duplicates.

## Parameters

<i>featurelist</i>	the list of keypoints already detected in the previous octaves.
--------------------	---

7.9.3.6 bool `mpeg7cdvs::AlpOctaveBF::empty ( ) const`

return true if the [AlpOctave](#) is empty

7.9.3.7 void `mpeg7cdvs::AlpOctaveBF::FrequencyFiltering ( )`7.9.3.8 void `mpeg7cdvs::AlpOctaveBF::getBlockInput ( unsigned char * data = NULL )`7.9.3.9 int `mpeg7cdvs::AlpOctaveBF::getHeight ( ) const` `[inline]`

Export the height value (read only)

## Returns

the height of this octave

7.9.3.10 int `mpeg7cdvs::AlpOctaveBF::getOctave ( ) const` `[inline]`

Export the octave value (read only), the meaning of which is defined in the `vl_sift` library: By convention, the octave of index 0 starts with the image full resolution.

Specifying an index greater than 0 starts the scale space at a lower resolution (e.g. 1 halves the resolution).

## Returns

the octave number

7.9.3.11 int `mpeg7cdvs::AlpOctaveBF::getWidth ( ) const` `[inline]`

Export the width value (read only)

## Returns

the width of this octave

7.9.3.12 bool `mpeg7cdvs::AlpOctaveBF::processOctave ( FeatureList & featurelist, unsigned char * data = NULL, int width = 0, int height = 0 )`7.9.3.13 void `mpeg7cdvs::AlpOctaveBF::SpatialFiltering ( )`



## 7.9.4 Field Documentation

7.9.4.1 int mpeg7cdvs::AlpOctaveBF::block\_buffer\_len

7.9.4.2 int mpeg7cdvs::AlpOctaveBF::block\_col\_n

7.9.4.3 int mpeg7cdvs::AlpOctaveBF::block\_col\_remain\_height

7.9.4.4 int mpeg7cdvs::AlpOctaveBF::block\_height

7.9.4.5 int mpeg7cdvs::AlpOctaveBF::block\_map[64]

7.9.4.6 int mpeg7cdvs::AlpOctaveBF::block\_row\_n

7.9.4.7 int mpeg7cdvs::AlpOctaveBF::block\_row\_remain\_width

7.9.4.8 int mpeg7cdvs::AlpOctaveBF::block\_width

7.9.4.9 int mpeg7cdvs::AlpOctaveBF::cur\_block\_id

7.9.4.10 int mpeg7cdvs::AlpOctaveBF::extrema\_h

7.9.4.11 int mpeg7cdvs::AlpOctaveBF::extrema\_sx

7.9.4.12 int mpeg7cdvs::AlpOctaveBF::extrema\_sy

7.9.4.13 int mpeg7cdvs::AlpOctaveBF::extrema\_w

7.9.4.14 **FrequencyFilter\*** mpeg7cdvs::AlpOctaveBF::filters

7.9.4.15 int mpeg7cdvs::AlpOctaveBF::h2

7.9.4.16 int mpeg7cdvs::AlpOctaveBF::h\_mode

7.9.4.17 std::vector<**FeatureAlpBF**> mpeg7cdvs::AlpOctaveBF::keypoints

raw key points detected in this octave (without orientation) // Keundong Lee, moved to public

7.9.4.18 int mpeg7cdvs::AlpOctaveBF::padWidth

7.9.4.19 int mpeg7cdvs::AlpOctaveBF::sx

7.9.4.20 int mpeg7cdvs::AlpOctaveBF::sy

7.9.4.21 int mpeg7cdvs::AlpOctaveBF::v\_mode

7.9.4.22 int mpeg7cdvs::AlpOctaveBF::w2

7.9.4.23 int mpeg7cdvs::AlpOctaveBF::windowSize

The documentation for this class was generated from the following file:

- [AlpOctaveBF.h](#)

## 7.10 Ball2D Struct Reference

```
#include <Points.h>
```

### Public Member Functions

- [Ball2D](#) ([Point2D](#) const &[center](#)=[Point2D](#)(0.0, 0.0), float [radius](#)=1.0)

### Data Fields

- [Point2D](#) [center](#)
- float [radius](#)

### 7.10.1 Constructor & Destructor Documentation

7.10.1.1 [Ball2D::Ball2D](#) ( [Point2D](#) const & [center](#) = [Point2D](#) (0.0, 0.0), float [radius](#) = 1.0 ) [inline]

### 7.10.2 Field Documentation

7.10.2.1 [Point2D](#) [Ball2D::center](#)

7.10.2.2 float [Ball2D::radius](#)

The documentation for this struct was generated from the following file:

- [Points.h](#)

## 7.11 mpeg7cdvs::BitInputStream Class Reference

This class represents an input stream of bits.

```
#include <BitInputStream.h>
```

### Public Member Functions

- [BitInputStream](#) ()  
*Create an empty object.*
- [BitInputStream](#) (const unsigned char \*buf, size\_t size)  
*Create and initialize a [BitInputStream](#).*
- virtual [~BitInputStream](#) ()  
*Close the stream, if not yet done, and destroy the object.*
- void [open](#) (const unsigned char \*buf, size\_t size)  
*Attach the buffer from which data will be read in all subsequent operations.*
- void [close](#) ()  
*closes this input stream and releases any resources associated with the stream*
- unsigned int [read](#) ()  
*Reads the next bit from the input stream. The operation fails if [eof\(\)](#) is true.*
- unsigned int [read](#) (unsigned int nbits)  
*Reads the specified number of bits from the input stream.*
- void [read](#) (unsigned char \*destination, unsigned int nbits)

*Reads the specified number of bits from the input stream into the destination buffer, assuming that the input is byte-aligned.*

- void [skip](#) (unsigned int nbits)

*Skip the next n bits while reading from the current position.*

- bool [eof](#) () const

*Informs about the read cursor position.*

- void [reset](#) ()

*Reposition the read pointer at the beginning of the stream.*

- unsigned int [align](#) ()

*Align the read pointer to the closest byte boundary.*

- size\_t [available](#) () const

*Returns the number of bits that can be read from this stream starting from the current position.*

- size\_t [consumed](#) () const

*Returns the number of bits that have been read so far.*

- const unsigned char \* [getPointer](#) () const

*Return the current read pointer, assuming it is byte-aligned.*

- size\_t [getSize](#) () const

*Get the size of the attached buffer.*

- void [jumpTo](#) (size\_t position)

*Jump to the indicated absolute position (in bits).*

### 7.11.1 Detailed Description

This class represents an input stream of bits.

#### Author

Massimo Balestri, Andrea Varesio, Marco Vecchietti

#### Date

2002

### 7.11.2 Constructor & Destructor Documentation

#### 7.11.2.1 mpeg7cdvs::BitInputStream::BitInputStream ( )

Create an empty object.

The open method must be first called to actually use the object.

#### 7.11.2.2 mpeg7cdvs::BitInputStream::BitInputStream ( const unsigned char \* buf, size\_t size )

Create and initialize a [BitInputStream](#).

Attach the buffer from which data will be read in all subsequent operations.

#### Parameters

<i>buf</i>	the buffer from which the data will be read
------------	---

<i>size</i>	the size of the buffer in bytes (minimum size is 4 bytes)
-------------	---

7.11.2.3 `virtual mpeg7cdvs::BitInputStream::~~BitInputStream ( ) [virtual]`

Close the stream, if not yet done, and destroy the object.

### 7.11.3 Member Function Documentation

7.11.3.1 `unsigned int mpeg7cdvs::BitInputStream::align ( )`

Align the read pointer to the closest byte boundary.

If the read pointer is already aligned, the read pointer is not changed.

#### Returns

the number of skipped bits

7.11.3.2 `size_t mpeg7cdvs::BitInputStream::available ( ) const`

Returns the number of bits that can be read from this stream starting from the current position.

#### Returns

the number of bits that can be read.

7.11.3.3 `void mpeg7cdvs::BitInputStream::close ( )`

closes this input stream and releases any resources associated with the stream

7.11.3.4 `size_t mpeg7cdvs::BitInputStream::consumed ( ) const`

Returns the number of bits that have been read so far.

#### Returns

the number of bits read so far.

7.11.3.5 `bool mpeg7cdvs::BitInputStream::eof ( ) const`

Informs about the read cursor position.

#### Returns

true if the end of the input buffer has been reached.

7.11.3.6 `const unsigned char* mpeg7cdvs::BitInputStream::getPointer ( ) const`

Return the current read pointer, assuming it is byte-aligned.

#### Returns

the read pointer

7.11.3.7 `size_t mpeg7cdvs::BitInputStream::getSize ( ) const`

Get the size of the attached buffer.

## Returns

the size in bytes.

7.11.3.8 `void mpeg7cdvs::BitInputStream::jumpTo ( size_t position )`

Jump to the indicated absolute position (in bits).

## Parameters

<i>position</i>	the number of bits to skip from the start of the buffer.
-----------------	--

7.11.3.9 `void mpeg7cdvs::BitInputStream::open ( const unsigned char * buf, size_t size )`

Attach the buffer from which data will be read in all subsequent operations.

## Parameters

<i>buf</i>	the buffer from which the data will be read
<i>size</i>	the size of the buffer in bytes (minimum size is 4 bytes)

7.11.3.10 `unsigned int mpeg7cdvs::BitInputStream::read ( )`

Reads the next bit from the input stream. The operation fails if `eof()` is true.

## Returns

the next bit from the input stream.

7.11.3.11 `unsigned int mpeg7cdvs::BitInputStream::read ( unsigned int nbits )`

Reads the specified number of bits from the input stream.

The operation fails if `eof()` is true.

## Parameters

<i>nbits</i>	the number of bits to read (in the range 1..32)
--------------	---

## Returns

the next n bits from the input stream.

7.11.3.12 `void mpeg7cdvs::BitInputStream::read ( unsigned char * destination, unsigned int nbits )`

Reads the specified number of bits from the input stream into the destination buffer, assuming that the input is byte-aligned.

The operation fails if `eof()` is true.

## Parameters

<i>destination</i>	the destination buffer
<i>nbits</i>	the number of bits to read (8*n, assuming n>0)

## 7.11.3.13 void mpeg7cdvs::BitInputStream::reset ( )

Reposition the read pointer at the beginning of the stream.

7.11.3.14 void mpeg7cdvs::BitInputStream::skip ( unsigned int *nbits* )

Skip the next n bits while reading from the current position.

If the end of the buffer is reached or even surpassed, [eof\(\)](#) will return true.

## Parameters

<i>nbits</i>	the number of bits to skip (0..MAXINT)
--------------	--

The documentation for this class was generated from the following file:

- [BitInputStream.h](#)

## 7.12 mpeg7cdvs::BitOutputStream Class Reference

This class represents an output stream of bits.

```
#include <BitOutputStream.h>
```

### Public Member Functions

- [BitOutputStream](#) ()  
*Creates an empty object.*
- [BitOutputStream](#) (unsigned char \*buf, size\_t size)  
*Create and initialize a [BitOutputStream](#).*
- virtual [~BitOutputStream](#) ()  
*Closes the stream, if not yet done, and destroys the object.*
- void [open](#) (unsigned char \*buf, size\_t size)  
*Attaches the object to a buffer of known size.*
- size\_t [close](#) ()  
*Close this input stream.*
- void [flush](#) (unsigned int fill)  
*Align to the next byte boundary and flushes this output stream forcing any buffered output bits to be written in the destination buffer.*
- void [write](#) (unsigned int bit)  
*writes one bit into the output stream.*
- void [write](#) (unsigned int value, unsigned int nbits)  
*Writes the specified number of bits into the input stream.*
- void [write](#) (unsigned char \*source, unsigned int nbits)  
*Writes the specified number of bits from the source buffer into the output stream, assuming that the output is byte-aligned.*
- void [reset](#) ()  
*Reposition the write pointer at the beginning of the stream.*

- void [skip](#) (unsigned int nbits)  
*Skip the next n bits while writing into the current position.*
- bool [eof](#) () const  
*Informs about the write cursor position.*
- void [align](#) (unsigned int fill)  
*Align the write pointer to the closest byte boundary.*
- size\_t [available](#) () const  
*Returns the number of bits that can be written into this stream starting from the current position.*
- size\_t [produced](#) () const  
*Returns the number of bits that have been written so far.*
- unsigned char \* [getPointer](#) () const  
*returns the current write pointer, assuming it is byte-aligned.*
- size\_t [getSize](#) () const  
*Get the size of the attached buffer.*
- void [jumpTo](#) (size\_t position)  
*Jump to the indicated absolute position (in bits).*

### 7.12.1 Detailed Description

This class represents an output stream of bits.

#### Author

Massimo Balestri, Andrea Varesio, Marco Vecchietti

#### Date

2002

### 7.12.2 Constructor & Destructor Documentation

#### 7.12.2.1 mpeg7cdvs::BitOutputStream::BitOutputStream ( )

Creates an empty object.

The open method must be first called to actually use the object.

#### 7.12.2.2 mpeg7cdvs::BitOutputStream::BitOutputStream ( unsigned char \* buf, size\_t size )

Create and initialize a [BitOutputStream](#).

#### 7.12.2.3 virtual mpeg7cdvs::BitOutputStream::~~BitOutputStream ( ) [virtual]

Closes the stream, if not yet done, and destroys the object.

### 7.12.3 Member Function Documentation

#### 7.12.3.1 void mpeg7cdvs::BitOutputStream::align ( unsigned int fill )

Align the write pointer to the closest byte boundary.

If the write pointer is already aligned, the write pointer is not changed.

**Parameters**

<i>fill</i>	the value to be used in order to fill the missing bits (must be 0 or 1).
-------------	--

**7.12.3.2 `size_t mpeg7cdvs::BitOutputStream::available ( ) const`**

Returns the number of bits that can be written into this stream starting from the current position.

**Returns**

the number of bits that can be written.

**7.12.3.3 `size_t mpeg7cdvs::BitOutputStream::close ( )`**

Close this input stream.

This method flushes data and releases any resources associated with the stream.

**Returns**

the total number of produced bits.

**7.12.3.4 `bool mpeg7cdvs::BitOutputStream::eof ( ) const`**

Informs about the write cursor position.

**Returns**

true if the end of the output buffer has been reached.

**7.12.3.5 `void mpeg7cdvs::BitOutputStream::flush ( unsigned int fill )`**

Align to the next byte boundary and flushes this output stream forcing any buffered output bits to be written in the destination buffer.

**Parameters**

<i>fill</i>	the value to be used in order to fill the missing bits (must be 0 or 1).
-------------	--

**7.12.3.6 `unsigned char* mpeg7cdvs::BitOutputStream::getPointer ( ) const`**

returns the current write pointer, assuming it is byte-aligned.

**7.12.3.7 `size_t mpeg7cdvs::BitOutputStream::getSize ( ) const`**

Get the size of the attached buffer.

**Returns**

the size in bytes.

**7.12.3.8 `void mpeg7cdvs::BitOutputStream::jumpTo ( size_t position )`**

Jump to the indicated absolute position (in bits).

The absolute position must be byte-aligned.



## Parameters

<i>position</i>	the number of bits to skip from the start of the buffer.
-----------------	--

7.12.3.9 void mpeg7cdvs::BitOutputStream::open ( unsigned char \* *buf*, size\_t *size* )

Attaches the object to a buffer of known size.

This will be the destination in which data will be written in all subsequent operations.

## Parameters

<i>buf</i>	the buffer in which the data will be written
<i>size</i>	the size of the buffer in bytes

7.12.3.10 size\_t mpeg7cdvs::BitOutputStream::produced ( ) const

Returns the number of bits that have been written so far.

## Returns

the number of bits written so far.

7.12.3.11 void mpeg7cdvs::BitOutputStream::reset ( )

Reposition the write pointer at the beginning of the stream.

7.12.3.12 void mpeg7cdvs::BitOutputStream::skip ( unsigned int *nbits* )

Skip the next *n* bits while writing into the current position.

This operation first flushes any buffered bits, then jumps to the new location. If the end of the buffer is reached or even surpassed, [eof\(\)](#) will return true.

## Parameters

<i>nbits</i>	the number of bits to skip (0..MAXINT)
--------------	--

7.12.3.13 void mpeg7cdvs::BitOutputStream::write ( unsigned int *bit* )

writes one bit into the output stream.

The operation fails if [eof\(\)](#) is true.

## Parameters

<i>bit</i>	the bit to write (must be 0 or 1)
------------	-----------------------------------

7.12.3.14 void mpeg7cdvs::BitOutputStream::write ( unsigned int *value*, unsigned int *nbits* )

Writes the specified number of bits into the input stream.

The operation fails if [eof\(\)](#) is true.

## Parameters

<i>value</i>	the value to be written into stream.
<i>nbits</i>	the number of bits to write (in the range 1..32)

7.12.3.15 void mpeg7cdvs::BitOutputStream::write ( unsigned char \* *source*, unsigned int *nbits* )

Writes the specified number of bits from the source buffer into the output stream, assuming that the output is byte-aligned.

The operation fails if [eof\(\)](#) is true.

## Parameters

<i>source</i>	the source buffer (MUST be unsigned char, do not cast int or short arrays!)
<i>nbits</i>	the number of bits to be copied from the source into this output stream (8*n, assuming n>0)

The documentation for this class was generated from the following file:

- [BitOutputStream.h](#)

## 7.13 BoundingBox Class Reference

A class containing four points which identify the object in the image.

```
#include <BoundingBox.h>
```

### Public Member Functions

- [BoundingBox](#) ()
- void [read](#) (const char \*fn)  
*Read the four points from a bbox file.*
- [mpeg7cdvs::CDVSPPOINT](#) \* [getAddress](#) ()  
*Get the bounding box base address.*
- double [find\\_overlap](#) ([BoundingBox](#) &other)  
*Compute ratio of intersection to union of two quadrilateral regions in an image.*

### Static Public Member Functions

- static double [cfind\\_overlap](#) (int width, int height, [mpeg7cdvs::CDVSPPOINT](#) \*quad1, [mpeg7cdvs::CDVSPPOINT](#) \*quad2)  
*Compute ratio of intersection to union of two quadrilateral regions in an image (backward compatible version).*

#### 7.13.1 Detailed Description

A class containing four points which identify the object in the image.

#### 7.13.2 Constructor & Destructor Documentation

##### 7.13.2.1 BoundingBox::BoundingBox ( )

#### 7.13.3 Member Function Documentation

7.13.3.1 `static double BoundingBox::cfind_overlap ( int width, int height, mpeg7cdvs::CDVSPPOINT * quad1, mpeg7cdvs::CDVSPPOINT * quad2 ) [static]`

Compute ratio of intersection to union of two quadrilateral regions in an image (backward compatible version).

This is the first version of find\_overlap, which limits the area of union and intersection to the query image area.

Parameters

<i>width</i>	width of the image
<i>height</i>	height of the image; height < 0 implies bottom-up ordering of rows (as in .bmp files)
<i>quad1</i>	vertices of first quadrilateral defined as points in image space (can be outside boundaries)
<i>quad2</i>	vertices of second quadrilateral defined as points in image space (can be outside boundaries)

Returns

area of intersection / area of union (~Jaccard Index)

7.13.3.2 `double BoundingBox::find_overlap ( BoundingBox & other )`

Compute ratio of intersection to union of two quadrilateral regions in an image.

This is the new version which computes the Jaccard index correctly.

Parameters

<i>other</i>	the other bounding box
--------------	------------------------

Returns

area of intersection / area of union (~Jaccard Index)

7.13.3.3 `mpeg7cdvs::CDVSPPOINT* BoundingBox::getAddress ( )`

Get the bounding box base address.

Returns

the address of the bbox vector

7.13.3.4 `void BoundingBox::read ( const char * fn )`

Read the four points from a bbox file.

The documentation for this class was generated from the following file:

- [BoundingBox.h](#)

## 7.14 mpeg7cdvs::Buffer Class Reference

A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.

```
#include <Buffer.h>
```

## Public Member Functions

- `Buffer ()`
- `virtual ~Buffer ()`
- `Buffer (size_t size)`  
*create a buffer of the given size*
- `Buffer (unsigned char *data, size_t size)`  
*copy the given array into this Buffer*
- `Buffer (const Buffer &)`  
*copy the given Buffer into this Buffer*
- `Buffer & operator= (const Buffer &)`  
*assign a Buffer to another*
- `void swap (Buffer &x)`  
*swap the content of two Buffer(s)*
- `void fill (unsigned char value=0)`  
*fill a Buffer with the given value*
- `size_t size () const`  
*return the current size of the Buffer*
- `bool resize (size_t newsiz)`  
*change buffer size; content is lost if newsiz if less than the current size*
- `bool empty () const`  
*return true if the Buffer is empty*
- `void clear ()`  
*clear the Buffer*
- `bool assign (const unsigned char *data, size_t size)`  
*assign the given data to Buffer*
- `bool equals (Buffer &buffer)`  
*compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)*
- `unsigned char * data ()`  
*access to Buffer's data (writable)*
- `const unsigned char * data () const`  
*access to Buffer's data (read only)*
- `void read (const char *fname)`  
*read Buffer from a file*
- `void write (const char *fname) const`  
*write Buffer to file*
- `int compare (const Buffer &other) const`  
*Compare this buffer with another; return the number of different bytes.*
- `bool operator== (const Buffer &other) const`  
*compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)*

### 7.14.1 Detailed Description

A container class for a byte array, intended to replace all `malloc()` and `new()` instructions in the main code.

This class properly deallocates memory when an exception is thrown.

#### Author

Massimo Balestri

#### Date

2013

## 7.14.2 Constructor & Destructor Documentation

7.14.2.1 `mpeg7cdvs::Buffer::Buffer ( )`

7.14.2.2 `virtual mpeg7cdvs::Buffer::~~Buffer ( )` `[virtual]`

7.14.2.3 `mpeg7cdvs::Buffer::Buffer ( size_t size )`

create a buffer of the given size

7.14.2.4 `mpeg7cdvs::Buffer::Buffer ( unsigned char * data, size_t size )`

copy the given array into this [Buffer](#)

7.14.2.5 `mpeg7cdvs::Buffer::Buffer ( const Buffer & )`

copy the given [Buffer](#) into this [Buffer](#)

## 7.14.3 Member Function Documentation

7.14.3.1 `bool mpeg7cdvs::Buffer::assign ( const unsigned char * data, size_t size )`

assign the given data to [Buffer](#)

7.14.3.2 `void mpeg7cdvs::Buffer::clear ( )`

clear the [Buffer](#)

7.14.3.3 `int mpeg7cdvs::Buffer::compare ( const Buffer & other ) const`

Compare this buffer with another; return the number of different bytes.

Parameters

<i>other</i>	the other <a href="#">Buffer</a>
--------------	----------------------------------

Returns

the number of differences; zero if no difference is found.

7.14.3.4 `unsigned char* mpeg7cdvs::Buffer::data ( )`

access to [Buffer](#)'s data (writable)

7.14.3.5 `const unsigned char* mpeg7cdvs::Buffer::data ( ) const`

access to [Buffer](#)'s data (read only)

7.14.3.6 `bool mpeg7cdvs::Buffer::empty ( ) const`

return true if the [Buffer](#) is empty

7.14.3.7 `bool mpeg7cdvs::Buffer::equals ( Buffer & buffer )`

compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)

7.14.3.8 `void mpeg7cdvs::Buffer::fill ( unsigned char value = 0 )`

fill a Buffer with the given value

7.14.3.9 `Buffer& mpeg7cdvs::Buffer::operator= ( const Buffer & )`

assign a Buffer to another

7.14.3.10 `bool mpeg7cdvs::Buffer::operator== ( const Buffer & other ) const`

compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)

7.14.3.11 `void mpeg7cdvs::Buffer::read ( const char * fname )`

read Buffer from a file

7.14.3.12 `bool mpeg7cdvs::Buffer::resize ( size_t newsize )`

change buffer size; content is lost if newsize if less than the current size

7.14.3.13 `size_t mpeg7cdvs::Buffer::size ( ) const`

return the current size of the Buffer

7.14.3.14 `void mpeg7cdvs::Buffer::swap ( Buffer & x )`

swap the content of two Buffer(s)

7.14.3.15 `void mpeg7cdvs::Buffer::write ( const char * fname ) const`

write Buffer to file

The documentation for this class was generated from the following file:

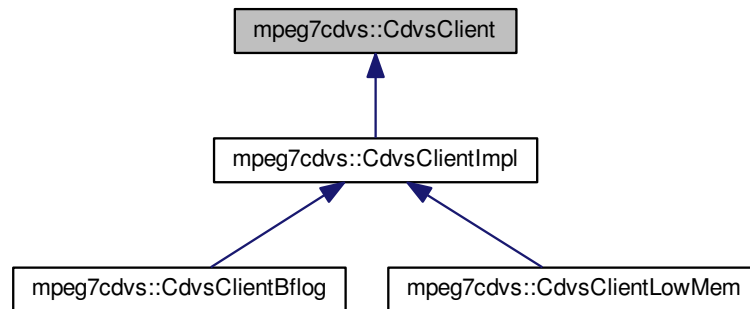
- [Buffer.h](#)

## 7.15 mpeg7cdvs::CdvsClient Class Reference

Interface to the client-side functionality of the CDVS Library.

```
#include <CdvsInterface.h>
```

Inheritance diagram for mpeg7cdvs::CdvsClient:



## Public Member Functions

- virtual [~CdvsClient](#) ()
- virtual unsigned int [encode](#) ([CdvsDescriptor](#) &output, int width, int height, const unsigned char \*input) const =0

*Encode the luminance component of an image producing a CDVS descriptor.*

## Static Public Member Functions

- static [CdvsClient](#) \* [cdvsClientFactory](#) (const [CdvsConfiguration](#) \*config, int mode)
- Create an instance of a CDVS Client producing descriptors according to the indicated mode.*

### 7.15.1 Detailed Description

Interface to the client-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.15.2 Constructor & Destructor Documentation

7.15.2.1 virtual mpeg7cdvs::CdvsClient::~~CdvsClient ( ) [inline], [virtual]

### 7.15.3 Member Function Documentation

7.15.3.1 static CdvsClient\* mpeg7cdvs::CdvsClient::cdvsClientFactory ( const CdvsConfiguration \* config, int mode ) [static]

Create an instance of a CDVS Client producing descriptors according to the indicated mode.

The calling entity takes ownership of the instance (i.e. must delete the instance when not used anymore).

## Parameters

<i>config</i>	the parameter configuration that will be used to produce descriptors.
<i>mode</i>	mode of the descriptors produced by the client instance.

## Returns

a pointer to the Cdvs Client instance

**7.15.3.2** `virtual unsigned int mpeg7cdvs::CdvsClient::encode ( CdvsDescriptor & output, int width, int height, const unsigned char * input ) const [pure virtual]`

Encode the luminance component of an image producing a CDVS descriptor.

## Parameters

<i>output</i>	the output CDVS descriptor
<i>width</i>	width of the image
<i>height</i>	height of the image
<i>input</i>	the buffer containing the luminance component of the image (Y component, 8 bit per pixel)

## Returns

the actual size of the encoded CDVS descriptor

Implemented in [mpeg7cdvs::CdvsClientImpl](#), [mpeg7cdvs::CdvsClientBflog](#), and [mpeg7cdvs::CdvsClientLowMem](#).

The documentation for this class was generated from the following file:

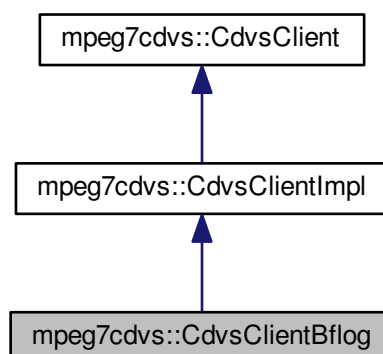
- [CdvsInterface.h](#)

## 7.16 mpeg7cdvs::CdvsClientBflog Class Reference

Block-based frequency domain Laplacian of Gaussian implementation of the high level interface to the client-side functionality of the CDVS Library.

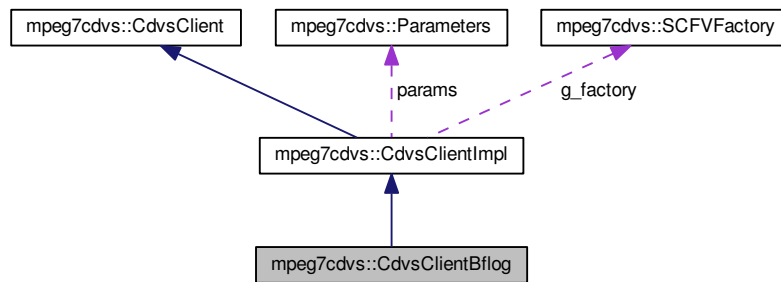
```
#include <CdvsClientBflog.h>
```

Inheritance diagram for mpeg7cdvs::CdvsClientBflog:





Collaboration diagram for mpeg7cdvs::CdvsClientBflog:



## Public Member Functions

- [CdvsClientBflog](#) (const [CdvsConfiguration](#) \*config, int mode)
- virtual [~CdvsClientBflog](#) ()
- virtual unsigned int [encode](#) ([CdvsDescriptor](#) &output, int width, int height, const unsigned char \*input) const  
*Encode the luminance component of an image producing a CDVS descriptor.*

## Additional Inherited Members

### 7.16.1 Detailed Description

Block-based frequency domain Laplacian of Gaussian implementation of the high level interface to the client-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.16.2 Constructor & Destructor Documentation

7.16.2.1 `mpeg7cdvs::CdvsClientBflog::CdvsClientBflog ( const CdvsConfiguration * config, int mode )`

7.16.2.2 `virtual mpeg7cdvs::CdvsClientBflog::~~CdvsClientBflog ( ) [virtual]`

### 7.16.3 Member Function Documentation

7.16.3.1 `virtual unsigned int mpeg7cdvs::CdvsClientBflog::encode ( CdvsDescriptor & output, int width, int height, const unsigned char * input ) const [virtual]`

Encode the luminance component of an image producing a CDVS descriptor.

## Parameters

<i>output</i>	the output CDVS descriptor
<i>width</i>	width of the image
<i>height</i>	height of the image
<i>input</i>	the buffer containing the luminance component of the image (Y component, 8 bit per pixel)

## Returns

the actual size of the encoded CDVS descriptor

Reimplemented from [mpeg7cdvs::CdvsClientImpl](#).

The documentation for this class was generated from the following file:

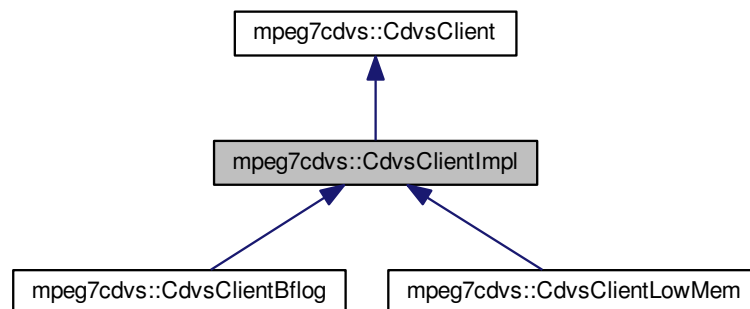
- [CdvsClientBflog.h](#)

## 7.17 mpeg7cdvs::CdvsClientImpl Class Reference

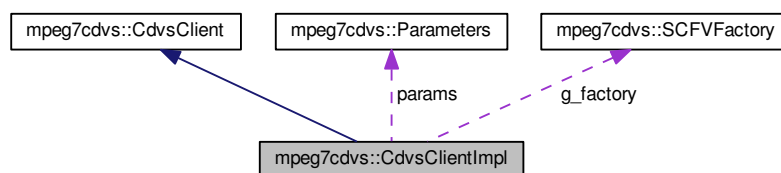
Main implementation of the high level interface to the client-side functionality of the CDVS Library.

```
#include <CdvsClientImpl.h>
```

Inheritance diagram for mpeg7cdvs::CdvsClientImpl:



Collaboration diagram for mpeg7cdvs::CdvsClientImpl:



## Public Member Functions

- [CdvsClientImpl](#) (const [CdvsConfiguration](#) \*config, int mode)
- virtual [~CdvsClientImpl](#) ()
- virtual unsigned int [encode](#) ([CdvsDescriptor](#) &output, int width, int height, const unsigned char \*buffer) const  
*Encode the luminance component of an image producing a CDVS descriptor.*

## Protected Attributes

- [SCFVFactory](#) g\_factory  
*global descriptor factory for the query*
- [Parameters](#) params  
*parameters used by this client*
- int [modeId](#)  
*the mode ID used by this client*

## Additional Inherited Members

### 7.17.1 Detailed Description

Main implementation of the high level interface to the client-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.17.2 Constructor & Destructor Documentation

7.17.2.1 `mpeg7cdvs::CdvsClientImpl::CdvsClientImpl ( const CdvsConfiguration * config, int mode )`

7.17.2.2 `virtual mpeg7cdvs::CdvsClientImpl::~~CdvsClientImpl ( )` `[virtual]`

### 7.17.3 Member Function Documentation

7.17.3.1 `virtual unsigned int mpeg7cdvs::CdvsClientImpl::encode ( CdvsDescriptor & output, int width, int height, const unsigned char * input ) const` `[virtual]`

Encode the luminance component of an image producing a CDVS descriptor.

#### Parameters

<i>output</i>	the output CDVS descriptor
<i>width</i>	width of the image
<i>height</i>	height of the image
<i>input</i>	the buffer containing the luminance component of the image (Y component, 8 bit per pixel)

#### Returns

the actual size of the encoded CDVS descriptor

Implements [mpeg7cdvs::CdvsClient](#).

Reimplemented in [mpeg7cdvs::CdvsClientBflog](#), and [mpeg7cdvs::CdvsClientLowMem](#).

### 7.17.4 Field Documentation

#### 7.17.4.1 SCFVFactory mpeg7cdvs::CdvsClientImpl::g\_factory [protected]

global descriptor factory for the query

#### 7.17.4.2 int mpeg7cdvs::CdvsClientImpl::modelId [protected]

the mode ID used by this client

#### 7.17.4.3 Parameters mpeg7cdvs::CdvsClientImpl::params [protected]

parameters used by this client

The documentation for this class was generated from the following file:

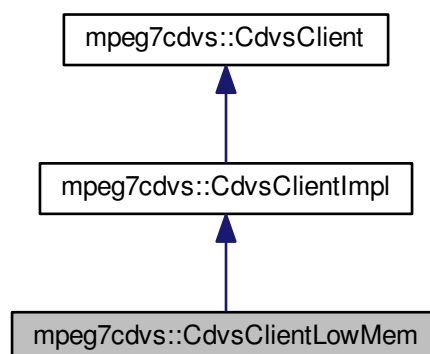
- [CdvsClientImpl.h](#)

## 7.18 mpeg7cdvs::CdvsClientLowMem Class Reference

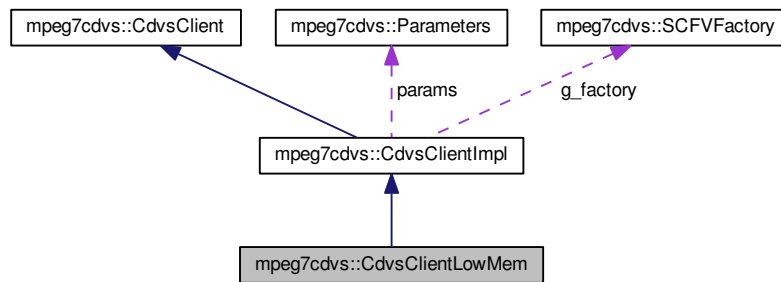
Low Memory implementation of the high level interface to the client-side functionality of the CDVS Library.

```
#include <CdvsClientLowMem.h>
```

Inheritance diagram for mpeg7cdvs::CdvsClientLowMem:



Collaboration diagram for mpeg7cdvs::CdvsClientLowMem:



## Public Member Functions

- [CdvsClientLowMem](#) (const [CdvsConfiguration](#) \*config, int mode)
- virtual [~CdvsClientLowMem](#) ()
- virtual unsigned int [encode](#) ([CdvsDescriptor](#) &output, int width, int height, const unsigned char \*input) const  
*Encode the luminance component of an image producing a CDVS descriptor.*

## Additional Inherited Members

### 7.18.1 Detailed Description

Low Memory implementation of the high level interface to the client-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.18.2 Constructor & Destructor Documentation

7.18.2.1 `mpeg7cdvs::CdvsClientLowMem::CdvsClientLowMem ( const CdvsConfiguration * config, int mode )`

7.18.2.2 `virtual mpeg7cdvs::CdvsClientLowMem::~~CdvsClientLowMem ( )` `[virtual]`

### 7.18.3 Member Function Documentation

7.18.3.1 `virtual unsigned int mpeg7cdvs::CdvsClientLowMem::encode ( CdvsDescriptor & output, int width, int height, const unsigned char * input ) const` `[virtual]`

Encode the luminance component of an image producing a CDVS descriptor.

## Parameters

<i>output</i>	the output CDVS descriptor
<i>width</i>	width of the image
<i>height</i>	height of the image
<i>input</i>	the buffer containing the luminance component of the image (Y component, 8 bit per pixel)

## Returns

the actual size of the encoded CDVS descriptor

Reimplemented from [mpeg7cdvs::CdvsClientImpl](#).

The documentation for this class was generated from the following file:

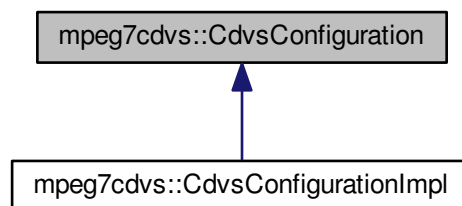
- [CdvsClientLowMem.h](#)

## 7.19 mpeg7cdvs::CdvsConfiguration Class Reference

Interface to all configuration parameters for clients and servers.

```
#include <CdvsInterface.h>
```

Inheritance diagram for mpeg7cdvs::CdvsConfiguration:



### Public Member Functions

- virtual [~CdvsConfiguration](#) ()
- virtual const [Parameters](#) & [getParameters](#) (int mode) const =0  
Get one of the [Parameters](#) instances (note that this class keeps an instance of all parameters for all modes).
- virtual [Parameters](#) & [setParameters](#) (int mode)=0  
Set some [Parameters](#) value for a specific mode.

### Static Public Member Functions

- static [CdvsConfiguration](#) \* [cdvsConfigurationFactory](#) (const char \*configfile=NULL)  
Create an instance of a CDVS configuration containing all default coding/decoding parameters.
- static int [getMode](#) (int descLen)  
Get the mode ID corresponding to a specific descriptor length.

### 7.19.1 Detailed Description

Interface to all configuration parameters for clients and servers.

Author

Massimo Balestri

Date

2014

### 7.19.2 Constructor & Destructor Documentation

7.19.2.1 `virtual mpeg7cdvs::CdvsConfiguration::~CdvsConfiguration ( ) [inline],[virtual]`

### 7.19.3 Member Function Documentation

7.19.3.1 `static CdvsConfiguration* mpeg7cdvs::CdvsConfiguration::cdvsConfigurationFactory ( const char * configfile = NULL ) [static]`

Create an instance of a CDVS configuration containing all default coding/decoding parameters.

The configuration instance can be used to initialize a client or a server CDVS instance. The configuration can be modified using the `setParameters` method. The calling entity takes ownership of the instance (i.e. must delete the instance when not used anymore).

Parameters

<i>configfile</i>	a file containing some or all parameters replacing the default values.
-------------------	--

Returns

a [CdvsConfiguration](#) instance

7.19.3.2 `static int mpeg7cdvs::CdvsConfiguration::getMode ( int descLen ) [static]`

Get the mode ID corresponding to a specific descriptor length.

The relation between length and mode ID is provided according to the MPEG CDVS specification:

- mode 1: 512 bytes
- mode 2: 1024 bytes
- mode 3: 2048 bytes
- mode 4: 4096 bytes
- mode 5: 8192 bytes
- mode 6: 16384 bytes

Parameters

<i>descLen</i>	the descriptor length (in bytes)
----------------	----------------------------------

Returns

the corresponding mode

7.19.3.3 `virtual const Parameters& mpeg7cdvs::CdvsConfiguration::getParameters ( int mode ) const` [pure virtual]

Get one of the [Parameters](#) instances (note that this class keeps an instance of all parameters for all modes).



## Parameters

<i>mode</i>	the mode for which parameters are requested.
-------------	--

## Returns

a read-only instance of the parameters.

Implemented in [mpeg7cdvs::CdvsConfigurationImpl](#).

#### 7.19.3.4 virtual Parameters& mpeg7cdvs::CdvsConfiguration::setParameters ( int *mode* ) [pure virtual]

Set some [Parameters](#) value for a specific mode.

## Parameters

<i>mode</i>	the mode for which parameters are requested.
-------------	--

## Returns

a modifiable instance of the parameters.

Implemented in [mpeg7cdvs::CdvsConfigurationImpl](#).

The documentation for this class was generated from the following file:

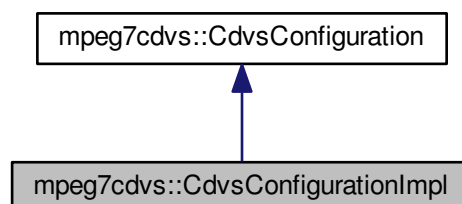
- [CdvsInterface.h](#)

## 7.20 mpeg7cdvs::CdvsConfigurationImpl Class Reference

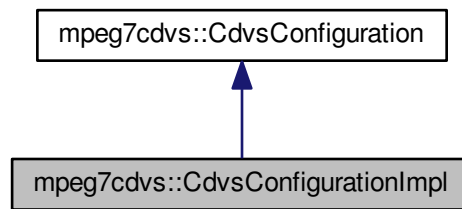
Interface to all configuration parameters for clients and servers.

```
#include <CdvsConfigurationImpl.h>
```

Inheritance diagram for mpeg7cdvs::CdvsConfigurationImpl:



Collaboration diagram for `mpeg7cdvs::CdvsConfigurationImpl`:



## Public Member Functions

- `CdvsConfigurationImpl` (`const char *configfile`)
- virtual `~CdvsConfigurationImpl` ()
- virtual const `Parameters` & `getParameters` (`int mode`) const  
*Get one of the `Parameters` instances (note that this class keeps an instance of all parameters for all modes).*
- virtual `Parameters` & `setParameters` (`int mode`)  
*Set some `Parameters` value for a specific mode.*

## Additional Inherited Members

### 7.20.1 Detailed Description

Interface to all configuration parameters for clients and servers.

#### Author

Massimo Balestri

#### Date

2014

### 7.20.2 Constructor & Destructor Documentation

7.20.2.1 `mpeg7cdvs::CdvsConfigurationImpl::CdvsConfigurationImpl ( const char * configfile )`

7.20.2.2 `virtual mpeg7cdvs::CdvsConfigurationImpl::~~CdvsConfigurationImpl ( )` `[inline]`, `[virtual]`

### 7.20.3 Member Function Documentation

7.20.3.1 `virtual const Parameters& mpeg7cdvs::CdvsConfigurationImpl::getParameters ( int mode ) const` `[virtual]`

Get one of the `Parameters` instances (note that this class keeps an instance of all parameters for all modes).

## Parameters

<i>mode</i>	the mode for which parameters are requested.
-------------	--

## Returns

a read-only instance of the parameters.

Implements [mpeg7cdvs::CdvsConfiguration](#).

7.20.3.2 virtual **Parameters&** mpeg7cdvs::CdvsConfigurationImpl::setParameters ( int *mode* ) [virtual]

Set some [Parameters](#) value for a specific mode.

## Parameters

<i>mode</i>	the mode for which parameters are requested.
-------------	--

## Returns

a modifiable instance of the parameters.

Implements [mpeg7cdvs::CdvsConfiguration](#).

The documentation for this class was generated from the following file:

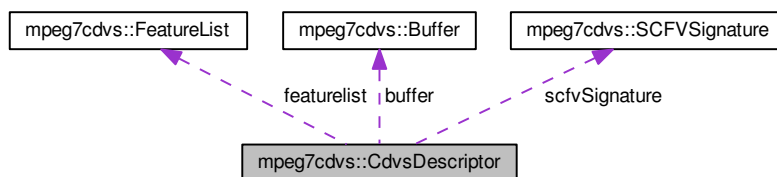
- [CdvsConfigurationImpl.h](#)

## 7.21 mpeg7cdvs::CdvsDescriptor Class Reference

Helper class to read/write/check CDVS descriptors according to the syntax defined in ISO/IEC 15938-13.

```
#include <CdvsDescriptor.h>
```

Collaboration diagram for mpeg7cdvs::CdvsDescriptor:



### Public Member Functions

- [CdvsDescriptor](#) ()
- virtual [~CdvsDescriptor](#) ()
- size\_t [encode](#) (const [Parameters](#) &params, [ImageBuffer](#) &image, const [SCFVFactory](#) &g\_factory)  
*Encode the CDVS descriptor.*
- size\_t [decode](#) (const [ParameterSet](#) &pset)  
*Decode the CDVS descriptor.*

- int [check](#) () const  
*Check the conformance of the descriptor to the syntax defined in ISO/IEC 15938-13.*
- void [clear](#) ()  
*Clear all data.*
- void [print](#) (const char \*title) const  
*Print the value of the syntax elements defined in ISO/IEC 15938-13.*
- unsigned int [getVersionID](#) () const  
*get the version ID*
- unsigned int [getModelID](#) () const  
*get the mode ID*
- bool [getGlobalHasBitSelection](#) () const  
*get the global descriptor bit selection flag*
- bool [getGlobalHasVariance](#) () const  
*get the global descriptor variance flag*
- bool [getRelevanceBitsPresent](#) () const  
*get the relevance bit flag*
- unsigned int [getOriginalImageXResolution](#) () const  
*get the original image X resolution*
- unsigned int [getOriginalImageYResolution](#) () const  
*get the original image Y resolution*
- unsigned int [getNumberOfLocalDescriptors](#) () const  
*get the number of local descriptors*
- unsigned int [getHistogramCountSize](#) () const  
*get the coordinate coding histogram count size*
- unsigned int [getHistogramMapSizeX](#) () const  
*get the coordinate coding map horizontal size*
- unsigned int [getHistogramMapSizeY](#) () const  
*get the coordinate coding map vertical size*
- void [setVersionID](#) (unsigned int vID)  
*set the version ID*
- void [setModelID](#) (unsigned int mID)  
*set the mode ID*
- void [setGlobalHasBitSelection](#) (bool gHasBS)  
*set the global descriptor bit selection flag*
- void [setGlobalHasVariance](#) (bool gHasV)  
*set the global descriptor variance flag*
- void [setRelevanceBitsPresent](#) (bool relevance)  
*set the relevance bit flag*
- void [setOriginalImageXResolution](#) (unsigned int oiXr)  
*set the original image X resolution*
- void [setOriginalImageYResolution](#) (unsigned int oiYr)  
*set the original image Y resolution*
- void [setNumberOfLocalDescriptors](#) (unsigned int nLD)  
*set the number of local descriptors*
- void [setHistogramCountSize](#) (unsigned int hCS)  
*set the coordinate coding histogram count size*
- void [setHistogramMapSizeX](#) (unsigned int hmsX)  
*set the coordinate coding map horizontal size*
- void [setHistogramMapSizeY](#) (unsigned int hmsY)  
*set the coordinate coding map vertical size*

## Data Fields

- [Buffer buffer](#)  
*the buffer containing the input/output bitstream*
- [FeatureList featurelist](#)  
*the list of key points*
- [SCFVSignature scfvSignature](#)  
*the global descriptor signature*

### 7.21.1 Detailed Description

Helper class to read/write/check CDVS descriptors according to the syntax defined in ISO/IEC 15938-13.

#### Author

Massimo Balestri (Telecom Italia)

#### Date

April, 2014

### 7.21.2 Constructor & Destructor Documentation

7.21.2.1 `mpeg7cdvs::CdvsDescriptor::CdvsDescriptor ( )`

7.21.2.2 `virtual mpeg7cdvs::CdvsDescriptor::~~CdvsDescriptor ( )` `[virtual]`

### 7.21.3 Member Function Documentation

7.21.3.1 `int mpeg7cdvs::CdvsDescriptor::check ( ) const`

Check the conformance of the descriptor to the syntax defined in ISO/IEC 15938-13.

#### Returns

the number of out of range fields

7.21.3.2 `void mpeg7cdvs::CdvsDescriptor::clear ( )`

Clear all data.

7.21.3.3 `size_t mpeg7cdvs::CdvsDescriptor::decode ( const ParameterSet & pset )`

Decode the CDVS descriptor.

#### Parameters

<i>pset</i>	set of parameters to apply for all modes from 0 to 6
-------------	--

#### Returns

the size of the consumed descriptor (bytes).

7.21.3.4 `size_t mpeg7cdvs::CdvsDescriptor::encode ( const Parameters & params, ImageBuffer & image, const SCFVFactory & g_factory )`

Encode the CDVS descriptor.

This implementation does not extract all the key points from the image, but only those selected for transmission by the feature selection stage.

Parameters

<i>params</i>	set of parameters to apply for one specific mode
<i>image</i>	the input image buffer
<i>g_factory</i>	the Global Descriptor factory instance that produces the GD signature for the specific mode selected in the parameters

Returns

the size of the produced descriptor (bytes).

7.21.3.5 `bool mpeg7cdvs::CdvsDescriptor::getGlobalHasBitSelection ( ) const`

get the global descriptor bit selection flag

7.21.3.6 `bool mpeg7cdvs::CdvsDescriptor::getGlobalHasVariance ( ) const`

get the global descriptor variance flag

7.21.3.7 `unsigned int mpeg7cdvs::CdvsDescriptor::getHistogramCountSize ( ) const`

get the coordinate coding histogram count size

7.21.3.8 `unsigned int mpeg7cdvs::CdvsDescriptor::getHistogramMapSizeX ( ) const`

get the coordinate coding map horizontal size

7.21.3.9 `unsigned int mpeg7cdvs::CdvsDescriptor::getHistogramMapSizeY ( ) const`

get the coordinate coding map vertical size

7.21.3.10 `unsigned int mpeg7cdvs::CdvsDescriptor::getModelID ( ) const`

get the mode ID

7.21.3.11 `unsigned int mpeg7cdvs::CdvsDescriptor::getNumberOfLocalDescriptors ( ) const`

get the number of local descriptors

7.21.3.12 `unsigned int mpeg7cdvs::CdvsDescriptor::getOriginalImageXResolution ( ) const`

get the original image X resolution

7.21.3.13 unsigned int mpeg7cdvs::CdvsDescriptor::getOriginalImageYResolution ( ) const

get the original image Y resolution

7.21.3.14 bool mpeg7cdvs::CdvsDescriptor::getRelevanceBitsPresent ( ) const

get the relevance bit flag

7.21.3.15 unsigned int mpeg7cdvs::CdvsDescriptor::getVersionID ( ) const

get the version ID

7.21.3.16 void mpeg7cdvs::CdvsDescriptor::print ( const char \* *title* ) const

Print the value of the syntax elements defined in ISO/IEC 15938-13.

Parameters

<i>title</i>	the title to print as header information
--------------	--

7.21.3.17 void mpeg7cdvs::CdvsDescriptor::setGlobalHasBitSelection ( bool *gHasBS* )

set the global descriptor bit selection flag

7.21.3.18 void mpeg7cdvs::CdvsDescriptor::setGlobalHasVariance ( bool *gHasV* )

set the global descriptor variance flag

7.21.3.19 void mpeg7cdvs::CdvsDescriptor::setHistogramCountSize ( unsigned int *hCS* )

set the coordinate coding histogram count size

7.21.3.20 void mpeg7cdvs::CdvsDescriptor::setHistogramMapSizeX ( unsigned int *hmsX* )

set the coordinate coding map horizontal size

7.21.3.21 void mpeg7cdvs::CdvsDescriptor::setHistogramMapSizeY ( unsigned int *hmsY* )

set the coordinate coding map vertical size

7.21.3.22 void mpeg7cdvs::CdvsDescriptor::setModelID ( unsigned int *mID* )

set the mode ID

7.21.3.23 void mpeg7cdvs::CdvsDescriptor::setNumberOfLocalDescriptors ( unsigned int *nLD* )

set the number of local descriptors

7.21.3.24 void mpeg7cdvs::CdvsDescriptor::setOriginalImageXResolution ( unsigned int *oiXr* )

set the original image X resolution

7.21.3.25 void mpeg7cdvs::CdvsDescriptor::setOriginalImageYResolution ( unsigned int *oiYr* )

set the original image Y resolution

7.21.3.26 void mpeg7cdvs::CdvsDescriptor::setRelevanceBitsPresent ( bool *relevance* )

set the relevance bit flag

7.21.3.27 void mpeg7cdvs::CdvsDescriptor::setVersionID ( unsigned int *vID* )

set the version ID

## 7.21.4 Field Documentation

7.21.4.1 Buffer mpeg7cdvs::CdvsDescriptor::buffer

the buffer containing the input/output bitstream

7.21.4.2 FeatureList mpeg7cdvs::CdvsDescriptor::featurelist

the list of key points

7.21.4.3 SCFVSignature mpeg7cdvs::CdvsDescriptor::scfvSignature

the global descriptor signature

The documentation for this class was generated from the following file:

- [CdvsDescriptor.h](#)

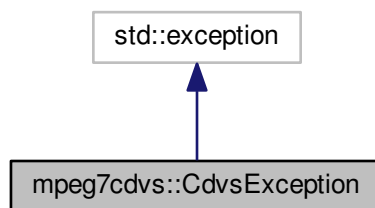
## 7.22 mpeg7cdvs::CdvsException Class Reference

Class defining a specific exception for CDVS.

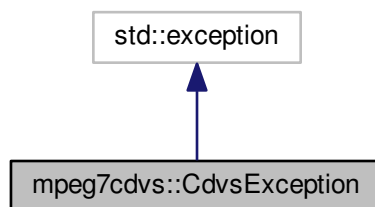
```
#include <CdvsException.h>
```



Inheritance diagram for mpeg7cdvs::CdvsException:



Collaboration diagram for mpeg7cdvs::CdvsException:



## Public Member Functions

- [CdvsException](#) (std::string str)  
*Create a new CDVS exception.*
- virtual [~CdvsException](#) () throw ()
- const char \* [what](#) () const throw ()  
*Get the exception message.*

### 7.22.1 Detailed Description

Class defining a specific exception for CDVS.

### 7.22.2 Constructor & Destructor Documentation

#### 7.22.2.1 mpeg7cdvs::CdvsException::CdvsException ( std::string str ) [inline]

Create a new CDVS exception.

## Parameters

<i>str</i>	the exception message string.
------------	-------------------------------

7.22.2.2 virtual `mpeg7cdvs::CdvsException::~CdvsException ( ) throw` `[inline],[virtual]`

### 7.22.3 Member Function Documentation

7.22.3.1 `const char* mpeg7cdvs::CdvsException::what ( ) const throw` `[inline]`

Get the exception message.

The documentation for this class was generated from the following file:

- [CdvsException.h](#)

## 7.23 mpeg7cdvs::CDVSPPOINT Struct Reference

A structure containing the x and y coordinate of a point in the image.

```
#include <CdvsPoint.h>
```

### Data Fields

- float [x](#)  
*the X coordinate*
- float [y](#)  
*the Y coordinate*

### 7.23.1 Detailed Description

A structure containing the x and y coordinate of a point in the image.

### 7.23.2 Field Documentation

7.23.2.1 `float mpeg7cdvs::CDVSPPOINT::x`

the X coordinate

7.23.2.2 `float mpeg7cdvs::CDVSPPOINT::y`

the Y coordinate

The documentation for this struct was generated from the following file:

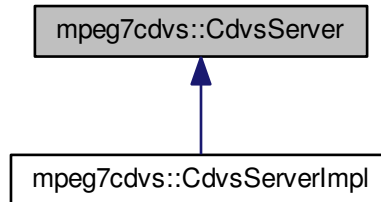
- [CdvsPoint.h](#)

## 7.24 mpeg7cdvs::CdvsServer Class Reference

Interface to the server-side functionality of the CDVS Library.

```
#include <CdvsInterface.h>
```

Inheritance diagram for mpeg7cdvs::CdvsServer:



## Public Member Functions

- virtual `~CdvsServer` ()
- virtual `size_t decode` (`CdvsDescriptor` &output, const char \*fname) const =0  
*Decode a compressed query descriptor stored in a file.*
- virtual `size_t decode` (`CdvsDescriptor` &output, const unsigned char \*bitstream=NULL, int size=0) const =0  
*Decode a compressed reference descriptor stored either in the bitstream parameter or in the `CdvsDescriptor` input/output `Buffer`.*
- virtual `PointPairs match` (const `CdvsDescriptor` &queryDescriptor, const `CdvsDescriptor` &refDescriptor, const `CDVSPPOINT` \*r\_bbox=NULL, `CDVSPPOINT` \*proj\_bbox=NULL, int matchType=`MATCH_TYPE_DEFAULT`) const =0  
*Pair-wise descriptor matching & localization function.*
- virtual `PointPairs match` (const `CdvsDescriptor` &queryDescriptor, unsigned int index, const `CDVSPPOINT` \*r\_bbox=NULL, `CDVSPPOINT` \*proj\_bbox=NULL, int matchType=`MATCH_TYPE_LOCAL`) const =0  
*Pair-wise descriptor matching & localization using a DB image as reference.*
- virtual void `createDB` (int mode, int reserve)=0  
*Create a `Database` of CDVS Descriptors for retrieval.*
- virtual unsigned int `addDescriptorToDB` (const `CdvsDescriptor` &refDescriptor, const char \*referenceImageId)=0  
*Add the given reference descriptor to the Data Base of reference images.*
- virtual bool `isDescriptorInDB` (const char \*referenceImageId) const =0  
*Verify if a given image is stored in the DB.*
- virtual bool `replaceDescriptorInDB` (const `CdvsDescriptor` &refDescriptor, const char \*referenceImageId, const char \*oldImageId=NULL)=0  
*Replace a given image in the DB with another one.*
- virtual void `clearDB` ()=0  
*Clear the DB removing all images.*
- virtual void `commitDB` ()=0  
*Commit all changes into the DB.*
- virtual void `storeDB` (const char \*localname, const char \*globalname) const =0  
*Store the Data Base permanently into a pair of files.*
- virtual void `loadDB` (const char \*localname, const char \*globalname)=0  
*Load the Data Base from a pair of files.*
- virtual `size_t sizeofDB` () const =0  
*Get the number of descriptors currently stored in the retrieval Data Base.*

- virtual int [retrieve](#) (std::vector< [RetrievalData](#) > &results, const [CdvsDescriptor](#) &queryDescriptor, unsigned int max\_matches) const =0  
*Retrieval function.*
- virtual std::string [getImageId](#) (unsigned int index) const =0  
*Get the id corresponding to the given image index in the DB.*

## Static Public Member Functions

- static [CdvsServer](#) \* [cdvsServerFactory](#) (const [CdvsConfiguration](#) \*config, bool twoWayMatch=true)  
*Create an instance of a CDVS Server for matching and retrieval of CDVS descriptors.*

### 7.24.1 Detailed Description

Interface to the server-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.24.2 Constructor & Destructor Documentation

7.24.2.1 virtual [mpeg7cdvs::CdvsServer::~CdvsServer](#) ( ) [inline], [virtual]

### 7.24.3 Member Function Documentation

7.24.3.1 virtual unsigned int [mpeg7cdvs::CdvsServer::addDescriptorToDB](#) ( const [CdvsDescriptor](#) & *refDescriptor*, const char \* *referenceImageId* ) [pure virtual]

Add the given reference descriptor to the Data Base of reference images.

#### Parameters

<i>refDescriptor</i>	the reference descriptor
<i>referenceImageId</i>	the string that identifies this image; may be a pathname or a numeric ID but must be expressed as text.

#### Returns

the index of the reference image in the DB

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

7.24.3.2 static [CdvsServer](#)\* [mpeg7cdvs::CdvsServer::cdvsServerFactory](#) ( const [CdvsConfiguration](#) \* *config*, bool *twoWayMatch* =true ) [static]

Create an instance of a CDVS Server for matching and retrieval of CDVS descriptors.

The calling entity takes ownership of the instance (i.e. must delete the instance when not used anymore).

## Parameters

<i>config</i>	the configuration that will be used to produce descriptors.
<i>twoWayMatch</i>	select one-way or two-way matching; default is two-way.

## Returns

a pointer to the Cdvs Server instance

## 7.24.3.3 virtual void mpeg7cdvs::CdvsServer::clearDB ( ) [pure virtual]

Clear the DB removing all images.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

## 7.24.3.4 virtual void mpeg7cdvs::CdvsServer::commitDB ( ) [pure virtual]

Commit all changes into the DB.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

## 7.24.3.5 virtual void mpeg7cdvs::CdvsServer::createDB ( int mode, int reserve ) [pure virtual]

Create a [Database](#) of CDVS Descriptors for retrieval.

## Parameters

<i>mode</i>	the mode identifier of all descriptors that will be stored in the DB;
<i>reserve</i>	the estimate number of CDVS Descriptor that will constitute the DB; the code will reserve a corresponding space in the DB.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

## 7.24.3.6 virtual size\_t mpeg7cdvs::CdvsServer::decode ( CdvsDescriptor &amp; output, const char \* fname ) const [pure virtual]

Decode a compressed query descriptor stored in a file.

## Parameters

<i>fname</i>	the input file name
<i>output</i>	the decoded <a href="#">CdvsDescriptor</a>

## Returns

the size of the consumed descriptor (bytes).

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

## 7.24.3.7 virtual size\_t mpeg7cdvs::CdvsServer::decode ( CdvsDescriptor &amp; output, const unsigned char \* bitstream = NULL, int size = 0 ) const [pure virtual]

Decode a compressed reference descriptor stored either in the bitstream parameter or in the [CdvsDescriptor](#) input/output [Buffer](#).

## Parameters

<i>output</i>	the decoded <a href="#">CdvsDescriptor</a>
<i>bitstream</i>	a buffer containing an encoded <a href="#">CdvsDescriptor</a> bitstream (optional parameter; if missing, the "buffer" member variable of <a href="#">CdvsDescriptor</a> will be used instead)
<i>size</i>	size in bytes of the bitstream buffer (must be specified only if bitstream is not null)

## Returns

the size of the consumed descriptor (bytes).

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

7.24.3.8 `virtual std::string mpeg7cdvs::CdvsServer::getImageId ( unsigned int index ) const` `[pure virtual]`

Get the id corresponding to the given image index in the DB.

## Parameters

<i>index</i>	the index in the DB of the image
--------------	----------------------------------

## Returns

a string containing the identifier of the image

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

7.24.3.9 `virtual bool mpeg7cdvs::CdvsServer::isDescriptorInDB ( const char * referenceImageId ) const` `[pure virtual]`

Verify if a given image is stored in the DB.

## Parameters

<i>referenceImageId</i>	the string that identifies this image; may be a pathname or a numeric ID but must be expressed as text.
-------------------------	---

## Returns

true if the image is present.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

7.24.3.10 `virtual void mpeg7cdvs::CdvsServer::loadDB ( const char * localname, const char * globalname )` `[pure virtual]`

Load the Data Base from a pair of files.

## Parameters

<i>localname</i>	the name of the local descriptors file;
<i>globalname</i>	the name of the global descriptors file;

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

7.24.3.11 `virtual PointPairs mpeg7cdvs::CdvsServer::match ( const CdvsDescriptor & queryDescriptor, const CdvsDescriptor & refDescriptor, const CDVSPPOINT * r_bbox = NULL, CDVSPPOINT * proj_bbox = NULL, int matchType = MATCH_TYPE_DEFAULT ) const` `[pure virtual]`

Pair-wise descriptor matching & localization function.

## Parameters

<i>queryDescriptor</i>	the query descriptor
<i>refDescriptor</i>	the reference descriptor
<i>r_bbox</i>	bounding box of object of interest in the second (reference) image; replaced by the full image coordinates if NULL.
<i>proj_bbox</i>	buffer to contain parameters of bounding box for a match projected in the coordinate system of the first (query) image; ignored if NULL.
<i>matchType</i>	type of matching; may be MATCH_TYPE_DEFAULT, MATCH_TYPE_BOTH, MATCH_TYPE_LOCAL, MATCH_TYPE_GLOBAL. Default is MATCH_TYPE_DEFAULT in this case.

## Returns

an instance of [PointPairs](#) which contains all matching points, plus local and global scores.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

```
7.24.3.12 virtual PointPairs mpeg7cdvs::CdvsServer::match ( const CdvsDescriptor & queryDescriptor, unsigned
int index, const CDVSPPOINT * r_bbox = NULL, CDVSPPOINT * proj_bbox = NULL, int matchType =
MATCH_TYPE_LOCAL ) const [pure virtual]
```

Pair-wise descriptor matching & localization using a DB image as reference.

This method can be called after a retrieval operation, to localize the retrieved object(s) in the query image. The default match type in this case is MATCH\_TYPE\_LOCAL.

## Parameters

<i>queryDescriptor</i>	the query descriptor
<i>index</i>	index of the reference descriptor in the DB
<i>r_bbox</i>	bounding box of object of interest in the DB image; replaced by the full image coordinates if NULL.
<i>proj_bbox</i>	buffer to contain parameters of bounding box for a match projected in the coordinate system of the query image; ignored if NULL.
<i>matchType</i>	type of matching; may be MATCH_TYPE_DEFAULT, MATCH_TYPE_BOTH, MATCH_TYPE_LOCAL, MATCH_TYPE_GLOBAL. Default is MATCH_TYPE_LOCAL in this case.

## Returns

an instance of [PointPairs](#) which contains all matching points, plus local and global scores.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

```
7.24.3.13 virtual bool mpeg7cdvs::CdvsServer::replaceDescriptorInDB ( const CdvsDescriptor & refDescriptor, const char *
referencelImageId, const char * oldImageId = NULL ) [pure virtual]
```

Replace a given image in the DB with another one.

If the image is not present no operation is performed.

## Parameters

<i>refDescriptor</i>	the reference descriptor of the new image
<i>referencelImageId</i>	the string that identifies the new image

<i>oldImageId</i>	the string that identifies the old image to be replaced; if NULL, referenceImageId will be also used as name of the image to replace
-------------------	--

**Returns**

true if the image was present (and its descriptor has been replaced).

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

**7.24.3.14** `virtual int mpeg7cdvs::CdvsServer::retrieve ( std::vector< RetrievalData > & results, const CdvsDescriptor & queryDescriptor, unsigned int max_matches ) const [pure virtual]`

Retrieval function.

**Notes:**

- it is assumed that database index is already pre-loaded and available through globals.
- query descriptor is also pre-loaded and passed via input parameters
- the task of this function is to produce a list of matching images in the database using only query descriptor, index, and descriptors of images stored in the database (included in the index)

**Parameters**

<i>results</i>	vector of information data about matching images (in order of relevance)
<i>queryDescriptor</i>	the query descriptor to be used as input query data of the retrieval operation
<i>max_matches</i>	- maximum number of matches to include in the list of results

**Returns**

number of matches found

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

**7.24.3.15** `virtual size_t mpeg7cdvs::CdvsServer::sizeofDB ( ) const [pure virtual]`

Get the number of descriptors currently stored in the retrieval Data Base.

**Returns**

the number of descriptors in the DB.

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

**7.24.3.16** `virtual void mpeg7cdvs::CdvsServer::storeDB ( const char * localname, const char * globalname ) const [pure virtual]`

Store the Data Base permanently into a pair of files.

**Parameters**

<i>localname</i>	the name of the local descriptors file;
<i>globalname</i>	the name of the global descriptors file;

Implemented in [mpeg7cdvs::CdvsServerImpl](#).

The documentation for this class was generated from the following file:

- [CdvsInterface.h](#)

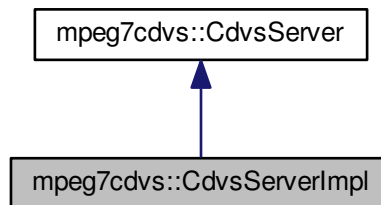


## 7.25 mpeg7cdvs::CdvsServerImpl Class Reference

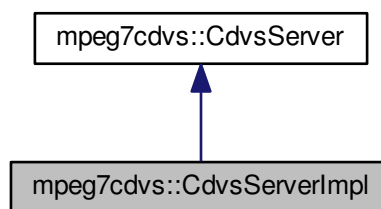
Implementation of the high level interface to the server-side functionality of the CDVS Library.

```
#include <CdvsServerImpl.h>
```

Inheritance diagram for mpeg7cdvs::CdvsServerImpl:



Collaboration diagram for mpeg7cdvs::CdvsServerImpl:



### Public Member Functions

- [CdvsServerImpl](#) (const [CdvsConfiguration](#) \*config, bool twoWayMatch=true)
- virtual [~CdvsServerImpl](#) ()
- virtual size\_t [decode](#) ([CdvsDescriptor](#) &output, const char \*fname) const  
*Decode a compressed query descriptor stored in a file.*
- virtual size\_t [decode](#) ([CdvsDescriptor](#) &output, const unsigned char \*bitstream, int size) const  
*Decode a compressed reference descriptor stored either in the bitstream parameter or in the [CdvsDescriptor](#) input/output [Buffer](#).*
- virtual [PointPairs](#) [match](#) (const [CdvsDescriptor](#) &queryDescriptor, const [CdvsDescriptor](#) &refDescriptor, const [CDVSPPOINT](#) \*r\_bbox, [CDVSPPOINT](#) \*proj\_bbox, int matchType) const  
*Pair-wise descriptor matching & localization function.*
- virtual [PointPairs](#) [match](#) (const [CdvsDescriptor](#) &queryDescriptor, unsigned int index, const [CDVSPPOINT](#) \*r\_bbox, [CDVSPPOINT](#) \*proj\_bbox, int matchType) const  
*Pair-wise descriptor matching & localization using a DB image as reference.*
- virtual void [createDB](#) (int mode, int reserve)

- Create a [Database](#) of CDVS Descriptors for retrieval.

  - virtual unsigned int [addDescriptorToDB](#) (const [CdvsDescriptor](#) &refDescriptor, const char \*referenceImageld)

Add the given reference descriptor to the Data Base of reference images.
- virtual bool [isDescriptorInDB](#) (const char \*referenceImageld) const

Verify if a given image is stored in the DB.
- virtual bool [replaceDescriptorInDB](#) (const [CdvsDescriptor](#) &refDescriptor, const char \*referenceImageld, const char \*oldImageld)

Replace a given image in the DB with another one.
- virtual void [clearDB](#) ()

Clear the DB removing all images.
- virtual void [storeDB](#) (const char \*localname, const char \*globalname) const

Store the Data Base permanently into a pair of files.
- virtual void [loadDB](#) (const char \*localname, const char \*globalname)

Load the Data Base from a pair of files.
- virtual size\_t [sizeofDB](#) () const

Get the number of descriptors currently stored in the retrieval Data Base.
- virtual int [retrieve](#) (std::vector< [RetrievalData](#) > &results, const [CdvsDescriptor](#) &cdvsDescriptor, unsigned int max\_matches) const

Retrieval function.
- virtual std::string [getImageld](#) (unsigned int index) const

Get the id corresponding to the given image index in the DB.
- virtual void [commitDB](#) ()

Commit all changes into the DB.

## Additional Inherited Members

### 7.25.1 Detailed Description

Implementation of the high level interface to the server-side functionality of the CDVS Library.

#### Author

Massimo Balestri

#### Date

2014

### 7.25.2 Constructor & Destructor Documentation

7.25.2.1 `mpeg7cdvs::CdvsServerImpl::CdvsServerImpl ( const CdvsConfiguration * config, bool twoWayMatch = true )`

7.25.2.2 `virtual mpeg7cdvs::CdvsServerImpl::~~CdvsServerImpl ( ) [virtual]`

### 7.25.3 Member Function Documentation

7.25.3.1 `virtual unsigned int mpeg7cdvs::CdvsServerImpl::addDescriptorToDB ( const CdvsDescriptor & refDescriptor, const char * referenceImageld ) [virtual]`

Add the given reference descriptor to the Data Base of reference images.

## Parameters

<i>refDescriptor</i>	the reference descriptor
<i>referenceImageId</i>	the string that identifies this image; may be a pathname or a numeric ID but must be expressed as text.

## Returns

the index of the reference image in the DB

Implements [mpeg7cdvs::CdvsServer](#).

### 7.25.3.2 virtual void mpeg7cdvs::CdvsServerImpl::clearDB ( ) [virtual]

Clear the DB removing all images.

Implements [mpeg7cdvs::CdvsServer](#).

### 7.25.3.3 virtual void mpeg7cdvs::CdvsServerImpl::commitDB ( ) [virtual]

Commit all changes into the DB.

Implements [mpeg7cdvs::CdvsServer](#).

### 7.25.3.4 virtual void mpeg7cdvs::CdvsServerImpl::createDB ( int *mode*, int *reserve* ) [virtual]

Create a [Database](#) of CDVS Descriptors for retrieval.

## Parameters

<i>mode</i>	the mode identifier of all descriptors that will be stored in the DB;
<i>reserve</i>	the estimate number of CDVS Descriptor that will constitute the DB; the code will reserve a corresponding space in the DB.

Implements [mpeg7cdvs::CdvsServer](#).

### 7.25.3.5 virtual size\_t mpeg7cdvs::CdvsServerImpl::decode ( CdvsDescriptor & *output*, const char \* *fname* ) const [virtual]

Decode a compressed query descriptor stored in a file.

## Parameters

<i>fname</i>	the input file name
<i>output</i>	the decoded <a href="#">CdvsDescriptor</a>

## Returns

the size of the consumed descriptor (bytes).

Implements [mpeg7cdvs::CdvsServer](#).

### 7.25.3.6 virtual size\_t mpeg7cdvs::CdvsServerImpl::decode ( CdvsDescriptor & *output*, const unsigned char \* *bitstream*, int *size* ) const [virtual]

Decode a compressed reference descriptor stored either in the bitstream parameter or in the [CdvsDescriptor](#) input/output [Buffer](#).

## Parameters

<i>output</i>	the decoded <a href="#">CdvsDescriptor</a>
<i>bitstream</i>	a buffer containing an encoded <a href="#">CdvsDescriptor</a> bitstream (optional parameter; if missing, the "buffer" member variable of <a href="#">CdvsDescriptor</a> will be used instead)
<i>size</i>	size in bytes of the bitstream buffer (must be specified only if bitstream is not null)

## Returns

the size of the consumed descriptor (bytes).

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.7** `virtual std::string mpeg7cdvs::CdvsServerImpl::getImageId ( unsigned int index ) const` `[virtual]`

Get the id corresponding to the given image index in the DB.

## Parameters

<i>index</i>	the index in the DB of the image
--------------	----------------------------------

## Returns

a string containing the identifier of the image

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.8** `virtual bool mpeg7cdvs::CdvsServerImpl::isDescriptorInDB ( const char * referenceImageId ) const` `[virtual]`

Verify if a given image is stored in the DB.

## Parameters

<i>referenceImageId</i>	the string that identifies this image; may be a pathname or a numeric ID but must be expressed as text.
-------------------------	---

## Returns

true if the image is present.

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.9** `virtual void mpeg7cdvs::CdvsServerImpl::loadDB ( const char * localname, const char * globalname )` `[virtual]`

Load the Data Base from a pair of files.

## Parameters

<i>localname</i>	the name of the local descriptors file;
<i>globalname</i>	the name of the global descriptors file;

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.10** `virtual PointPairs mpeg7cdvs::CdvsServerImpl::match ( const CdvsDescriptor & queryDescriptor, const CdvsDescriptor & refDescriptor, const CDVSPPOINT * r_bbox, CDVSPPOINT * proj_bbox, int matchType ) const` `[virtual]`

Pair-wise descriptor matching & localization function.

## Parameters

<i>queryDescriptor</i>	the query descriptor
<i>refDescriptor</i>	the reference descriptor
<i>r_bbox</i>	bounding box of object of interest in the second (reference) image; replaced by the full image coordinates if NULL.
<i>proj_bbox</i>	buffer to contain parameters of bounding box for a match projected in the coordinate system of the first (query) image; ignored if NULL.
<i>matchType</i>	type of matching; may be MATCH_TYPE_DEFAULT, MATCH_TYPE_BOTH, MATCH_TYPE_LOCAL, MATCH_TYPE_GLOBAL. Default is MATCH_TYPE_DEFAULT in this case.

## Returns

an instance of [PointPairs](#) which contains all matching points, plus local and global scores.

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.11** `virtual PointPairs mpeg7cdvs::CdvsServerImpl::match ( const CdvsDescriptor & queryDescriptor, unsigned int index, const CDVSPPOINT * r_bbox, CDVSPPOINT * proj_bbox, int matchType ) const [virtual]`

Pair-wise descriptor matching & localization using a DB image as reference.

This method can be called after a retrieval operation, to localize the retrieved object(s) in the query image. The default match type in this case is MATCH\_TYPE\_LOCAL.

## Parameters

<i>queryDescriptor</i>	the query descriptor
<i>index</i>	index of the reference descriptor in the DB
<i>r_bbox</i>	bounding box of object of interest in the DB image; replaced by the full image coordinates if NULL.
<i>proj_bbox</i>	buffer to contain parameters of bounding box for a match projected in the coordinate system of the query image; ignored if NULL.
<i>matchType</i>	type of matching; may be MATCH_TYPE_DEFAULT, MATCH_TYPE_BOTH, MATCH_TYPE_LOCAL, MATCH_TYPE_GLOBAL. Default is MATCH_TYPE_LOCAL in this case.

## Returns

an instance of [PointPairs](#) which contains all matching points, plus local and global scores.

Implements [mpeg7cdvs::CdvsServer](#).

**7.25.3.12** `virtual bool mpeg7cdvs::CdvsServerImpl::replaceDescriptorInDB ( const CdvsDescriptor & refDescriptor, const char * referenceImageId, const char * oldImageId ) [virtual]`

Replace a given image in the DB with another one.

If the image is not present no operation is performed.

## Parameters

<i>refDescriptor</i>	the reference descriptor of the new image
<i>referenceImageId</i>	the string that identifies the new image

<i>oldImageId</i>	the string that identifies the old image to be replaced; if NULL, <i>referenceImageId</i> will be also used as name of the image to replace
-------------------	---

**Returns**

true if the image was present (and its descriptor has been replaced).

Implements [mpeg7cdvs::CdvsServer](#).

7.25.3.13 `virtual int mpeg7cdvs::CdvsServerImpl::retrieve ( std::vector< RetrievalData > & results, const CdvsDescriptor & queryDescriptor, unsigned int max_matches ) const [virtual]`

Retrieval function.

**Notes:**

- it is assumed that database index is already pre-loaded and available through globals.
- query descriptor is also pre-loaded and passed via input parameters
- the task of this function is to produce a list of matching images in the database using only query descriptor, index, and descriptors of images stored in the database (included in the index)

**Parameters**

<i>results</i>	vector of information data about matching images (in order of relevance)
<i>queryDescriptor</i>	the query descriptor to be used as input query data of the retrieval operation
<i>max_matches</i>	- maximum number of matches to include in the list of results

**Returns**

number of matches found

Implements [mpeg7cdvs::CdvsServer](#).

7.25.3.14 `virtual size_t mpeg7cdvs::CdvsServerImpl::sizeofDB ( ) const [virtual]`

Get the number of descriptors currently stored in the retrieval Data Base.

**Returns**

the number of descriptors in the DB.

Implements [mpeg7cdvs::CdvsServer](#).

7.25.3.15 `virtual void mpeg7cdvs::CdvsServerImpl::storeDB ( const char * localname, const char * globalname ) const [virtual]`

Store the Data Base permanently into a pair of files.

**Parameters**

<i>localname</i>	the name of the local descriptors file;
<i>globalname</i>	the name of the global descriptors file;

Implements [mpeg7cdvs::CdvsServer](#).

The documentation for this class was generated from the following file:

- [CdvsServerImpl.h](#)

## 7.26 mpeg7cdvs::CsscCoordinateCoding::CircularSumContext Struct Reference

Basic structure for Cssc.

```
#include <CsscCoordinateCoding.h>
```

### Data Fields

- long [vCount](#) [SUM\_HIST\_COUNT\_SIZE]  
*Histogram Count Arithmetic Coding model initialization data.*
- long [vInitialMap](#) [2]  
*Circular scan map Arithmetic Coding model initialization data (initial)*
- long [vMap](#) [MAXIMUM\_SUM\_CONTEXT+1][2]  
*Circular scan map Arithmetic Coding model initialization data (following)*

### 7.26.1 Detailed Description

Basic structure for Cssc.

### 7.26.2 Field Documentation

7.26.2.1 long mpeg7cdvs::CsscCoordinateCoding::CircularSumContext::vCount[SUM\_HIST\_COUNT\_SIZE]

Histogram Count Arithmetic Coding model initialization data.

7.26.2.2 long mpeg7cdvs::CsscCoordinateCoding::CircularSumContext::vInitialMap[2]

Circular scan map Arithmetic Coding model initialization data (initial)

7.26.2.3 long mpeg7cdvs::CsscCoordinateCoding::CircularSumContext::vMap[MAXIMUM\_SUM\_CONTEXT+1][2]

Circular scan map Arithmetic Coding model initialization data (following)

The documentation for this struct was generated from the following file:

- [CsscCoordinateCoding.h](#)

## 7.27 mpeg7cdvs::CompressedFeatureList Class Reference

Container class for all compressed features of an image.

```
#include <FeatureList.h>
```

### Public Member Functions

- [CompressedFeatureList](#) ()  
*default constructor*
- [CompressedFeatureList](#) (int nFeatures, int descLen)  
*parametric constructor (allocates memory)*
- [CompressedFeatureList](#) (const [CompressedFeatureList](#) &a)  
*copy constructor*

- virtual `~CompressedFeatureList ()`
- `CompressedFeatureList (const FeatureList &other, bool relevantOnly=false)`  
*Copy constructor from `FeatureList`, optionally including relevance sorting.*
- `CompressedFeatureList & operator= (CompressedFeatureList other)`  
*Assignment operator.*
- void `swap (CompressedFeatureList &other)`  
*Swap this instance with another.*
- int `nFeatures () const`  
*Get the number of features.*
- int `descrBytes () const`  
*Get the size (number of bytes) of each feature stored in this compressed feature list.*
- void `setFilename (const char *filename)`  
*Set the name of the image (to be stored for subsequent retrieval).*
- int `matchDescriptors_oneWay (PointPairs &pairs, const CompressedFeatureList &otherList, float ratio-Threshold) const`  
*Match the features of the current list with the ones contained in otherList in a one way fashion.*
- int `matchDescriptors_twoWay (PointPairs &pairs, const CompressedFeatureList &otherList, float ratio-Threshold) const`  
*Match the features of the current list with the ones contained in otherList in a two way fashion.*
- std::streamoff `readFromFile (char *filename)`  
*Read an entire feature list from the given file.*
- std::streamoff `read (std::istream &sin)`  
*Read an entire feature list from the given input stream.*
- std::streamoff `writeToFile (char *filename) const`  
*Write an entire feature list into the given file.*
- std::streamoff `write (std::ostream &sout) const`  
*Write an entire feature list into the given output stream.*
- void `print () const`  
*Print a summary of the content.*

## Static Public Member Functions

- static int `getDistance (const unsigned char *mine, const unsigned char *other, int nbytes)`  
*Get the distance of one feature from another feature.*

## Data Fields

- unsigned short \* `Ycoord`  
*the X coordinate of the ALP keypoint*
- unsigned short \* `Xcoord`  
*the Y coordinate of the ALP keypoint*
- unsigned char \* `features`  
*all compressed features*
- std::string `imagefile`  
*pathname of the image file.*
- int `imageHeight`  
*the (possibly scaled) image height.*
- int `imageWidth`  
*the (possibly scaled) image width.*
- int `originalHeight`  
*the original image height.*
- int `originalWidth`  
*the original image width.*



## Protected Attributes

- int [numFeatures](#)  
*number of features of this image*
- int [nDescLength](#)  
*descriptor length in bytes.*

### 7.27.1 Detailed Description

Container class for all compressed features of an image.

This is only used in the database in order to minimize the memory usage.

Date

2014

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 mpeg7cdvs::CompressedFeatureList::CompressedFeatureList ( )

default constructor

#### 7.27.2.2 mpeg7cdvs::CompressedFeatureList::CompressedFeatureList ( int *nFeatures*, int *descLen* )

parametric constructor (allocates memory)

#### 7.27.2.3 mpeg7cdvs::CompressedFeatureList::CompressedFeatureList ( const **CompressedFeatureList** & *a* )

copy constructor

#### 7.27.2.4 virtual mpeg7cdvs::CompressedFeatureList::~~CompressedFeatureList ( ) [virtual]

#### 7.27.2.5 mpeg7cdvs::CompressedFeatureList::CompressedFeatureList ( const **FeatureList** & *other*, bool *relevantOnly* = false )

Copy constructor from [FeatureList](#), optionally including relevance sorting.

Parameters

<i>other</i>	the other <a href="#">FeatureList</a> instance to copy
<i>relevantOnly</i>	if true, this method copies from a <a href="#">FeatureList</a> instance only the features having the highest relevance

### 7.27.3 Member Function Documentation

#### 7.27.3.1 int mpeg7cdvs::CompressedFeatureList::descrBytes ( ) const [inline]

Get the size (number of bytes) of each feature stored in this compressed feature list.

Returns

the size in bytes of the features.

References [nDescLength](#).

7.27.3.2 `static int mpeg7cdvs::CompressedFeatureList::getDistance ( const unsigned char * mine, const unsigned char * other, int nbytes ) [static]`

Get the distance of one feature from another feature.

## Parameters

<i>mine</i>	my feature
<i>other</i>	the other feature
<i>nbytes</i>	the number of bytes to use as input data

## Returns

the distance

**7.27.3.3** `int mpeg7cdvs::CompressedFeatureList::matchDescriptors_oneWay ( PointPairs & pairs, const CompressedFeatureList & otherList, float ratioThreshold ) const`

[Match](#) the features of the current list with the ones contained in otherList in a one way fashion.

The coordinates of the matching points are stored in the pairs parameter.

## Parameters

<i>pairs</i>	computed matching pairs of points
<i>otherList</i>	the other list.
<i>ratioThreshold</i>	the threshold used in the ratio test.

## Returns

the number of matched features.

**7.27.3.4** `int mpeg7cdvs::CompressedFeatureList::matchDescriptors_twoWay ( PointPairs & pairs, const CompressedFeatureList & otherList, float ratioThreshold ) const`

[Match](#) the features of the current list with the ones contained in otherList in a two way fashion.

The coordinates of the matching points are stored in the pairs parameter.

## Parameters

<i>pairs</i>	computed matching pairs of points
<i>otherList</i>	the other list.
<i>ratioThreshold</i>	the threshold used in the ratio test.

## Returns

the number of matched features.

**7.27.3.5** `int mpeg7cdvs::CompressedFeatureList::nFeatures ( ) const` `[inline]`

Get the number of features.

References numFeatures.

**7.27.3.6** `CompressedFeatureList& mpeg7cdvs::CompressedFeatureList::operator= ( CompressedFeatureList other )`

Assignment operator.

## Parameters

<i>other</i>	the other <a href="#">CompressedFeatureList</a> instance
--------------	--

## Returns

a [CompressedFeatureList](#) instance

### 7.27.3.7 void mpeg7cdvs::CompressedFeatureList::print ( ) const

Print a summary of the content.

### 7.27.3.8 std::streamoff mpeg7cdvs::CompressedFeatureList::read ( std::istream & *sin* )

Read an entire feature list from the given input stream.

## Parameters

<i>sin</i>	the input stream.
------------	-------------------

## Returns

the number of bytes that have been read from the input stream.

### 7.27.3.9 std::streamoff mpeg7cdvs::CompressedFeatureList::readFromFile ( char \* *filename* )

Read an entire feature list from the given file.

## Parameters

<i>filename</i>	the pathname of the file containing the feature list.
-----------------	---

## Returns

the number of bytes that have been read from the file.

### 7.27.3.10 void mpeg7cdvs::CompressedFeatureList::setFilename ( const char \* *filename* )

Set the name of the image (to be stored for subsequent retrieval).

## Parameters

<i>filename</i>	pathname of the image.
-----------------	------------------------

### 7.27.3.11 void mpeg7cdvs::CompressedFeatureList::swap ( [CompressedFeatureList](#) & *other* )

Swap this instance with another.

## Parameters

<i>other</i>	the other instance to swap with
--------------	---------------------------------

### 7.27.3.12 std::streamoff mpeg7cdvs::CompressedFeatureList::write ( std::ostream & *sout* ) const

Write an entire feature list into the given output stream.

## Parameters

<i>sout</i>	the output stream.
-------------	--------------------

## Returns

the number of bytes that have been written from the input stream.

7.27.3.13 std::streamoff mpeg7cdvs::CompressedFeatureList::writeToFile ( char \* *filename* ) const

Write an entire feature list into the given file.

## Parameters

<i>filename</i>	the pathname of the file where to store the feature list.
-----------------	---

## Returns

the number of bytes that have been written into the file.

## 7.27.4 Field Documentation

## 7.27.4.1 unsigned char\* mpeg7cdvs::CompressedFeatureList::features

all compressed features

## 7.27.4.2 std::string mpeg7cdvs::CompressedFeatureList::imagefile

pathname of the image file.

## 7.27.4.3 int mpeg7cdvs::CompressedFeatureList::imageHeight

the (possibly scaled) image height.

## 7.27.4.4 int mpeg7cdvs::CompressedFeatureList::imageWidth

the (possibly scaled) image width.

## 7.27.4.5 int mpeg7cdvs::CompressedFeatureList::nDescLength [protected]

descriptor length in bytes.

Referenced by descrBytes().

## 7.27.4.6 int mpeg7cdvs::CompressedFeatureList::numFeatures [protected]

number of features of this image

Referenced by nFeatures().

## 7.27.4.7 int mpeg7cdvs::CompressedFeatureList::originalHeight

the original image height.

#### 7.27.4.8 int mpeg7cdvs::CompressedFeatureList::originalWidth

the original image width.

#### 7.27.4.9 unsigned short\* mpeg7cdvs::CompressedFeatureList::Xcoord

the Y coordinate of the ALP keypoint

#### 7.27.4.10 unsigned short\* mpeg7cdvs::CompressedFeatureList::Ycoord

the X coordinate of the ALP keypoint

The documentation for this class was generated from the following file:

- [FeatureList.h](#)

## 7.28 mpeg7cdvs::CsscCoordinateCoding Class Reference

Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.

```
#include <CsscCoordinateCoding.h>
```

### Data Structures

- struct [CircularSumContext](#)

*Basic structure for Cssc.*

### Public Member Functions

- [CsscCoordinateCoding](#) (const [Parameters](#) &param)  
*Constructor using the given parameters to set the [CsscCoordinateCoding](#) behaviour.*
- virtual [~CsscCoordinateCoding](#) ()
- void [toBinary](#) ([BitOutputStream](#) &writer)  
*Convert the stored information into a binary stream.*
- void [fromBinary](#) ([BitInputStream](#) &reader)  
*Convert a binary stream into the stored information.*
- int [compare](#) (const [CsscCoordinateCoding](#) &other)  
*Compare this instance with another one.*
- void [exportVars](#) (unsigned int &histogramCountSize, unsigned int &histogramMapSizeX, unsigned int &histogramMapSizeY) const  
*Export the value of the histogram count and size.*
- void [generateHistogramMap](#) ([FeatureList](#) &featurelist, int numPoints)  
*Generation of new matrix representation based on circular scanning.*
- void [generateFeatureList](#) ([FeatureList](#) &descriptors)  
*Reconstruction of the original histogram map starting from circular scanning representation.*
- void [StartTrainingMode](#) ()
- void [EndTrainingMode](#) ()
- int [AddImageSample](#) ([FeatureList](#) &featurelist)
- int [writeSeparateContext](#) (char \*filename)
- int [readSeparateContext](#) (char \*filename)

## Static Public Member Functions

- static int [readSeparateContext](#) (char \*filename, [CircularSumContext](#) &cCsc)

## Static Public Attributes

- static const int [SUM\\_HIST\\_COUNT\\_SIZE](#) = 64  
*Histogram count context lenght.*
- static const int [CONTEXT\\_RANGE](#) = 5  
*Sum-based context range.*
- static const int [MAXIMUM\\_SUM\\_CONTEXT](#) = (2\*[CONTEXT\\_RANGE](#)\*[CONTEXT\\_RANGE](#) + [CONTEXT\\_RANGE](#))  
*Maximum value of sum context.*

### 7.28.1 Detailed Description

Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.

Date

2012

### 7.28.2 Constructor & Destructor Documentation

#### 7.28.2.1 mpeg7cdvs::CsscCoordinateCoding::CsscCoordinateCoding ( const Parameters & param )

Constructor using the given parameters to set the [CsscCoordinateCoding](#) behaviour.

Parameters

<i>param</i>	the set of parameters to initialize this object.
--------------	--

#### 7.28.2.2 virtual mpeg7cdvs::CsscCoordinateCoding::~CsscCoordinateCoding ( ) [virtual]

### 7.28.3 Member Function Documentation

#### 7.28.3.1 int mpeg7cdvs::CsscCoordinateCoding::AddImageSample ( FeatureList & featurelist )

#### 7.28.3.2 int mpeg7cdvs::CsscCoordinateCoding::compare ( const CsscCoordinateCoding & other )

Compare this instance with another one.

Parameters

<i>other</i>	the other instance.
--------------	---------------------

Returns

0 if equal, or the number of different values if different.

7.28.3.3 void mpeg7cdvs::CsscCoordinateCoding::EndTrainingMode ( )

7.28.3.4 void mpeg7cdvs::CsscCoordinateCoding::exportVars ( unsigned int & *histogramCountSize*, unsigned int & *histogramMapSizeX*, unsigned int & *histogramMapSizeY* ) const

Export the value of the histogram count and size.



## Parameters

<i>histogramCount-Size</i>	size of histogram count
<i>histogramMap-SizeX</i>	size of histogram map (X)
<i>histogramMap-SizeY</i>	size of histogram map (Y)

## 7.28.3.5 void mpeg7cdvs::CsscCoordinateCoding::fromBinary ( BitInputStream &amp; reader )

Convert a binary stream into the stored information.

## Parameters

<i>reader</i>	the bitstream reader object.
---------------	------------------------------

## 7.28.3.6 void mpeg7cdvs::CsscCoordinateCoding::generateFeatureList ( FeatureList &amp; descriptors )

Reconstruction of the original histogram map starting from circular scanning representation.

## Parameters

<i>descriptors</i>	reconstructed list of keypoints and descriptors
--------------------	---

## 7.28.3.7 void mpeg7cdvs::CsscCoordinateCoding::generateHistogramMap ( FeatureList &amp; featurelist, int numPoints )

Generation of new matrix representation based on circular scanning.

## Parameters

<i>featurelist</i>	list of keypoints.
<i>numPoints</i>	the number of features to encode (only the first numPoint features in featurelist will be encoded)

## 7.28.3.8 static int mpeg7cdvs::CsscCoordinateCoding::readSeparateContext ( char \* filename, CircularSumContext &amp; cCsc ) [static]

## 7.28.3.9 int mpeg7cdvs::CsscCoordinateCoding::readSeparateContext ( char \* filename )

## 7.28.3.10 void mpeg7cdvs::CsscCoordinateCoding::StartTrainingMode ( )

## 7.28.3.11 void mpeg7cdvs::CsscCoordinateCoding::toBinary ( BitOutputStream &amp; writer )

Convert the stored information into a binary stream.

## Parameters

<i>writer</i>	the bitstream writer object.
---------------	------------------------------

## 7.28.3.12 int mpeg7cdvs::CsscCoordinateCoding::writeSeparateContext ( char \* filename )

## 7.28.4 Field Documentation

7.28.4.1 `const int mpeg7cdvs::CsscCoordinateCoding::CONTEXT_RANGE = 5` `[static]`

Sum-based context range.

7.28.4.2 `const int mpeg7cdvs::CsscCoordinateCoding::MAXIMUM_SUM_CONTEXT = (2*CONTEXT_RANGE*CONTEXT_RANGE + CONTEXT_RANGE)` `[static]`

Maximum value of sum context.

7.28.4.3 `const int mpeg7cdvs::CsscCoordinateCoding::SUM_HIST_COUNT_SIZE = 64` `[static]`

Histogram count context lenght.

The documentation for this class was generated from the following file:

- [CsscCoordinateCoding.h](#)

## 7.29 mpeg7cdvs::Database Class Reference

The image database implementation containing helper methods for image retrieval.

```
#include <Database.h>
```

### Public Member Functions

- [Database](#) ()
- unsigned int [getMode](#) () const  
*get the mode shared by all descriptors in the DB.*
- size\_t [addImage](#) (const [FeatureList](#) &featList, const char \*filename)  
*Add an image to the database.*
- size\_t [replaceImage](#) (size\_t index, const [FeatureList](#) &features, const char \*filename)  
*Replace an image to the database.*
- void [merge](#) (const [Database](#) &otherDB)  
*Merge two databases.*
- int [matchCompressedDescriptors\\_oneWay](#) ([PointPairs](#) &pairs, const [CompressedFeatureList](#) &query, int imageDBIndex, float ratioThreshold) const  
*Euclidean match of a query against an image contained in the DB, with index imageDBIndex in a one way fashion.*
- int [matchCompressedDescriptors\\_twoWay](#) ([PointPairs](#) &pairs, const [CompressedFeatureList](#) &query, int imageDBIndex, float ratioThreshold) const  
*Euclidean match of a query against an image contained in the DB, with index imageDBIndex in a two way fashion.*
- std::streamoff [readFromFile](#) (const char \*filename)  
*Read an entire database from the given file.*
- std::streamoff [read](#) (std::istream &sin)  
*Read an entire database from the given input stream.*
- std::streamoff [readHeader](#) (const char \*filename)  
*Read only the modeld and the hardwareMode information from a database.*
- std::streamoff [writeToFile](#) (const char \*filename) const  
*Write an entire database into the given file.*
- std::streamoff [write](#) (std::ostream &sout) const  
*Write an entire database into the given output stream.*
- void [copyImageName](#) (char \*output, unsigned int index, size\_t maxlen) const

- Copy the i-th image name into the given output.*
- const std::string & [getImageName](#) (unsigned int index) const  
*Get the name of the image at position i in the database.*
- size\_t [find](#) (const char \*filename) const  
*Find the index of the given image in the database.*
- size\_t [size](#) () const  
*Get the size of the DB (number of images in the index).*
- bool [hasRecallGraph](#) () const  
*Check if a recall graph for this DB exists.*
- const [recallGraphNode\\_t](#) & [getRecallGraph](#) (unsigned int index) const  
*Get the recall graph node of a specific image.*
- [~Database](#) ()
- void [clear](#) ()  
*free allocated resources.*

## Data Fields

- std::vector  
< [CompressedFeatureList](#) > [images](#)  
*vector containing the features of all images in the database.*
- [recallGraph\\_t](#) [recallGraph](#)  
*a graph of db images that have relationships with other db images.*
- unsigned int [modelId](#)  
*modelId used to build the database.*

### 7.29.1 Detailed Description

The image database implementation containing helper methods for image retrieval.

#### Author

Gianluca Francini

#### Date

2011

### 7.29.2 Constructor & Destructor Documentation

7.29.2.1 `mpeg7cdvs::Database::Database ( )`

7.29.2.2 `mpeg7cdvs::Database::~~Database ( )`

### 7.29.3 Member Function Documentation

7.29.3.1 `size_t mpeg7cdvs::Database::addImage ( const FeatureList & featList, const char * filename )`

Add an image to the database.

## Parameters

<i>featList</i>	features of the image being added.
<i>filename</i>	pathname of the image being added.

## Returns

the position (index) of the image in the database.

**7.29.3.2 void mpeg7cdvs::Database::clear ( )**

free allocated resources.

**7.29.3.3 void mpeg7cdvs::Database::copyImageName ( char \* *output*, unsigned int *index*, size\_t *maxlen* ) const**

Copy the i-th image name into the given output.

## Parameters

<i>output</i>	the output buffer
<i>index</i>	index of the image in the database
<i>maxlen</i>	maximum length of the name (if longer, the name will be clipped).

**7.29.3.4 size\_t mpeg7cdvs::Database::find ( const char \* *filename* ) const**

Find the index of the given image in the database.

## Parameters

<i>filename</i>	the name of the image
-----------------	-----------------------

## Returns

the index of the image in the DB, or -1 if not found.

**7.29.3.5 const std::string& mpeg7cdvs::Database::getImageName ( unsigned int *index* ) const [inline]**

Get the name of the image at position i in the database.

## Parameters

<i>index</i>	index of the image in the database
--------------	------------------------------------

## Returns

the image name

References images.

**7.29.3.6 unsigned int mpeg7cdvs::Database::getMode ( ) const [inline]**

get the mode shared by all descriptors in the DB.

## Returns

the mode id.

References modeld.

7.29.3.7 `const recallGraphNode_t& mpeg7cdvs::Database::getRecallGraph ( unsigned int index ) const` `[inline]`

Get the recall graph node of a specific image.

Parameters

<i>index</i>	the index of the image in the database.
--------------	---

Returns

the recall graph node of the i-th image.

References recallGraph.

7.29.3.8 `bool mpeg7cdvs::Database::hasRecallGraph ( ) const` `[inline]`

Check if a recall graph for this DB exists.

The recall graph indicates the relationships among images in the DB.

Returns

true if the recall graph is present.

References recallGraph.

7.29.3.9 `int mpeg7cdvs::Database::matchCompressedDescriptors_oneWay ( PointPairs & pairs, const CompressedFeatureList & query, int imageDBIndex, float ratioThreshold ) const`

Euclidean match of a query against an image contained in the DB, with index *imageDBIndex* in a one way fashion.

The coordinates of the matched points are stored in the [PointPairs](#) container class.

Parameters

<i>pairs</i>	computed matching pairs of points
<i>query</i>	features of the query image.
<i>imageDBIndex</i>	index of the image contained in the database that will be compared to the query.
<i>ratioThreshold</i>	the threshold used in the ratio test.

Returns

number of matched points.

7.29.3.10 `int mpeg7cdvs::Database::matchCompressedDescriptors_twoWay ( PointPairs & pairs, const CompressedFeatureList & query, int imageDBIndex, float ratioThreshold ) const`

Euclidean match of a query against an image contained in the DB, with index *imageDBIndex* in a two way fashion.

The coordinates of the matched points are stored in the [PointPairs](#) container class.

Parameters

<i>pairs</i>	computed matching pairs of points
<i>query</i>	features of the query image.

<i>imageDBindex</i>	index of the image contained in the database that will be compared to the query.
<i>ratioThreshold</i>	the threshold used in the ratio test.

**Returns**

number of matched points.

**7.29.3.11 void mpeg7cdvs::Database::merge ( const Database & otherDB )**

Merge two databases.

The databases must have an index with the same number of codewords (if this is not empty).

**Parameters**

<i>otherDB</i>	the other db to merge to the current one.
----------------	---

**7.29.3.12 std::streamoff mpeg7cdvs::Database::read ( std::istream & sin )**

Read an entire database from the given input stream.

**Parameters**

<i>sin</i>	the input stream.
------------	-------------------

**Returns**

the number of bytes that have been read from the input stream.

**7.29.3.13 std::streamoff mpeg7cdvs::Database::readFromFile ( const char \* filename )**

Read an entire database from the given file.

**Parameters**

<i>filename</i>	the pathname of the file containing the database.
-----------------	---

**Returns**

the number of bytes that have been read from the file.

**7.29.3.14 std::streamoff mpeg7cdvs::Database::readHeader ( const char \* filename )**

Read only the modeld and the hardwareMode information from a database.

**Parameters**

<i>filename</i>	the pathname of the file containing the database.
-----------------	---

**Returns**

the number of bytes that have been read from the input stream.

**7.29.3.15 size\_t mpeg7cdvs::Database::replacelImage ( size\_t index, const FeatureList & features, const char \* filename )**

Replace an image to the database.

## Parameters

<i>index</i>	index of the image to replace.
<i>features</i>	features of the new image.
<i>filename</i>	pathname of the new image.

## Returns

the position (index) of the replaced image in the database.

7.29.3.16 `size_t mpeg7cdvs::Database::size ( ) const [inline]`

Get the size of the DB (number of images in the index).

## Returns

the size of the DB

References images.

7.29.3.17 `std::streamoff mpeg7cdvs::Database::write ( std::ostream & sout ) const`

Write an entire database into the given output stream.

## Parameters

<i>sout</i>	the output stream.
-------------	--------------------

## Returns

the number of bytes that have been written from the input stream.

7.29.3.18 `std::streamoff mpeg7cdvs::Database::writeToFile ( const char * filename ) const`

Write an entire database into the given file.

## Parameters

<i>filename</i>	the pathname of the file where to store the database.
-----------------	---

## Returns

the number of bytes that have been written into the file.

## 7.29.4 Field Documentation

7.29.4.1 `std::vector<CompressedFeatureList> mpeg7cdvs::Database::images`

vector containing the features of all images in the database.

Referenced by getImageName(), and size().

7.29.4.2 `unsigned int mpeg7cdvs::Database::modelId`

modelId used to build the database.

Referenced by getMode().

#### 7.29.4.3 recallGraph\_t mpeg7cdvs::Database::recallGraph

a graph of db images that have relationships with other db images.

Referenced by getRecallGraph(), and hasRecallGraph().

The documentation for this class was generated from the following file:

- [Database.h](#)

## 7.30 DistratEigen Class Reference

Simplified and faster version of DISTRAT, based on the Eigen library.

```
#include <DistratEigen.h>
```

### Public Member Functions

- virtual [~DistratEigen](#) ()
- [DistratEigen](#) (const float \*x1, const float \*x2, const float \*y1, const float \*y2, int size)  
*Parametric constructor.*
- int [estimateInliers](#) (bool useParametric=false, bool computeInliers=true, unsigned int percentile=99, int \*inlierIndexes=NULL)  
*Function computing the estimation of the number of inliers (DISTRAT core).*

### Data Fields

- bool [m\\_bFitsGood](#)  
*result of Goodness of Fit*
- float [m\\_c](#)  
*output produced by Goodness of Fit*
- float [m\\_GoFThreshold](#)  
*Goodness of Fit threshold.*

#### 7.30.1 Detailed Description

Simplified and faster version of DISTRAT, based on the Eigen library.

The DISTRAT algorithm performs a geometric consistency check with better performances than RANSAC. This DISTRAT implementation is based on the Eigen C++ library and does not depend on any other class/lib.

#### Author

Massimo Balestri

#### Date

13 Jan 2012

#### 7.30.2 Constructor & Destructor Documentation

7.30.2.1 virtual DistratEigen::~~DistratEigen ( ) [virtual]

7.30.2.2 DistratEigen::DistratEigen ( const float \* x1, const float \* x2, const float \* y1, const float \* y2, int size )

Parametric constructor.



## Parameters

<i>x1</i>	a vector containing the first coordinate of all the points belonging to the first image
<i>x2</i>	contains the second coordinate on the first image
<i>y1</i>	contains the first coordinate on the second image
<i>y2</i>	contains the second coordinate on the second image
<i>size</i>	the number of elements of all (x1, x2, y1, y2) vectors

## 7.30.3 Member Function Documentation

7.30.3.1 `int DistratEigen::estimateInliers ( bool useParametric = false, bool computeInliers = true, unsigned int percentile = 99, int * inlierIndexes = NULL )`

Function computing the estimation of the number of inliers (DISTRAT core).

## Parameters

<i>useParametric</i>	if true the parametric version of Distrat is used, instead of the non-parametric one
<i>computeInliers</i>	if true the index of inliers is produced
<i>percentile</i>	acceptable values: 99, 98, 97, 96, 95
<i>inlierIndexes</i>	the output indexes of inlier points (if the pointer is NULL these values are not provided).

## Returns

the number of inliers

## 7.30.4 Field Documentation

7.30.4.1 `bool DistratEigen::m_bFitsGood`

result of Goodness of Fit

7.30.4.2 `float DistratEigen::m_c`

output produced by Goodness of Fit

7.30.4.3 `float DistratEigen::m_GoFThreshold`

Goodness of Fit threshold.

The documentation for this class was generated from the following file:

- [DistratEigen.h](#)

## 7.31 mpeg7cdvs::Feature Class Reference

Container class for the features of a single point (storing coordinates, scale, orientation, peak and descriptor of a point).

```
#include <Feature.h>
```

## Public Member Functions

- [Feature](#) (void)

- void `toFile` (FILE \*file) const  
*Write the feature into a file.*
- void `fromFile` (FILE \*file)  
*Write the feature into a file.*

## Data Fields

- float `x`  
*the X coordinate of the ALP keypoint*
- float `y`  
*the Y coordinate of the ALP keypoint*
- float `scale`  
*the scale of the ALP keypoint*
- float `orientation`  
*the orientation of the ALP keypoint*
- float `peak`  
*the peak of the ALP keypoint*
- float `curvRatio`  
*the ratio of the curvatures*
- float `curvSigma`  
*the curvature at sigma*
- float `descr` [`descrLength`]  
*the SIFT descriptor of the ALP keypoint*
- float `pdf`  
*probability of this point to be matched*
- int `spatialIndex`  
*indicates the order of transmission of this point*
- unsigned short `relevance`  
*relevance of the keypoint, computed on the basis of his characteristics*
- int `qdescr` [`descrLength`]  
*the quantized (ternarized) descriptor values.*
- int `octave`  
*octave of this feature*
- int `iscale`  
*int scale*

## Static Public Attributes

- static const unsigned int `descrLength` = 128  
*the size of a feature (key point)*

### 7.31.1 Detailed Description

Container class for the features of a single point (storing coordinates, scale, orientation, peak and descriptor of a point).

#### Author

Gianluca Francini

#### Date

2011

## 7.31.2 Constructor & Destructor Documentation

### 7.31.2.1 mpeg7cdvs::Feature::Feature ( void )

## 7.31.3 Member Function Documentation

### 7.31.3.1 void mpeg7cdvs::Feature::fromFile ( FILE \* *file* )

Write the feature into a file.

Parameters

<i>file</i>	the output file.
-------------	------------------

### 7.31.3.2 void mpeg7cdvs::Feature::toFile ( FILE \* *file* ) const

Write the feature into a file.

Parameters

<i>file</i>	the output file.
-------------	------------------

## 7.31.4 Field Documentation

### 7.31.4.1 float mpeg7cdvs::Feature::curvRatio

the ratio of the curvatures

### 7.31.4.2 float mpeg7cdvs::Feature::curvSigma

the curvature at sigma

### 7.31.4.3 float mpeg7cdvs::Feature::descr[descrLength]

the SIFT descriptor of the ALP keypoint

### 7.31.4.4 const unsigned int mpeg7cdvs::Feature::descrLength = 128 [static]

the size of a feature (key point)

### 7.31.4.5 int mpeg7cdvs::Feature::iscale

int scale

### 7.31.4.6 int mpeg7cdvs::Feature::octave

octave of this feature

### 7.31.4.7 float mpeg7cdvs::Feature::orientation

the orientation of the ALP keypoint

#### 7.31.4.8 float mpeg7cdvs::Feature::pdf

probability of this point to be matched

#### 7.31.4.9 float mpeg7cdvs::Feature::peak

the peak of the ALP keypoint

#### 7.31.4.10 int mpeg7cdvs::Feature::qdescr[descrLength]

the quantized (ternarized) descriptor values.

#### 7.31.4.11 unsigned short mpeg7cdvs::Feature::relevance

relevance of the keypoint, computed on the basis of his characteristics

#### 7.31.4.12 float mpeg7cdvs::Feature::scale

the scale of the ALP keypoint

#### 7.31.4.13 int mpeg7cdvs::Feature::spatialIndex

indicates the order of transmission of this point

#### 7.31.4.14 float mpeg7cdvs::Feature::x

the X coordinate of the ALP keypoint

#### 7.31.4.15 float mpeg7cdvs::Feature::y

the Y coordinate of the ALP keypoint

The documentation for this class was generated from the following file:

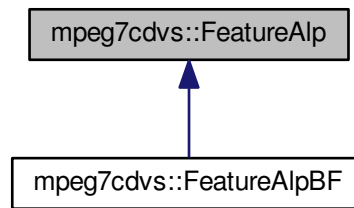
- [Feature.h](#)

## 7.32 mpeg7cdvs::FeatureAlp Class Reference

Definition of keypoint for Alp.

```
#include <AlpOctave.h>
```

Inheritance diagram for mpeg7cdvs::FeatureAlp:



### Public Member Functions

- float [distL1](#) (const [FeatureAlp](#) &other) const  
*Compute distance from another [FeatureAlp](#);*

### Data Fields

- float [x](#)  
*the X coordinate of the ALP keypoint*
- float [y](#)  
*the Y coordinate of the ALP keypoint*
- float [sigma](#)  
*the sigma of the Gaussian filter used to detect this point*
- float [orientation](#)  
*the orientation of the ALP keypoint*
- float [peak](#)  
*the peak of the ALP keypoint*
- float [curvRatio](#)  
*the ratio of the curvatures*
- float [curvSigma](#)  
*the curvature at sigma*
- float [pdf](#)  
*probability of this point to be matched*
- int [spatialIndex](#)  
*indicates the order of transmission of this point*
- int [octave](#)  
*octave of this feature*
- int [iscale](#)  
*int scale*
- int [ix](#)  
*original X coordinate*
- int [iy](#)  
*original Y coordinate*

## Friends

- `std::ostream & operator<< (std::ostream &outstr, const FeatureAlp &f)`  
*Print the [Feature](#) to cout.*

### 7.32.1 Detailed Description

Definition of keypoint for Alp.

This is smaller than [Feature](#) because it does not include the SIFT descriptor.

### 7.32.2 Member Function Documentation

#### 7.32.2.1 `float mpeg7cdvs::FeatureAlp::distL1 ( const FeatureAlp & other ) const`

Compute distance from another [FeatureAlp](#);

Parameters

<i>other</i>	the other feature
--------------	-------------------

Returns

the L1 distance

### 7.32.3 Friends And Related Function Documentation

#### 7.32.3.1 `std::ostream& operator<< ( std::ostream & outstr, const FeatureAlp & f )` [[friend](#)]

Print the [Feature](#) to cout.

### 7.32.4 Field Documentation

#### 7.32.4.1 `float mpeg7cdvs::FeatureAlp::curvRatio`

the ratio of the curvatures

#### 7.32.4.2 `float mpeg7cdvs::FeatureAlp::curvSigma`

the curvature at sigma

#### 7.32.4.3 `int mpeg7cdvs::FeatureAlp::iscale`

int scale

#### 7.32.4.4 `int mpeg7cdvs::FeatureAlp::ix`

original X coordinate

#### 7.32.4.5 `int mpeg7cdvs::FeatureAlp::iy`

original Y coordinate

#### 7.32.4.6 int mpeg7cdvs::FeatureAlp::octave

octave of this feature

#### 7.32.4.7 float mpeg7cdvs::FeatureAlp::orientation

the orientation of the ALP keypoint

#### 7.32.4.8 float mpeg7cdvs::FeatureAlp::pdf

probability of this point to be matched

#### 7.32.4.9 float mpeg7cdvs::FeatureAlp::peak

the peak of the ALP keypoint

#### 7.32.4.10 float mpeg7cdvs::FeatureAlp::sigma

the sigma of the Gaussian filter used to detect this point

#### 7.32.4.11 int mpeg7cdvs::FeatureAlp::spatialIndex

indicates the order of transmission of this point

#### 7.32.4.12 float mpeg7cdvs::FeatureAlp::x

the X coordinate of the ALP keypoint

#### 7.32.4.13 float mpeg7cdvs::FeatureAlp::y

the Y coordinate of the ALP keypoint

The documentation for this class was generated from the following file:

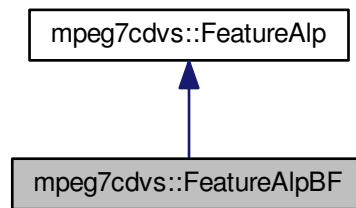
- [AlpOctave.h](#)

## 7.33 mpeg7cdvs::FeatureAlpBF Class Reference

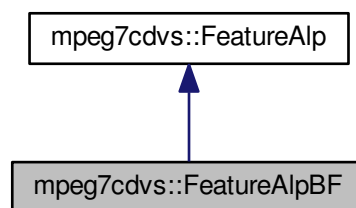
Inherits all member variables declared in [FeatureAlp](#) and adds two new member variables.

```
#include <AlpOctaveBF.h>
```

Inheritance diagram for `mpeg7cdvs::FeatureAlpBF`:



Collaboration diagram for `mpeg7cdvs::FeatureAlpBF`:



## Data Fields

- char `mode`  
*point mode*
- char `block_id`  
*detected in the block id*

## Additional Inherited Members

### 7.33.1 Detailed Description

Inherits all member variables declared in `FeatureAlp` and adds two new member variables.

### 7.33.2 Field Documentation

#### 7.33.2.1 char `mpeg7cdvs::FeatureAlpBF::block_id`

detected in the block id



## 7.33.2.2 char mpeg7cdvs::FeatureAlpBF::mode

point mode

The documentation for this class was generated from the following file:

- [AlpOctaveBF.h](#)

## 7.34 mpeg7cdvs::FeatureList Class Reference

Container class for all features of an image.

```
#include <FeatureList.h>
```

### Public Member Functions

- [FeatureList](#) ()
- void [clear](#) ()  
*Clear all memory.*
- void [setResolution](#) (int imgWidth, int imgHeight, int [originalWidth](#), int [originalHeight](#))  
*Store the resolution of the image from which the SIFT points were extracted.*
- int [nFeatures](#) () const  
*Get the number of features of the image.*
- void [addFeature](#) (const [Feature](#) &f)  
*Add a feature to the current list of features of the image.*
- void [sortSpatialIndex](#) ()  
*Sort the list of features on the basis of the spatial position of the points, used in the compression of coordinates.*
- void [sortRelevance](#) ()  
*Sort the list of features on the basis of the relevance.*
- int [compareDescriptors](#) (const [FeatureList](#) &otherList, bool compressed=false) const  
*Compare the descriptor contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.*
- int [compareCoordinates](#) (const [FeatureList](#) &otherList, bool compressed=false, int blockWidth=1) const  
*Compare the coordinates contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.*
- int [compareKeypoints](#) (const [FeatureList](#) &otherList) const  
*Compare the key points properties contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.*
- void [toFile](#) (FILE \*file) const  
*Write the entire [FeatureList](#) into a file.*
- void [toFile](#) (const char \*filename) const  
*Write the entire [FeatureList](#) into a file.*
- void [fromFile](#) (FILE \*file)  
*Read the entire [FeatureList](#) from a file.*
- void [fromFile](#) (const char \*filename)  
*Read the entire [FeatureList](#) from a file.*
- void [select](#) (const std::vector< int > &indices)  
*Select a subset of features on the basis of the given indices; all other elements are discarded.*
- void [selectFromTo](#) (int startInd, int endInd)  
*Select a subset of features on the basis of the given range; all other elements are discarded.*
- void [selectFirst](#) (int n)  
*Select the first n features; all other elements are discarded.*

- void [compress](#) (int numberOfElementGroups)  
*Performs the compression of the SIFT descriptor.*
- void [toBinary](#) ([BitOutputStream](#) &writer, bool writeRelevance, int numFeatures)  
*Serialize [FeatureList](#) into a stream of bits.*
- void [fromBinary](#) ([BitInputStream](#) &reader, bool readRelevance)  
*De-serialize [FeatureList](#) from a stream of bits.*
- int [computeMaxPoints](#) (const [Parameters](#) &params, int targetBits)  
*Computes the maximum number of points to be added to the descriptor for a given bitrate.*
- void [setRelevantPoints](#) (int num)  
*Set the first n points as more relevant.*
- int [getRelevantPoints](#) () const  
*Get the number of relevant points.*
- void [print](#) () const  
*Print a summary of the featurelist content.*

## Data Fields

- unsigned int [qdescr\\_size](#)  
*The number of quantized elements in the key-point features (qdescr)*
- int [imageHeight](#)  
*the (possibly scaled) image height.*
- int [imageWidth](#)  
*the (possibly scaled) image width.*
- int [originalHeight](#)  
*the original image height.*
- int [originalWidth](#)  
*the original image width.*
- std::vector< [Feature](#) > [features](#)  
*the vector of features extracted from the image.*

## Static Public Attributes

- static const int [MAX\\_NUM\\_FEATURES](#) = 65536  
*theoretical limit set by the CDVS syntax*

## Friends

- class [CompressedFeatureList](#)

### 7.34.1 Detailed Description

Container class for all features of an image.

#### Author

Gianluca Francini

#### Date

2011

## 7.34.2 Constructor & Destructor Documentation

### 7.34.2.1 mpeg7cdvs::FeatureList::FeatureList ( )

## 7.34.3 Member Function Documentation

### 7.34.3.1 void mpeg7cdvs::FeatureList::addFeature ( const Feature & f )

Add a feature to the current list of features of the image.

Parameters

<i>f</i>	the feature to be added.
----------	--------------------------

### 7.34.3.2 void mpeg7cdvs::FeatureList::clear ( )

Clear all memory.

### 7.34.3.3 int mpeg7cdvs::FeatureList::compareCoordinates ( const FeatureList & otherList, bool compressed = false, int blockWidth = 1 ) const

Compare the coordinates contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.

This is mainly used for debugging.

Parameters

<i>otherList</i>	the other list to compare.
<i>compressed</i>	indicates if both descriptors are compressed
<i>blockWidth</i>	if compressed, indicates the quantization block width (in pixels)

Returns

the number of different values in the two descriptors.

### 7.34.3.4 int mpeg7cdvs::FeatureList::compareDescriptors ( const FeatureList & otherList, bool compressed = false ) const

Compare the descriptor contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.

This is mainly used for debugging.

Parameters

<i>otherList</i>	the other list to compare.
<i>compressed</i>	indicates if both descriptors are compressed

Returns

the number of different values in the two descriptors.

### 7.34.3.5 int mpeg7cdvs::FeatureList::compareKeypoints ( const FeatureList & otherList ) const

Compare the key points properties contained in this [FeatureList](#) with the one contained in otherList, and return the number of different values.

This is mainly used for debugging.

## Parameters

<i>otherList</i>	the other list to compare.
------------------	----------------------------

## Returns

the number of different values in the two lists.

7.34.3.6 void mpeg7cdvs::FeatureList::compress ( int *numberOfElementGroups* )

Performs the compression of the SIFT descriptor.

## Parameters

<i>numberOfElementGroups</i>	the number of element groups of this descriptor
------------------------------	---

7.34.3.7 int mpeg7cdvs::FeatureList::computeMaxPoints ( const Parameters & *params*, int *targetBits* )

Computes the maximum number of points to be added to the descriptor for a given bitrate.

This method does not assume any pre-computed statistics, just try to encode the features and discover how many bits are used.

## Parameters

<i>params</i>	the current running parameters
<i>targetBits</i>	the target number of bits to fill

## Returns

the number of points

7.34.3.8 void mpeg7cdvs::FeatureList::fromBinary ( BitInputStream & *reader*, bool *readRelevance* )

De-serialize [FeatureList](#) from a stream of bits.

## Parameters

<i>reader</i>	the bitstream reader object.
<i>readRelevance</i>	read also the relevance value, used for the higher querylengths

7.34.3.9 void mpeg7cdvs::FeatureList::fromFile ( FILE \* *file* )

Read the entire [FeatureList](#) from a file.

## Parameters

<i>file</i>	the input file.
-------------	-----------------

7.34.3.10 void mpeg7cdvs::FeatureList::fromFile ( const char \* *filename* )

Read the entire [FeatureList](#) from a file.

## Parameters

<i>filename</i>	the input filename.
-----------------	---------------------

7.34.3.11 `int mpeg7cdvs::FeatureList::getRelevantPoints ( ) const`

Get the number of relevant points.

## Returns

the number of relevant points (generally smaller than the total number of key points).

7.34.3.12 `int mpeg7cdvs::FeatureList::nFeatures ( ) const`

Get the number of features of the image.

## Returns

the number of features currently stored in the features vector.

7.34.3.13 `void mpeg7cdvs::FeatureList::print ( ) const`

Print a summary of the featurelist content.

7.34.3.14 `void mpeg7cdvs::FeatureList::select ( const std::vector< int > & indices )`

Select a subset of features on the basis of the given indices; all other elements are discarded.

## Parameters

<i>indices</i>	indices of elements to keep.
----------------	------------------------------

7.34.3.15 `void mpeg7cdvs::FeatureList::selectFirst ( int n )`

Select the first n features; all other elements are discarded.

## Parameters

<i>n</i>	the number of elements to keep.
----------	---------------------------------

7.34.3.16 `void mpeg7cdvs::FeatureList::selectFromTo ( int startInd, int endInd )`

Select a subset of features on the basis of the given range; all other elements are discarded.

## Parameters

<i>startInd</i>	first elements to keep.
<i>endInd</i>	last elements to keep.

7.34.3.17 `void mpeg7cdvs::FeatureList::setRelevantPoints ( int num )`

Set the first n points as more relevant.

## Parameters

<i>num</i>	the number of releval points.
------------	-------------------------------

7.34.3.18 void mpeg7cdvs::FeatureList::setResolution ( int *imgWidth*, int *imgHeight*, int *originalWidth*, int *originalHeight* )

Store the resolution of the image from which the SIFT points were extracted.

## Parameters

<i>imgWidth</i>	the (possibly scaled) image width.
<i>imgHeight</i>	the (possibly scaled) image height.
<i>originalWidth</i>	the width of the original image.
<i>originalHeight</i>	the height the original image.

7.34.3.19 void mpeg7cdvs::FeatureList::sortRelevance ( )

Sort the list of features on the basis of the relevance.

7.34.3.20 void mpeg7cdvs::FeatureList::sortSpatialIndex ( )

Sort the list of features on the basis of the spatial position of the points, used in the compression of coordinates.

7.34.3.21 void mpeg7cdvs::FeatureList::toBinary ( BitOutputStream & *writer*, bool *writeRelevance*, int *numFeatures* )

Serialize [FeatureList](#) into a stream of bits.

## Parameters

<i>writer</i>	the bitstream writer object.
<i>writeRelevance</i>	write also the relevance value, used for the higher querylengths.
<i>numFeatures</i>	the number of features to encode.

7.34.3.22 void mpeg7cdvs::FeatureList::toFile ( FILE \* *file* ) const

Write the entire [FeatureList](#) into a file.

## Parameters

<i>file</i>	the output file.
-------------	------------------

7.34.3.23 void mpeg7cdvs::FeatureList::toFile ( const char \* *filename* ) const

Write the entire [FeatureList](#) into a file.

## Parameters

<i>filename</i>	the output filename.
-----------------	----------------------

## 7.34.4 Friends And Related Function Documentation

7.34.4.1 friend class **CompressedFeatureList** [[friend](#)]

### 7.34.5 Field Documentation

#### 7.34.5.1 `std::vector<Feature> mpeg7cdvs::FeatureList::features`

the vector of features extracted from the image.

#### 7.34.5.2 `int mpeg7cdvs::FeatureList::imageHeight`

the (possibly scaled) image height.

#### 7.34.5.3 `int mpeg7cdvs::FeatureList::imageWidth`

the (possibly scaled) image width.

#### 7.34.5.4 `const int mpeg7cdvs::FeatureList::MAX_NUM_FEATURES = 65536` `[static]`

theoretical limit set by the CDVS syntax

#### 7.34.5.5 `int mpeg7cdvs::FeatureList::originalHeight`

the original image height.

#### 7.34.5.6 `int mpeg7cdvs::FeatureList::originalWidth`

the original image width.

#### 7.34.5.7 `unsigned int mpeg7cdvs::FeatureList::qdescr_size`

The number of quantized elements in the key-point features (qdescr)

The documentation for this class was generated from the following file:

- [FeatureList.h](#)

## 7.35 FileManager Class Reference

Helper class to manage lists of file names.

```
#include <FileManager.h>
```

### Public Member Functions

- [FileManager](#) ()
- virtual [~FileManager](#) ()
- void [setDatasetPath](#) (const char \*basedir)  
*Set the dataset base directory.*
- void [setAnnotationPath](#) (const char \*basedir)  
*Set the annotation base directory.*
- `size_t` [readAnnotation](#) (const char \*filename)  
*Read the list of images from the given annotation file.*
- `std::string` [getAbsolutePathname](#) (`size_t` i) const



- Get the first image name found at the i-th position in the annotation file.*
- `std::string getRelativePathname (size_t i) const`  
*Get the first image name found at the i-th position in the annotation file.*
- `std::string replaceExt (const std::string &imageName, const char *ext) const`  
*Convert a pathname into a pathname with the given extension.*
- `std::string getQueryName (size_t i) const`  
*Get the first image name found at the i-th position in the annotation file.*
- `std::string getReferenceName (size_t i) const`  
*Get the second image name found at the i-th position in the annotation file.*

### 7.35.1 Detailed Description

Helper class to manage lists of file names.

#### Author

Massimo Balestri

#### Date

2012

### 7.35.2 Constructor & Destructor Documentation

#### 7.35.2.1 FileManager::FileManager ( )

#### 7.35.2.2 virtual FileManager::~~FileManager ( ) [virtual]

### 7.35.3 Member Function Documentation

#### 7.35.3.1 std::string FileManager::getAbsolutePathname ( size\_t i ) const

Get the first image name found at the i-th position in the annotation file.

The image name is provided as an absolute pathname.

#### Parameters

<i>i</i>	the index of the image in the annotation file.
----------	--

#### Returns

the absolute pathname of the image.

#### 7.35.3.2 std::string FileManager::getQueryName ( size\_t i ) const

Get the first image name found at the i-th position in the annotation file.

The image name is provided as an absolute pathname.

#### Parameters

<i>i</i>	the index of the image in the annotation file.
----------	--

**Returns**

the absolute pathname of the image.

**7.35.3.3 `std::string FileManager::getReferenceName ( size_t i ) const`**

Get the second image name found at the i-th position in the annotation file.

The image name is provided as an absolute pathname.

**Parameters**

<i>i</i>	the index of the image in the annotation file.
----------	--

**Returns**

the absolute pathname of the image.

**7.35.3.4 `std::string FileManager::getRelativePathname ( size_t i ) const`**

Get the first image name found at the i-th position in the annotation file.

The image name is provided as a relative pathname.

**Parameters**

<i>i</i>	the index of the image in the annotation file.
----------	--

**Returns**

the relative pathname of the image.

**7.35.3.5 `size_t FileManager::readAnnotation ( const char * filename )`**

Read the list of images from the given annotation file.

**Parameters**

<i>filename</i>	the name of the annotation text file containing the list of images.
-----------------	---

**Returns**

the number of lines read from the filename.

**7.35.3.6 `std::string FileManager::replaceExt ( const std::string & imageName, const char * ext ) const`**

Convert a pathname into a pathname with the given extension.

**Parameters**

<i>imageName</i>	the original image name;
<i>ext</i>	new extension;

**Returns**

the modified pathname.

**7.35.3.7 void FileManager::setAnnotationPath ( const char \* *basedir* )**

Set the annotation base directory.

**Parameters**

<i>basedir</i>	the base directory.
----------------	---------------------

**7.35.3.8 void FileManager::setDatasetPath ( const char \* *basedir* )**

Set the dataset base directory.

**Parameters**

<i>basedir</i>	the base directory.
----------------	---------------------

The documentation for this class was generated from the following file:

- [FileManager.h](#)

## 7.36 mpeg7cdvs::Filter Class Reference

A class containing a separable Gaussian filter kernel.

```
#include <AlpOctave.h>
```

**Public Member Functions**

- [Filter](#) (double [sigma](#))  
*Create a separable Gaussian filter kernel using the given sigma.*
- void [print](#) () const  
*Print the filter data.*

**Data Fields**

- int [ntaps](#)  
*the number of filter taps*
- double [sigma](#)  
*the sigma value corresponding to this Gaussian filter*
- float [kernel](#) [[maxsize](#)]  
*the filter kernel values*

**Static Public Attributes**

- static const int [maxsize](#) = 27  
*max size of the filter kernel*

### 7.36.1 Detailed Description

A class containing a separable Gaussian filter kernel.

A class containing a separable Gaussian and LoG filter kernel in frequency domain.

The filter must be applied in the two spatial directions (horizontally and vertically) in whatever order.

### 7.36.2 Constructor & Destructor Documentation

#### 7.36.2.1 `mpeg7cdvs::Filter::Filter ( double sigma )`

Create a separable Gaussian filter kernel using the given sigma.

Parameters

<i>sigma</i>	the sigma of the Gaussian filter
--------------	----------------------------------

### 7.36.3 Member Function Documentation

#### 7.36.3.1 `void mpeg7cdvs::Filter::print ( ) const`

Print the filter data.

### 7.36.4 Field Documentation

#### 7.36.4.1 `float mpeg7cdvs::Filter::kernel[maxsize]`

the filter kernel values

#### 7.36.4.2 `const int mpeg7cdvs::Filter::maxsize = 27` `[static]`

max size of the filter kernel

#### 7.36.4.3 `int mpeg7cdvs::Filter::ntaps`

the number of filter taps

#### 7.36.4.4 `double mpeg7cdvs::Filter::sigma`

the sigma value corresponding to this Gaussian filter

The documentation for this class was generated from the following file:

- [AlpOctave.h](#)

## 7.37 `mpeg7cdvs::FrequencyFilter` Class Reference

```
#include <AlpOctaveBF.h>
```

## Public Member Functions

- [FrequencyFilter](#) ()
- virtual [~FrequencyFilter](#) ()
- void [Convolution](#) (int o\_cur, int type, int loc)

## Static Public Member Functions

- static void [open\\_lib](#) ()  
*to be called once at the beginning of main*
- static void [close\\_lib](#) ()  
*to be called once at the end of main*

## Data Fields

- float \* [inmat](#)
- fftwf\_complex \* [fblock](#)  
*current frequency block data.*
- fftwf\_complex \* [fblock\\_flog](#)  
*current frequency block log data.*
- fftwf\_plan [fftplan](#)  
*pointer to plans used for fft*
- fftwf\_plan [ifftplan](#)  
*pointer to plans used for ifft*

### 7.37.1 Constructor & Destructor Documentation

7.37.1.1 `mpeg7cdvs::FrequencyFilter::FrequencyFilter ( )`

7.37.1.2 `virtual mpeg7cdvs::FrequencyFilter::~~FrequencyFilter ( )` [virtual]

### 7.37.2 Member Function Documentation

7.37.2.1 `static void mpeg7cdvs::FrequencyFilter::close_lib ( )` [static]

to be called once at the end of main

7.37.2.2 `void mpeg7cdvs::FrequencyFilter::Convolution ( int o_cur, int type, int loc )`

7.37.2.3 `static void mpeg7cdvs::FrequencyFilter::open_lib ( )` [static]

to be called once at the beginning of main

### 7.37.3 Field Documentation

7.37.3.1 `fftwf_complex* mpeg7cdvs::FrequencyFilter::fblock`

current frequency block data.

7.37.3.2 `fftwf_complex* mpeg7cdvs::FrequencyFilter::fblock_flog`

current frequency block log data.

#### 7.37.3.3 `fftwf_plan` `mpeg7cdvs::FrequencyFilter::fftplan`

pointer to plans used for fft

#### 7.37.3.4 `fftwf_plan` `mpeg7cdvs::FrequencyFilter::ifftplan`

pointer to plans used for ifft

#### 7.37.3.5 `float*` `mpeg7cdvs::FrequencyFilter::inmat`

The documentation for this class was generated from the following file:

- [AlpOctaveBF.h](#)

## 7.38 HiResTimer Class Reference

C++ wrapper class for the C functions implementing the high resolution timer.

```
#include <HiResTimer.h>
```

### Public Member Functions

- void [start](#) ()  
*Start the timer.*
- void [stop](#) ()  
*Stop the timer.*
- double [elapsed](#) ()  
*Get the elapsed time (in seconds) from start to stop.*

### 7.38.1 Detailed Description

C++ wrapper class for the C functions implementing the high resolution timer.

Author

Massimo Balestri

Date

2012

### 7.38.2 Member Function Documentation

#### 7.38.2.1 `double` `HiResTimer::elapsed ( )` [`inline`]

Get the elapsed time (in seconds) from start to stop.

**Returns**

the elapsed time in seconds.

References `elapsed_time()`.

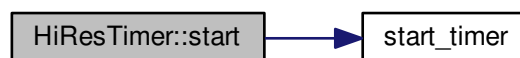
Here is the call graph for this function:

**7.38.2.2 void HiResTimer::start ( ) [inline]**

Start the timer.

References `start_timer()`.

Here is the call graph for this function:

**7.38.2.3 void HiResTimer::stop ( ) [inline]**

Stop the timer.

References `stop_timer()`.

Here is the call graph for this function:



The documentation for this class was generated from the following file:

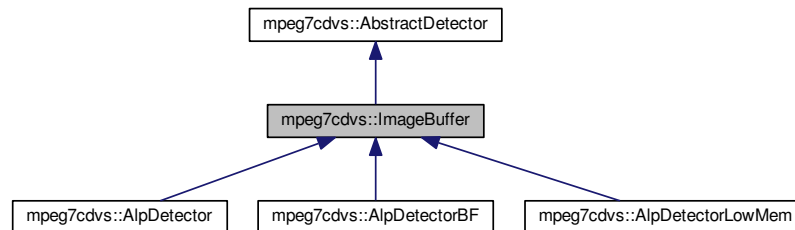
- [HiResTimer.h](#)

## 7.39 mpeg7cdvs::ImageBuffer Class Reference

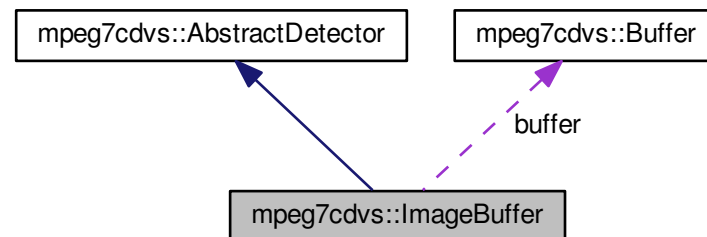
A container class for a bidimensional image; it's the base class of all keypoint detector classes.

```
#include <ImageBuffer.h>
```

Inheritance diagram for mpeg7cdvs::ImageBuffer:



Collaboration diagram for mpeg7cdvs::ImageBuffer:



### Public Member Functions

- `ImageBuffer ()`
- `virtual ~ImageBuffer ()`
- `void swap (ImageBuffer &other)`  
*swap the content of two ImageBuffer(s)*
- `void read (int width, int height, const unsigned char *buffer)`  
*Read a planar luminance image from a buffer.*
- `void resample (ImageBuffer &dest) const`  
*Convert this image into a destination image having a different resolution by filtering and sampling the original image.*
- `void resample (double rfactor)`  
*Resample this image using the given reduction factor (the original image is discarded).*
- `void resampleIfGreater (int maxSize)`  
*Resample this image if either the horizontal or the vertical dimension of the image is greater that the given maximum size.*



## Static Public Member Functions

- static void [print](#) (const std::vector< [Feature](#) > &f, const char \*source)  
*Print the given list of features.*
- static void [printHeader](#) (const char \*source, size\_t npoints)  
*Print the header of [printDescr\(\)](#).*
- static void [printDescr](#) (const [Feature](#) &d)  
*Print the descriptor of the given feature.*
- static void [writeBMP](#) (const char \*filename, const float \*source, int w, int h)  
*write a BMP file containing the given image (luminance only).*
- static void [writeRawData](#) (const char \*filename, const float \*source, int w, int h)  
*write a raw data file containing h, w and the source float matrix.*

## Data Fields

- [Buffer](#) [buffer](#)  
*buffer containing the image data*

## Protected Member Functions

- bool [resize](#) (int newheight, int newwidth)  
*Resize the current image.*

## Static Protected Member Functions

- static unsigned int [scalarQuantize](#) (float value, const float \*data, size\_t size)  
*Perform scalar quantization on given data.*
- static float [fastScalarQuantize](#) (float value, const float \*data, const float \*output, size\_t size)  
*Perform scalar quantization on given data and return the corresponding output.*
- static float [fastInterpolate](#) (float value, const float \*data, const float \*output, size\_t size)  
*Perform scalar quantization then interpolate the output.*
- static bool [sortPdfPredicate](#) (const [Feature](#) &a, const [Feature](#) &b)  
*Predicate used to order features.*
- static bool [sortPredicate](#) (const float &a, const float &b)  
*Predicate used to order float values.*

### 7.39.1 Detailed Description

A container class for a bidimensional image; it's the base class of all keypoint detector classes.

This class properly deallocates memory when an exception is thrown.

#### Author

Giovanni Cordara, Massimo Balestri

#### Date

2013

## 7.39.2 Constructor & Destructor Documentation

7.39.2.1 `mpeg7cdvs::ImageBuffer::ImageBuffer ( )`

7.39.2.2 `virtual mpeg7cdvs::ImageBuffer::~~ImageBuffer ( )` `[virtual]`

## 7.39.3 Member Function Documentation

7.39.3.1 `static float mpeg7cdvs::ImageBuffer::fastInterpolate ( float value, const float * data, const float * output, size_t size )` `[static]`, `[protected]`

Perform scalar quantization then interpolate the output.

### Parameters

<i>value</i>	the value to quantize
<i>data</i>	the quantization centroids
<i>output</i>	the corresponding output values (probabilities)
<i>size</i>	the size of the quantization centroid array

### Returns

the interpolated probability corresponding to the given value

7.39.3.2 `static float mpeg7cdvs::ImageBuffer::fastScalarQuantize ( float value, const float * data, const float * output, size_t size )` `[static]`, `[protected]`

Perform scalar quantization on given data and return the corresponding output.

### Parameters

<i>value</i>	the value to quantize
<i>data</i>	the quantization centroids
<i>output</i>	the corresponding output values (probabilities)
<i>size</i>	the size of the quantization centroid array

### Returns

the probability corresponding to the given value

7.39.3.3 `static void mpeg7cdvs::ImageBuffer::print ( const std::vector< Feature > & f, const char * source )` `[static]`

Print the given list of features.

### Parameters

<i>f</i>	a vector of features
<i>source</i>	the name of the keypoint detector that produced the given list of features

7.39.3.4 `static void mpeg7cdvs::ImageBuffer::printDescr ( const Feature & d )` `[static]`

Print the descriptor of the given feature.

## Parameters

<i>d</i>	the feature
----------	-------------

7.39.3.5 `static void mpeg7cdvs::ImageBuffer::printHeader ( const char * source, size_t npoints )` [static]

Print the header of [printDescr\(\)](#).

## Parameters

<i>source</i>	the name of the detector under test
<i>npoints</i>	the number of points that will be printed

7.39.3.6 `void mpeg7cdvs::ImageBuffer::read ( int width, int height, const unsigned char * buffer )`

Read a planar luminance image from a buffer.

This method can be used only if the image is already available as an 8-bit planar luminance buffer.

## Parameters

<i>width</i>	width of the image
<i>height</i>	height of the image
<i>buffer</i>	the buffer containing the luminance component of the image

7.39.3.7 `void mpeg7cdvs::ImageBuffer::resample ( ImageBuffer & dest ) const`

Convert this image into a destination image having a different resolution by filtering and sampling the original image.

## Parameters

<i>dest</i>	the destination image
-------------	-----------------------

## Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

7.39.3.8 `void mpeg7cdvs::ImageBuffer::resample ( double rfactor )`

Resample this image using the given reduction factor (the original image is discarded).

## Parameters

<i>rfactor</i>	the reduction factor (must be < 1)
----------------	------------------------------------

## Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

7.39.3.9 `void mpeg7cdvs::ImageBuffer::resampleIfGreater ( int maxSize )`

Resample this image if either the horizontal or the vertical dimension of the image is greater that the given maximum size.

## Parameters

<i>maxSize</i>	the maximum size to set
----------------	-------------------------

## Exceptions

<a href="#">CdvsException</a>	in case of error
-------------------------------	------------------

**7.39.3.10** `bool mpeg7cdvs::ImageBuffer::resize ( int newheight, int newwidth )` `[protected]`

Resize the current image.

## Parameters

<i>newheight</i>	the new height
<i>newwidth</i>	the new width

## Returns

true if successful

**7.39.3.11** `static unsigned int mpeg7cdvs::ImageBuffer::scalarQuantize ( float value, const float * data, size_t size )`  
`[static], [protected]`

Perform scalar quantization on given data.

## Parameters

<i>value</i>	the value to quantize
<i>data</i>	the quantization centroids
<i>size</i>	the size of the quantization centroid array

## Returns

the index such that the distance between value and data[index] is minimum

**7.39.3.12** `static bool mpeg7cdvs::ImageBuffer::sortPdfPredicate ( const Feature & a, const Feature & b )` `[static], [protected]`

Predicate used to order features.

## Parameters

<i>a</i>	first element to compare
<i>b</i>	second element to compare

## Returns

true if  $a > b$

**7.39.3.13** `static bool mpeg7cdvs::ImageBuffer::sortPredicate ( const float & a, const float & b )` `[static], [protected]`

Predicate used to order float values.

## Parameters

<i>a</i>	first element to compare
<i>b</i>	second element to compare

## Returns

true if  $a > b$

## 7.39.3.14 void mpeg7cdvs::ImageBuffer::swap ( ImageBuffer &amp; other )

swap the content of two ImageBuffer(s)

## Parameters

<i>other</i>	the other <a href="#">ImageBuffer</a> instance
--------------	--

7.39.3.15 static void mpeg7cdvs::ImageBuffer::writeBMP ( const char \* filename, const float \* source, int w, int h )  
[static]

write a BMP file containing the given image (luminance only).

7.39.3.16 static void mpeg7cdvs::ImageBuffer::writeRawData ( const char \* filename, const float \* source, int w, int h )  
[static]

write a raw data file containing h, w and the source float matrix.

## 7.39.4 Field Documentation

## 7.39.4.1 Buffer mpeg7cdvs::ImageBuffer::buffer

buffer containing the image data

The documentation for this class was generated from the following file:

- [ImageBuffer.h](#)

## 7.40 mpeg7cdvs::LookUpTable Class Reference

A simple look up table implementation, to perform a bit count very quickly.

```
#include <SCFVIndex.h>
```

## Public Member Functions

- [LookUpTable](#) ()

## Data Fields

- char *f* [(1<< 16)]  
*the look up table*

### 7.40.1 Detailed Description

A simple look up table implementation, to perform a bit count very quickly.

### 7.40.2 Constructor & Destructor Documentation

#### 7.40.2.1 `mpeg7cdvs::LookUpTable::LookUpTable ( )`

### 7.40.3 Field Documentation

#### 7.40.3.1 `char mpeg7cdvs::LookUpTable::f[(1 << 16)]`

the look up table

The documentation for this class was generated from the following file:

- [SCFVIndex.h](#)

## 7.41 Match Class Reference

Helper class to sort images according to a pair of values.

```
#include <Match.h>
```

### Static Public Member Functions

- static bool [sortMatchPredicate](#) (const [Match](#) &m1, const [Match](#) &m2)  
*Predicate used by the STL algorithms `sort()` and `stable_sort()` to rank the images according to the number of common features.*
- static bool [sortMatchByWeight](#) (const [Match](#) &m1, const [Match](#) &m2)  
*Predicate used by the STL algorithms `sort()` and `stable_sort()` to rank the images according to the number of common features.*

### Data Fields

- int [featureInd](#)  
*secondary feature*
- int [otherFeatureInd](#)  
*primary feature*
- double [weight](#)  
*weight of this match*

### 7.41.1 Detailed Description

Helper class to sort images according to a pair of values.

It is used in retrieval to take into account the number of matched points before and after geometric verification.

Author

Gianluca Francini

Date

2011

## 7.41.2 Member Function Documentation

### 7.41.2.1 static bool Match::sortMatchByWeight ( const Match & *m1*, const Match & *m2* ) [inline], [static]

Predicate used by the STL algorithms sort() and stable\_sort() to rank the images according to the number of common features.

#### Parameters

<i>m1</i>	first image values.
<i>m2</i>	second image values.

#### Returns

true if *m1* is strictly greater than *m2*.

References otherFeatureInd, and weight.

### 7.41.2.2 static bool Match::sortMatchPredicate ( const Match & *m1*, const Match & *m2* ) [inline], [static]

Predicate used by the STL algorithms sort() and stable\_sort() to rank the images according to the number of common features.

#### Parameters

<i>m1</i>	first image values.
<i>m2</i>	second image values.

#### Returns

true if *m1* is strictly greater than *m2*.

References featureInd, and otherFeatureInd.

## 7.41.3 Field Documentation

### 7.41.3.1 int Match::featureInd

secondary feature

Referenced by sortMatchPredicate().

### 7.41.3.2 int Match::otherFeatureInd

primary feature

Referenced by sortMatchByWeight(), and sortMatchPredicate().

### 7.41.3.3 double Match::weight

weight of this match

Referenced by sortMatchByWeight().

The documentation for this class was generated from the following file:

- [Match.h](#)

## 7.42 mpeg7cdvs::Parameters Class Reference

Container for all encoding/decoding parameters associated to each target bitrate defined by MPEG CDVS.

```
#include <Parameters.h>
```

### Public Member Functions

- [Parameters](#) (void)
- [~Parameters](#) (void)
- int [readParameters](#) (const char \*filename, int mode)  
*Read text file and load parameters related to a specified mode.*
- int [readParameters](#) (int mode)  
*Load hard-coded parameters related to a specified mode.*
- unsigned int [getModelID](#) () const  
*Get the modelID of this set of parameters.*

### Static Public Member Functions

- static void [readAll](#) (const char \*filename, [Parameters](#) params[])  
*Read all modes form a text file into the given vector of parameters.*
- static void [readAll](#) ([Parameters](#) params[])  
*Load all modes using default parameters into the given vector of parameters.*

### Data Fields

- int [descLength](#)  
*length in bytes of the CDVS descriptor (i.e. 512, 1024, 2048)*
- int [resizeMaxSize](#)  
*maximum size of one side of the image*
- int [blockWidth](#)  
*coordinate coding: spatial resolution of the coordinates (max error = blockWidth/2)*
- int [ctxTableIdx](#)  
*coordinate coding: index of the context table to use*
- char [modeExt](#) [40]  
*descriptor extension*
- unsigned int [selectMaxPoints](#)  
*feature extraction: max number of points used to describe an image*
- unsigned int [numRelevantPoints](#)  
*feature extraction: number of points considered relevant in the retrieval process*
- int [numberOfElementGroups](#)  
*feature compression: number of element groups in a compressed local feature descriptor*
- float [ratioThreshold](#)  
*DISTRAT: threshold for descriptor matching.*
- unsigned int [minNumInliers](#)  
*DISTRAT: min number of inliers after the geometric check.*
- double [wmThreshold](#)  
*Weighted matching threshold.*
- double [wmThreshold2Way](#)  
*Two way matching weighted threshold.*
- double [wmMixed](#)



- Weighted matching threshold for mixed cases.*

  - double [wmMixed2Way](#)

*Two way weighted matching threshold for mixed cases.*
- int [debugLevel](#)

*0 = off, 1 = on (quiet), 2 = on (verbose), 3 = verbose + dump files*
- int [ransacNumTests](#)

*RANSAC: number of iterations in RANSAC.*
- float [ransacThreshold](#)

*RANSAC: distortion threshold to be used by RANSAC.*
- unsigned int [chiSquarePercentile](#)

*percentile used in DISTRAT for Chi-square computation*
- int [retrievalLoops](#)

*number of loops performed in the final stage of the retrieval process*
- double [wmRetrieval](#)

*Weighted matching threshold for retrieval.*
- double [wmRetrieval2Way](#)

*Two way weighted matching threshold for retrieval.*
- int [retrievalMaxPoints](#)

*max number of points used in the retrieval experiment*
- int [queryExpansionLoops](#)

*number of query expansion loops to perform in the retrieval experiment*
- float [scfvThreshold](#)

*threshold value to control the sparsity of scfv vector – add by linjie*
- bool [hasVar](#)

*indicates if using the gradient vector w.r.t the variance of Gaussian function – add by linjie*
- float [locationBits](#)

*average bits per key point to encode location information;*
- bool [hasBitSelection](#)

*indicates if the Global Descriptor uses the bit selection algorithm to reduce its size*
- float [gdThreshold](#)

*global descriptor threshold*
- float [gdThresholdMixed](#)

*global descriptor threshold for mixed cases*

## Static Public Attributes

- static const int [nBits](#) = 8

*number of bits to represent the mode ID*
- static const int [nModes](#) = 7

*the max number of processing modes*

### 7.42.1 Detailed Description

Container for all encoding/decoding parameters associated to each target bitrate defined by MPEG CDVS.

The actual value of each parameter is read from a text file, although default values are provided in this class. Each set of parameters is associated to a profile, and a single instance of this class may only contain parameters for a single profile. Please note that changing any parameter in the parameters file may break compatibility between encoder and decoder.

**Author**

Giovanni Cordara, Massimo Balestri

**Date**

2011

**7.42.2 Constructor & Destructor Documentation**

7.42.2.1 `mpeg7cdvs::Parameters::Parameters ( void )`

7.42.2.2 `mpeg7cdvs::Parameters::~~Parameters ( void )`

**7.42.3 Member Function Documentation**

7.42.3.1 `unsigned int mpeg7cdvs::Parameters::getModelID ( ) const`

Get the modelID of this set of parameters.

The mode id cannot be changed; it is read from the parameters file.

**Returns**

the mode id value

7.42.3.2 `static void mpeg7cdvs::Parameters::readAll ( const char * filename, Parameters params[ ] ) [static]`

Read all modes form a text file into the given vector of parameters.

**Parameters**

<i>filename</i>	filename pathname of the file containing the specific parameter values. If NULL it is ignored.
<i>params</i>	the vector of parameters to fill.

7.42.3.3 `static void mpeg7cdvs::Parameters::readAll ( Parameters params[ ] ) [static]`

Load all modes using default parameters into the given vector of parameters.

**Parameters**

<i>params</i>	the vector of parameters to fill.
---------------	-----------------------------------

7.42.3.4 `int mpeg7cdvs::Parameters::readParameters ( const char * filename, int mode )`

Read text file and load parameters related to a specified mode.

A maximum of nModes are supported.

**Parameters**

<i>filename</i>	pathname of the file containing the specific parameter values. If NULL it is ignored.
<i>mode</i>	one of the MPEG CDVS supported modes, from 0 to 6.

**Returns**

0 if successful, an error code otherwise.

7.42.3.5 int mpeg7cdvs::Parameters::readParameters ( int *mode* )

Load hard-coded parameters related to a specified mode.

A maximum of nModes are supported.

## Parameters

<i>mode</i>	one of the MPEG CDVS supported modes, from 0 to 6.
-------------	--

## Returns

0 if successful, an error code otherwise.

## 7.42.4 Field Documentation

## 7.42.4.1 int mpeg7cdvs::Parameters::blockWidth

coordinate coding: spatial resolution of the coordinates (max error = blockWidth/2)

## 7.42.4.2 unsigned int mpeg7cdvs::Parameters::chiSquarePercentile

percentile used in DISTRAT for Chi-square computation

## 7.42.4.3 int mpeg7cdvs::Parameters::ctxTableIdx

coordinate coding: index of the context table to use

## 7.42.4.4 int mpeg7cdvs::Parameters::debugLevel

0 = off, 1 = on (quiet), 2 = on (verbose), 3 = verbose + dump files

## 7.42.4.5 int mpeg7cdvs::Parameters::descLength

length in bytes of the CDVS descriptor (i.e. 512, 1024, 2048)

## 7.42.4.6 float mpeg7cdvs::Parameters::gdThreshold

global descriptor threshold

## 7.42.4.7 float mpeg7cdvs::Parameters::gdThresholdMixed

global descriptor threshold for mixed cases

## 7.42.4.8 bool mpeg7cdvs::Parameters::hasBitSelection

indicates if the Global Descriptor uses the bit selection algorithm to reduce its size

## 7.42.4.9 bool mpeg7cdvs::Parameters::hasVar

indicates if using the gradient vector w.r.t the variance of Gaussian function – add by linjie

**7.42.4.10 float mpeg7cdvs::Parameters::locationBits**

average bits per key point to encode location information;

**7.42.4.11 unsigned int mpeg7cdvs::Parameters::minNumInliers**

DISTRAT: min number of inliers after the geometric check.

**7.42.4.12 char mpeg7cdvs::Parameters::modeExt[40]**

descriptor extension

**7.42.4.13 const int mpeg7cdvs::Parameters::nBits = 8 [static]**

number of bits to represent the mode ID

**7.42.4.14 const int mpeg7cdvs::Parameters::nModes = 7 [static]**

the max number of processing modes

**7.42.4.15 int mpeg7cdvs::Parameters::numberOfElementGroups**

feature compression: number of element groups in a compressed local feature descriptor

**7.42.4.16 unsigned int mpeg7cdvs::Parameters::numRelevantPoints**

feature extraction: number of points considered relevant in the retrieval process

**7.42.4.17 int mpeg7cdvs::Parameters::queryExpansionLoops**

number of query expansion loops to perform in the retrieval experiment

**7.42.4.18 int mpeg7cdvs::Parameters::ransacNumTests**

RANSAC: number of iterations in RANSAC.

**7.42.4.19 float mpeg7cdvs::Parameters::ransacThreshold**

RANSAC: distortion threshold to be used by RANSAC.

**7.42.4.20 float mpeg7cdvs::Parameters::ratioThreshold**

DISTRAT: threshold for descriptor matching.

**7.42.4.21 int mpeg7cdvs::Parameters::resizeMaxSize**

maximum size of one side of the image

**7.42.4.22 int mpeg7cdvs::Parameters::retrievalLoops**

number of loops performed in the final stage of the retrieval process

**7.42.4.23 int mpeg7cdvs::Parameters::retrievalMaxPoints**

max number of points used in the retrieval experiment

**7.42.4.24 float mpeg7cdvs::Parameters::scfvThreshold**

threshold value to control the sparsity of scfv vector – add by linjie

**7.42.4.25 unsigned int mpeg7cdvs::Parameters::selectMaxPoints**

feature extraction: max number of points used to describe an image

**7.42.4.26 double mpeg7cdvs::Parameters::wmMixed**

Weighted matching threshold for mixed cases.

**7.42.4.27 double mpeg7cdvs::Parameters::wmMixed2Way**

Two way weighted matching threshold for mixed cases.

**7.42.4.28 double mpeg7cdvs::Parameters::wmRetrieval**

Weighted matching threshold for retrieval.

**7.42.4.29 double mpeg7cdvs::Parameters::wmRetrieval2Way**

Two way weighted matching threshold for retrieval.

**7.42.4.30 double mpeg7cdvs::Parameters::wmThreshold**

Weighted matching threshold.

**7.42.4.31 double mpeg7cdvs::Parameters::wmThreshold2Way**

Two way matching weighted threshold.

The documentation for this class was generated from the following file:

- [Parameters.h](#)

**7.43 mpeg7cdvs::PointPairs Class Reference**

Parameter class, used to pass around matched point coordinates.

```
#include <PointPairs.h>
```

## Public Member Functions

- [PointPairs](#) ()  
*default constructor*
- [PointPairs](#) (int maxPairs)  
*alternate constructor*
- [PointPairs](#) (const [PointPairs](#) &other)  
*copy constructor*
- [PointPairs](#) & operator= ([PointPairs](#) other)  
*assignment operator*
- virtual [~PointPairs](#) ()  
*destructor*
- bool [hasLocalizationInliers](#) () const  
*Return true if the [PointPairs](#) instance contains localization information.*
- void [addPair](#) (float x\_1, float x\_2, float y\_1, float y\_2, double weight=0, int mtype=[match\\_2way\\_DISJOINT1](#))  
*Add a new pair to the matched pairs and increment the nMatched count.*
- void [toFullResolution](#) (int query\_maxres, int query\_fullres, int ref\_maxres, int ref\_fullres)  
*Convert matched coordinates to the full resolution of the original image.*
- double [getTotalWeight](#) () const  
*Get the total weight of all matched points.*
- double [getInlierWeight](#) () const  
*Get the weight of inlier points (these are a subset of all points, which passed the geometric verification test).*

## Data Fields

- double [local\\_score](#)  
*matching score provided by local descriptors*
- double [global\\_score](#)  
*matching score provided by global descriptor*
- double [local\\_threshold](#)  
*local threshold used for weighted matching of keypoints*
- double [global\\_threshold](#)  
*global threshold used for matching the global descriptor*
- double [score](#)  
*final normalized score (in a range from 0 to 1)*
- int [nMatched](#)  
*actual number of matched points*
- int [size](#)  
*size of all buffers*
- float \* [x1](#)  
*x-coordinates of matching points of the first image*
- float \* [x2](#)  
*y-coordinates of matching points of the first image*
- float \* [y1](#)  
*x-coordinates of matching points of the second image*
- float \* [y2](#)  
*y-coordinates of matching points of the second image*
- double \* [weights](#)  
*weights of each match*
- int \* [match\\_dirs](#)  
*indicates the direction of matching ( $A \leq B$ ,  $A = B$ ,  $B = A$ ) in 2-way matching (not used in 1-way matching)*

- int [nInliers](#)  
*indicates the number of pairs which actually match according to the geometric verification*
- int \* [inlierIndexes](#)  
*indicates the indices of the pairs that have passed the geometric verification*

### 7.43.1 Detailed Description

Parameter class, used to pass around matched point coordinates.

Used by the matching methods in the Feature\*List\* classes.

#### Author

Emanuele Plebani

#### Date

2012

### 7.43.2 Constructor & Destructor Documentation

#### 7.43.2.1 mpeg7cdvs::PointPairs::PointPairs ( )

default constructor

#### 7.43.2.2 mpeg7cdvs::PointPairs::PointPairs ( int *maxPairs* )

alternate constructor

#### 7.43.2.3 mpeg7cdvs::PointPairs::PointPairs ( const PointPairs & *other* )

copy constructor

#### 7.43.2.4 virtual mpeg7cdvs::PointPairs::~~PointPairs ( ) [virtual]

destructor

### 7.43.3 Member Function Documentation

#### 7.43.3.1 void mpeg7cdvs::PointPairs::addPair ( float *x\_1*, float *x\_2*, float *y\_1*, float *y\_2*, double *weight* = 0, int *mtype* = *match\_2way\_DISJOINT1* )

Add a new pair to the matched pairs and increment the nMatched count.

#### Parameters

<i>x_1</i>	x-coordinate of matching point of the first image
<i>x_2</i>	y-coordinate of matching point of the first image
<i>y_1</i>	x-coordinate of matching point of the second image

<i>y_2</i>	y-coordinate of matching point of the second image
<i>weight</i>	weight of matching (defaults is zero if not used)
<i>mtype</i>	direction of matching for 2-way matching ( $A \leq B$ , $A > B$ or $B > A$ (defaults is $A > B$ if not used)

#### 7.43.3.2 `double mpeg7cdvs::PointPairs::getInlierWeight ( ) const`

Get the weight of inlier points (these are a subset of all points, which passed the geometric verification test).

##### Returns

the weight of inlier points.

#### 7.43.3.3 `double mpeg7cdvs::PointPairs::getTotalWeight ( ) const`

Get the total weight of all matched points.

##### Returns

the total weight of all matched points.

#### 7.43.3.4 `bool mpeg7cdvs::PointPairs::hasLocalizationInliers ( ) const`

Return true if the [PointPairs](#) instance contains localization information.

#### 7.43.3.5 `PointPairs& mpeg7cdvs::PointPairs::operator= ( PointPairs other )`

assignment operator

#### 7.43.3.6 `void mpeg7cdvs::PointPairs::toFullResolution ( int query_maxres, int query_fullres, int ref_maxres, int ref_fullres )`

Convert matched coordinates to the full resolution of the original image.

This method may change the scale of the coordinates of matching points stored in x1, x2, y1, y2.

##### Parameters

<i>query_maxres</i>	the greater dimension of the (possibly scaled) query image.
<i>query_fullres</i>	the greater dimension of the original query image.
<i>ref_maxres</i>	the greater dimension of the (possibly scaled) reference image.
<i>ref_fullres</i>	the greater dimension of the original reference image.

### 7.43.4 Field Documentation

#### 7.43.4.1 `double mpeg7cdvs::PointPairs::global_score`

matching score provided by global descriptor

#### 7.43.4.2 `double mpeg7cdvs::PointPairs::global_threshold`

global threshold used for matching the global descriptor



**7.43.4.3 int\* mpeg7cdvs::PointPairs::inlierIndexes**

indicates the indices of the pairs that have passed the geometric verification

**7.43.4.4 double mpeg7cdvs::PointPairs::local\_score**

matching score provided by local descriptors

**7.43.4.5 double mpeg7cdvs::PointPairs::local\_threshold**

local threshold used for weighted matching of keypoints

**7.43.4.6 int\* mpeg7cdvs::PointPairs::match\_dirs**

indicates the direction of matching ( $A \leq B$ ,  $A \geq B$ ,  $B \geq A$ ) in 2-way matching (not used in 1-way matching)

**7.43.4.7 int mpeg7cdvs::PointPairs::nInliers**

indicates the number of pairs which actually match according to the geometric verification

**7.43.4.8 int mpeg7cdvs::PointPairs::nMatched**

actual number of matched points

**7.43.4.9 double mpeg7cdvs::PointPairs::score**

final normalized score (in a range from 0 to 1)

**7.43.4.10 int mpeg7cdvs::PointPairs::size**

size of all buffers

**7.43.4.11 double\* mpeg7cdvs::PointPairs::weights**

weights of each match

**7.43.4.12 float\* mpeg7cdvs::PointPairs::x1**

x-coordinates of matching points of the first image

**7.43.4.13 float\* mpeg7cdvs::PointPairs::x2**

y-coordinates of matching points of the first image

**7.43.4.14 float\* mpeg7cdvs::PointPairs::y1**

x-coordinates of matching points of the second image

#### 7.43.4.15 float\* mpeg7cdvs::PointPairs::y2

y-coordinates of matching points of the second image

The documentation for this class was generated from the following file:

- [PointPairs.h](#)

## 7.44 Polygon Class Reference

A base class for all polygons;.

```
#include <BoundingBox.h>
```

### Public Member Functions

- [Polygon](#) ()
- double [inter](#) (int na, const [point](#) \*a, int nb, const [point](#) \*b)  
*Compute the intersection area of two polygons.*

#### 7.44.1 Detailed Description

A base class for all polygons;.

#### 7.44.2 Constructor & Destructor Documentation

##### 7.44.2.1 Polygon::Polygon ( )

#### 7.44.3 Member Function Documentation

##### 7.44.3.1 double Polygon::inter ( int na, const point \* a, int nb, const point \* b )

Compute the intersection area of two polygons.

#### Parameters

<i>na</i>	number of points of polygon A
<i>a</i>	points of polygon A
<i>nb</i>	number of points of polygon B
<i>b</i>	points of polygon B

#### Returns

the intersection area.

The documentation for this class was generated from the following file:

- [BoundingBox.h](#)

## 7.45 Projective2D Class Reference

Computes and performs homographies between two images.

```
#include <Projective2D.h>
```

## Public Member Functions

- [Projective2D](#) ()
- bool [isIdentity](#) () const  
*Check if the projectivity is the identity.*
- void [moveByHomography](#) ([Point2DArray](#) &outCoordinates, [Point2DArray](#) const &inCoordinates, Eigen::Matrix3f const &H) const  
*Transforms input coordinates into output coordinates by a homography.*
- void [moveByHomography](#) ([Point2DArray](#) &outCoordinates, [Point2DArray](#) const &inCoordinates) const  
*Transforms input coordinates into output coordinates using the homography stored in the object.*
- void [modelFit](#) (Eigen::VectorXf &distances, [Point2DArray](#) const &fromPoints, [Point2DArray](#) const &toPoints, const Eigen::Matrix3f &H) const  
*Computes the error when approximating a target set of points by transforming another set of points through a homography.*
- void [makeHomography](#) (Eigen::Matrix3f &H, [Point2DArray](#) const &fromX, [Point2DArray](#) const &toX) const  
*Computes a homography that approximates one set of points by transforming another set of points.*
- void [makeHomography](#) ([Point2DArray](#) const &fromX, [Point2DArray](#) const &toX)  
*Computes a homography that approximates one set of points by transforming another set of points, and stores it internally.*
- void [ransac](#) (Eigen::VectorXi &consensusSet, [Point2DArray](#) const &fromPoints, [Point2DArray](#) const &toPoints, int nTests, float threshold)  
*Identifies correct matches as a subset of matches represented by two sets of points using the RANSAC algorithm.*
- void [ransac](#) ([Point2DArray](#) const &fromPoints, [Point2DArray](#) const &toPoints, int nTests, float threshold)  
*Builds a new homography through the RANSAC algorithm and saves it internally.*

### 7.45.1 Detailed Description

Computes and performs homographies between two images.

Author

Skjalg Lepsoy

Date

2011

### 7.45.2 Constructor & Destructor Documentation

#### 7.45.2.1 [Projective2D::Projective2D](#) ( )

### 7.45.3 Member Function Documentation

#### 7.45.3.1 bool [Projective2D::isIdentity](#) ( ) const

Check if the projectivity is the identity.

Returns

true if the projectivity is the identity.

**7.45.3.2 void Projective2D::makeHomography ( Eigen::Matrix3f & *H*, Point2DArray const & *fromX*, Point2DArray const & *toX* ) const**

Computes a homography that approximates one set of points by transforming another set of points.

Uses the direct linear transformation method with normalization described by Hartley & Zisserman: Multiview Geometry 2nd edition, Alg. 4.2 page 109.

**Parameters**

<i>H</i>	output homography.
<i>fromX</i>	input points to be transformed.
<i>toX</i>	input points to be approximated.

**7.45.3.3 void Projective2D::makeHomography ( Point2DArray const & *fromX*, Point2DArray const & *toX* )**

Computes a homography that approximates one set of points by transforming another set of points, and stores it internally.

Uses the direct linear transformation method with normalization described by Hartley & Zisserman: Multiview Geometry 2nd edition, Alg. 4.2 page 109.

**Parameters**

<i>fromX</i>	input points to be transformed.
<i>toX</i>	input points to be approximated.

**7.45.3.4 void Projective2D::modelFit ( Eigen::VectorXf & *distances*, Point2DArray const & *fromPoints*, Point2DArray const & *toPoints*, const Eigen::Matrix3f & *H* ) const**

Computes the error when approximating a target set of points by transforming another set of points through a homography.

**Parameters**

<i>distances</i>	output approximation errors.
<i>fromPoints</i>	input list of points to be transformed.
<i>toPoints</i>	input target list of points.
<i>H</i>	homography to be used.

**7.45.3.5 void Projective2D::moveByHomography ( Point2DArray & *outCoordinates*, Point2DArray const & *inCoordinates*, Eigen::Matrix3f const & *H* ) const**

Transforms input coordinates into output coordinates by a homography.

**Parameters**

<i>outCoordinates</i>	output coordinates.
<i>inCoordinates</i>	input coordinates.
<i>H</i>	homography to be used.

**7.45.3.6 void Projective2D::moveByHomography ( Point2DArray & *outCoordinates*, Point2DArray const & *inCoordinates* ) const**

Transforms input coordinates into output coordinates using the homography stored in the object.

## Parameters

<i>outCoordinates</i>	output coordinates.
<i>inCoordinates</i>	input coordinates.

7.45.3.7 `void Projective2D::ransac ( Eigen::VectorXi & consensusSet, Point2DArray const & fromPoints, Point2DArray const & toPoints, int nTests, float threshold )`

Identifies correct matches as a subset of matches represented by two sets of points using the RANSAC algorithm. The transformation between the two sets is a homography.

## Parameters

<i>consensusSet</i>	output binary vector indicating the correct matches.
<i>fromPoints</i>	input list of matched points of the first image.
<i>toPoints</i>	input list of matched points of the second image.
<i>nTests</i>	input number of iterations to use in RANSAC (typically 10).
<i>threshold</i>	input maximum distance in order to judge a match as correct.

7.45.3.8 `void Projective2D::ransac ( Point2DArray const & fromPoints, Point2DArray const & toPoints, int nTests, float threshold )`

Builds a new homography through the RANSAC algorithm and saves it internally.

## Parameters

<i>fromPoints</i>	input list of matched points of the first image.
<i>toPoints</i>	input list of matched points of the second image.
<i>nTests</i>	input number of iterations to use in RANSAC (typically 10).
<i>threshold</i>	input maximum distance in order to judge a match as correct.

The documentation for this class was generated from the following file:

- [Projective2D.h](#)

## 7.46 QRA Struct Reference

Structure containing the name of a query image and the list of names of matching reference pictures.

```
#include <map.h>
```

## Data Fields

- [FILENAME query](#)  
*query image*
- `int n_matches`  
*number of matching images*
- [FILENAME matches \[MAX\\_MATCHES\]](#)  
*list of matches*

### 7.46.1 Detailed Description

Structure containing the name of a query image and the list of names of matching reference pictures.

## 7.46.2 Field Documentation

### 7.46.2.1 FILENAME QRA::matches[MAX\_MATCHES]

list of matches

### 7.46.2.2 int QRA::n\_matches

number of matching images

### 7.46.2.3 FILENAME QRA::query

query image

The documentation for this struct was generated from the following file:

- [map.h](#)

## 7.47 mpeg7cdvs::RetrievalData Struct Reference

A structure containing the output of a retrieval operation.

```
#include <CdvsPoint.h>
```

### Data Fields

- unsigned int [nMatched](#)  
*number of matched points*
- unsigned int [nInliers](#)  
*number of inliers points*
- unsigned int [index](#)  
*index of this image in the image DB*
- float [gScore](#)  
*score assigned by the global descriptor matching*
- float [fScore](#)  
*score assigned by the local descriptors matching*

### 7.47.1 Detailed Description

A structure containing the output of a retrieval operation.

### 7.47.2 Field Documentation

#### 7.47.2.1 float mpeg7cdvs::RetrievalData::fScore

score assigned by the local descriptors matching

#### 7.47.2.2 float mpeg7cdvs::RetrievalData::gScore

score assigned by the global descriptor matching

## 7.47.2.3 unsigned int mpeg7cdvs::RetrievalData::index

index of this image in the image DB

## 7.47.2.4 unsigned int mpeg7cdvs::RetrievalData::nInliers

number of inliers points

## 7.47.2.5 unsigned int mpeg7cdvs::RetrievalData::nMatched

number of matched points

The documentation for this struct was generated from the following file:

- [CdvsPoint.h](#)

## 7.48 mpeg7cdvs::SCFVFactory Class Reference

A class to produce SCFV signatures.

```
#include <SCFVIndex.h>
```

### Public Member Functions

- [SCFVFactory](#) ()
- void [init](#) (const [Parameters](#) &params)  
*Initialize the class with the correct set of parameters.*
- void [generateSCFV](#) (const [FeatureList](#) &featureList, [SCFVSignature](#) &signature, int nNumFeatures) const  
*Generate a global descriptor signature using the given feature list.*
- bool [hasVariance](#) () const  
*Indicates if using the variance information of the Gaussian function.*
- bool [hasBitSelection](#) () const  
*Indicates if using the bit selection information.*

### 7.48.1 Detailed Description

A class to produce SCFV signatures.

### 7.48.2 Constructor & Destructor Documentation

## 7.48.2.1 mpeg7cdvs::SCFVFactory::SCFVFactory ( )

### 7.48.3 Member Function Documentation

## 7.48.3.1 void mpeg7cdvs::SCFVFactory::generateSCFV ( const FeatureList &amp; featureList, SCFVSignature &amp; signature, int nNumFeatures ) const

Generate a global descriptor signature using the given feature list.

## Parameters

<i>featureList</i>	the key points to use as input information
<i>signature</i>	the output signature
<i>nNumFeatures</i>	the number of features to encode

7.48.3.2 `bool mpeg7cdvs::SCFVFactory::hasBitSelection ( ) const [inline]`

Indicates if using the bit selection information.

## Returns

true if using the bit selection information

7.48.3.3 `bool mpeg7cdvs::SCFVFactory::hasVariance ( ) const [inline]`

Indicates if using the variance information of the Gaussian function.

## Returns

true if using the variance information

7.48.3.4 `void mpeg7cdvs::SCFVFactory::init ( const Parameters & params )`

Initialize the class with the correct set of parameters.

## Parameters

<i>params</i>	the input parameters to use
---------------	-----------------------------

The documentation for this class was generated from the following file:

- [SCFVIndex.h](#)

## 7.49 mpeg7cdvs::SCFVIndex Class Reference

A class to manage an indexed list of SCFV signatures.

```
#include <SCFVIndex.h>
```

### Public Member Functions

- [SCFVIndex](#) ()
- void [append](#) (const [SCFVSignature](#) &scfvSignature)  
*append the given SCFV signature to the current index*
- void [replace](#) (size\_t index, const [SCFVSignature](#) &scfvSignature)  
*replace the given SCFV signature with the given one at the given index*
- void [write](#) (std::string sIndexName) const  
*write the SCFV index to file*
- void [read](#) (std::string sIndexName)  
*read the SCFV index from file*
- void [query](#) (const [SCFVSignature](#) &querySignature, std::vector< std::pair< double, unsigned int > > &v-ImageScoresNumbers, size\_t numRankedOutput) const



Use a binary SCFV signature as a query to retrieve a ranked list of signatures matching the given one.

- void [query\\_bitselection](#) (const [SCFVSignature](#) &querySignature, std::vector< std::pair< double, unsigned int > > &vImageScoresNumbers, size\_t numRankedOutput) const

Use a subset of a binary SCFV signature as a query to retrieve a ranked list of signatures matching the given one.

- void [generateWeight](#) (const [SCFVSignature](#) &querySignature, float \*W2\_log, float \*W2\_log\_var, float weight\_base) const

Produces an optional table of weights to reduce the importance of features that are too common.

- size\_t [numberImages](#) () const

Get the number of images (actually signatures) contained in this index.

- const [SCFVSignature](#) & [getImage](#) (unsigned int index) const

Get the SCFV signature of a specific image.

- void [resize](#) (size\_t num)

Resize the index to num elements.

- void [reserve](#) (size\_t num)

Reserve memory for the given number of signatures.

- void [clear](#) ()

Clear all signatures.

- float [matchImages](#) (const [SCFVSignature](#) &signature1, const [SCFVSignature](#) &signature2, unsigned int \*pNumWords1, unsigned int \*pNumWords2, unsigned int \*overlap) const

Match two signatures and return a matching score.

- float [matchImages\\_bitselection](#) (const [SCFVSignature](#) &signature1, const [SCFVSignature](#) &signature2, unsigned int \*pNumWords1, unsigned int \*pNumWords2, unsigned int \*overlap) const

Match two signatures applying bit selection and return a matching score.

- void [loadHammingWeight](#) ()

Initialize the index with Hamming distance weights.

### 7.49.1 Detailed Description

A class to manage an indexed list of SCFV signatures.

Includes methods to read/write/append SCFV signatures, to use a signature as a query, and to match two signatures.

### 7.49.2 Constructor & Destructor Documentation

#### 7.49.2.1 mpeg7cdvs::SCFVIndex::SCFVIndex ( )

### 7.49.3 Member Function Documentation

#### 7.49.3.1 void mpeg7cdvs::SCFVIndex::append ( const [SCFVSignature](#) & scfvSignature )

append the given SCFV signature to the current index

#### 7.49.3.2 void mpeg7cdvs::SCFVIndex::clear ( ) [inline]

Clear all signatures.

#### 7.49.3.3 void mpeg7cdvs::SCFVIndex::generateWeight ( const [SCFVSignature](#) & querySignature, float \* W2\_log, float \* W2\_log\_var, float weight\_base ) const

Produces an optional table of weights to reduce the importance of features that are too common.

## Parameters

<i>querySignature</i>	the query image signature
<i>W2_log</i>	(output) the logarithmic weight for mean values
<i>W2_log_var</i>	(output) the logarithmic weight for variance values
<i>weight_base</i>	the basic weight from which the table is produced.

7.49.3.4 `const SCFVSignature& mpeg7cdvs::SCFVIndex::getImage ( unsigned int index ) const` `[inline]`

Get the SCFV signature of a specific image.

## Parameters

<i>index</i>	index of the image in the database of images.
--------------	---

## Returns

the image signature

7.49.3.5 `void mpeg7cdvs::SCFVIndex::loadHammingWeight ( )`

Initialize the index with Hamming distance weights.

7.49.3.6 `float mpeg7cdvs::SCFVIndex::matchImages ( const SCFVSignature & signature1, const SCFVSignature & signature2, unsigned int * pNumWords1, unsigned int * pNumWords2, unsigned int * overlap ) const`

[Match](#) two signatures and return a matching score.

## Parameters

<i>signature1</i>	the first SCFV signature
<i>signature2</i>	the second SCFV signature
<i>pNumWords1</i>	the visited number of words of signature1
<i>pNumWords2</i>	the visited number of words of signature2
<i>overlap</i>	unused

## Returns

the matching score

7.49.3.7 `float mpeg7cdvs::SCFVIndex::matchImages_bitselection ( const SCFVSignature & signature1, const SCFVSignature & signature2, unsigned int * pNumWords1, unsigned int * pNumWords2, unsigned int * overlap ) const`

[Match](#) two signatures applying bit selection and return a matching score.

## Parameters

<i>signature1</i>	the first SCFV signature
<i>signature2</i>	the second SCFV signature
<i>pNumWords1</i>	the visited number of words of signature1

<i>pNumWords2</i>	the visited number of words of signature2
<i>overlap</i>	unused

**Returns**

the matching score

**7.49.3.8** `size_t mpeg7cdvs::SCFVIndex::numberImages ( ) const [inline]`

Get the number of images (actually signatures) contained in this index.

**Returns**

the number of images.

**7.49.3.9** `void mpeg7cdvs::SCFVIndex::query ( const SCFVSignature & querySignature, std::vector< std::pair< double, unsigned int > > & vImageScoresNumbers, size_t numRankedOuput ) const`

Use a binary SCFV signature as a query to retrieve a ranked list of signatures matching the given one.

**Parameters**

<i>querySignature</i>	the query signature
<i>vImageScores-Numbers</i>	the output ordered list of images matching the query
<i>numRanked-Ouput</i>	the number of maximum output images required

**7.49.3.10** `void mpeg7cdvs::SCFVIndex::query_bitselection ( const SCFVSignature & querySignature, std::vector< std::pair< double, unsigned int > > & vImageScoresNumbers, size_t numRankedOuput ) const`

Use a subset of a binary SCFV signature as a query to retrieve a ranked list of signatures matching the given one.

**Parameters**

<i>querySignature</i>	the query signature
<i>vImageScores-Numbers</i>	the output ordered list of images matching the query
<i>numRanked-Ouput</i>	the number of maximum output images required

**7.49.3.11** `void mpeg7cdvs::SCFVIndex::read ( std::string sIndexName )`

read the SCFV index from file

**7.49.3.12** `void mpeg7cdvs::SCFVIndex::replace ( size_t index, const SCFVSignature & scfvSignature )`

replace the given SCFV signature with the given one at the given index

**7.49.3.13** `void mpeg7cdvs::SCFVIndex::reserve ( size_t num ) [inline]`

Reserve memory for the given number of signatures.

## Parameters

<i>num</i>	the number of signatures to be reserved in the index.
------------	---

7.49.3.14 `void mpeg7cdvs::SCFVIndex::resize ( size_t num ) [inline]`

Resize the index to num elements.

## Parameters

<i>num</i>	the number of elements required to be in the index
------------	--

7.49.3.15 `void mpeg7cdvs::SCFVIndex::write ( std::string sIndexName ) const`

write the SCFV index to file

The documentation for this class was generated from the following file:

- [SCFVIndex.h](#)

## 7.50 mpeg7cdvs::SCFVSignature Class Reference

Container class for a Scalable Fisher Vector binary signature; allows reading/writing from/to a bitstream, fetching/storing from/into a file, and comparing a signature with another.

```
#include <SCFVIndex.h>
```

### Public Member Functions

- [SCFVSignature](#) (bool [hasVar](#), bool [hasBitSelection](#))  
*Constructor declaring if this signature contains variance information and bit selection.*
- void [clear](#) ()  
*clear all data*
- size\_t [size](#) () const  
*get the size of the binary signature (uncompressed)*
- int [compressedNumBits](#) () const  
*get the number of bits of the encoded signature (compressed)*
- void [write](#) (BitOutputStream &out) const  
*write the binary signature into the given output stream*
- void [read](#) (BitInputStream &in)  
*read the binary signature from the given input stream*
- unsigned int [getVisited](#) () const  
*get the number of visited words*
- void [setVisited](#) ()  
*compute and store the correct number of visited words*
- float [getNorm](#) () const  
*get the norm of this signature*
- void [setNorm](#) ()  
*compute and store the correct norm for this signature*
- bool [hasVar](#) () const  
*tell if this signature has variance information*
- void [hasVar](#) (bool value)

- set this signature as one containing variance information (if value is true)*
- bool [hasBitSelection](#) () const  
*tell if this signature performs bit selection*
- void [hasBitSelection](#) (bool value)  
*set this signature as one performing bit selection (if value is true)*
- int [compare](#) (const [SCFVSignature](#) &other) const  
*compare two signatures (only for debugging)*
- void [toFile](#) (FILE \*file) const  
*write the signature to file*
- void [fromFile](#) (FILE \*file)  
*read the signature from file*
- void [print](#) () const  
*print a summary of the signature data*

## Data Fields

- unsigned int [m\\_vWordBlock](#) [[numberCentroids](#)]  
*Scalable Fisher Vector binary signature.*
- unsigned int [m\\_vWordVarBlock](#) [[numberCentroids](#)]  
*Scalable Fisher Vector binary variance.*

## Static Public Attributes

- static const unsigned int [table\\_bit\\_selection](#) []  
*subset of bits used in low bitrate applications*

### 7.50.1 Detailed Description

Container class for a Scalable Fisher Vector binary signature; allows reading/writing from/to a bitstream, fetching/storing from/into a file, and comparing a signature with another.

### 7.50.2 Constructor & Destructor Documentation

#### 7.50.2.1 mpeg7cdvs::SCFVSignature::SCFVSignature ( bool *hasVar*, bool *hasBitSelection* )

Constructor declaring if this signature contains variance information and bit selection.

##### Parameters

<i>hasVar</i>	true if this signature contains variance information (used normally at high bitrates)
<i>hasBitSelection</i>	true if this signature performs bit selection (used normally at very low bitrates)

### 7.50.3 Member Function Documentation

#### 7.50.3.1 void mpeg7cdvs::SCFVSignature::clear ( )

clear all data

#### 7.50.3.2 int mpeg7cdvs::SCFVSignature::compare ( const [SCFVSignature](#) & *other* ) const

compare two signatures (only for debugging)

7.50.3.3 `int mpeg7cdvs::SCFVSignature::compressedNumBits ( ) const`

get the number of bits of the encoded signature (compressed)

7.50.3.4 `void mpeg7cdvs::SCFVSignature::fromFile ( FILE * file )`

read the signature from file

7.50.3.5 `float mpeg7cdvs::SCFVSignature::getNorm ( ) const`

get the norm of this signature

7.50.3.6 `unsigned int mpeg7cdvs::SCFVSignature::getVisited ( ) const`

get the number of visited words

7.50.3.7 `bool mpeg7cdvs::SCFVSignature::hasBitSelection ( ) const`

tell if this signature performs bit selection

7.50.3.8 `void mpeg7cdvs::SCFVSignature::hasBitSelection ( bool value )`

set this signature as one performing bit selection (if value is true)

7.50.3.9 `bool mpeg7cdvs::SCFVSignature::hasVar ( ) const`

tell if this signature has variance information

7.50.3.10 `void mpeg7cdvs::SCFVSignature::hasVar ( bool value )`

set this signature as one containing variance information (if value is true)

7.50.3.11 `void mpeg7cdvs::SCFVSignature::print ( ) const`

print a summary of the signature data

7.50.3.12 `void mpeg7cdvs::SCFVSignature::read ( BitInputStream & in )`

read the binary signature from the given input stream

7.50.3.13 `void mpeg7cdvs::SCFVSignature::setNorm ( )`

compute and store the correct norm for this signature

7.50.3.14 `void mpeg7cdvs::SCFVSignature::setVisited ( )`

compute and store the correct number of visited words

7.50.3.15 `size_t mpeg7cdvs::SCFVSignature::size ( ) const`

get the size of the binary signature (uncompressed)

7.50.3.16 `void mpeg7cdvs::SCFVSignature::toFile ( FILE * file ) const`

write the signature to file

7.50.3.17 `void mpeg7cdvs::SCFVSignature::write ( BitOutputStream & out ) const`

write the binary signature into the given output stream

## 7.50.4 Field Documentation

7.50.4.1 `unsigned int mpeg7cdvs::SCFVSignature::m_vWordBlock[numberCentroids]`

Scalable Fisher Vector binary signature.

7.50.4.2 `unsigned int mpeg7cdvs::SCFVSignature::m_vWordVarBlock[numberCentroids]`

Scalable Fisher Vector binary variance.

7.50.4.3 `const unsigned int mpeg7cdvs::SCFVSignature::table_bit_selection[]` `[static]`

subset of bits used in low bitrate applications

The documentation for this class was generated from the following file:

- [SCFVIndex.h](#)

## 7.51 TM\_COUNTER Union Reference

Structure used by the high precision timer.

```
#include <timer.h>
```

### Data Fields

- struct {  
    time\_t [tv\\_sec](#)  
    long [tv\\_nsec](#)  
} [tmspec](#)
- struct {  
    time\_t [tv\\_sec](#)  
    long [tv\\_usec](#)  
} [osxtmspec](#)
- long long [counter](#)

### 7.51.1 Detailed Description

Structure used by the high precision timer.

### 7.51.2 Field Documentation

7.51.2.1 `long long TM_COUNTER::counter`

7.51.2.2 `struct { ... } TM_COUNTER::osxtmspec`

7.51.2.3 `struct { ... } TM_COUNTER::tmspec`

7.51.2.4 `long TM_COUNTER::tv_nsec`

7.51.2.5 `time_t TM_COUNTER::tv_sec`

7.51.2.6 `long TM_COUNTER::tv_usec`

The documentation for this union was generated from the following file:

- [timer.h](#)

## 7.52 TM\_STATE Struct Reference

Structure used by the high precision timer.

```
#include <timer.h>
```

### Data Fields

- `long long` [freq](#)

### 7.52.1 Detailed Description

Structure used by the high precision timer.

### 7.52.2 Field Documentation

7.52.2.1 `long long TM_STATE::freq`

The documentation for this struct was generated from the following file:

- [timer.h](#)

## 7.53 TraceManager Class Reference

Helper class to manage trace files in text or XML format.

```
#include <TraceManager.h>
```



## Public Member Functions

- [TraceManager](#) ()
- virtual [~TraceManager](#) ()
- bool [isEnabled](#) () const  
*Indicates if tracing has been enabled.*
- void [openTxt](#) (const char \*fname)  
*Open a text trace file (alternative to openXml).*
- void [openXml](#) (const char \*fname)  
*Open an XML trace file (alternative to openTxt).*
- void [start](#) (const char \*section)  
*Start the given section in the trace file.*
- void [stop](#) (const char \*section)  
*Stop the given section in the trace file.*
- void [matchResults](#) (const [mpeg7cdvs::PointPairs](#) &pairs, [mpeg7cdvs::CDVSPPOINT](#) \*proj\_bbox, int loop-Counter)  
*Trace information on the match results.*
- void [matchPair](#) (const char \*q\_fname, const char \*r\_fname)  
*Trace information on the names of the files that constitute the matching pair.*

### 7.53.1 Detailed Description

Helper class to manage trace files in text or XML format.

#### Author

Massimo Balestri, Alberto Messina

#### Date

2012

### 7.53.2 Constructor & Destructor Documentation

7.53.2.1 [TraceManager::TraceManager](#) ( )

7.53.2.2 virtual [TraceManager::~~TraceManager](#) ( ) [virtual]

### 7.53.3 Member Function Documentation

7.53.3.1 bool [TraceManager::isEnabled](#) ( ) const

Indicates if tracing has been enabled.

#### Returns

true if tracing is enabled; false if no tracing data is currently being written into the trace file.

7.53.3.2 void [TraceManager::matchPair](#) ( const char \* *q\_fname*, const char \* *r\_fname* )

Trace information on the names of the files that constitute the matching pair.

## Parameters

<i>q_fname</i>	query file name
<i>r_fname</i>	reference file name

7.53.3.3 void TraceManager::matchResults ( const mpeg7cdvs::PointPairs & *pairs*, mpeg7cdvs::CDVSPPOINT \* *proj\_bbox*, int *loopCounter* )

Trace information on the match results.

## Parameters

<i>pairs</i>	the number and coordinates of matched features
<i>proj_bbox</i>	the projected bounding box
<i>loopCounter</i>	the index of this match in the list of matches

7.53.3.4 void TraceManager::openTxt ( const char \* *fname* )

Open a text trace file (alternative to openXml).

## Parameters

<i>fname</i>	text trace file name.
--------------	-----------------------

7.53.3.5 void TraceManager::openXml ( const char \* *fname* )

Open an XML trace file (alternative to openTxt).

## Parameters

<i>fname</i>	XML trace file name.
--------------	----------------------

7.53.3.6 void TraceManager::start ( const char \* *section* )

Start the given section in the trace file.

## Parameters

<i>section</i>	the section name.
----------------	-------------------

7.53.3.7 void TraceManager::stop ( const char \* *section* )

Stop the given section in the trace file.

## Parameters

<i>section</i>	the section name.
----------------	-------------------

The documentation for this class was generated from the following file:

- [TraceManager.h](#)

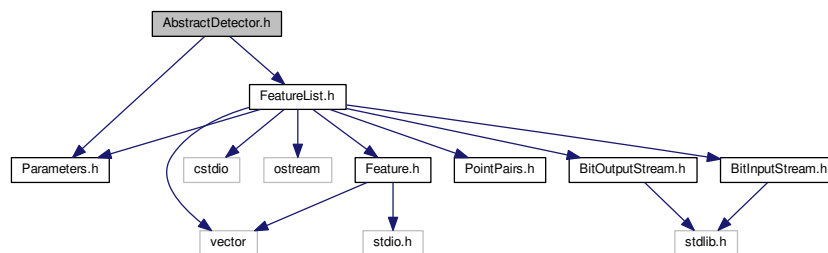
## Chapter 8

# File Documentation

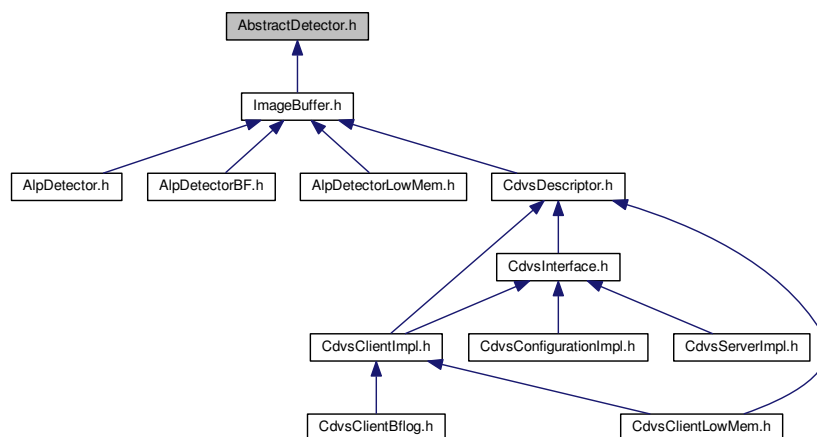
### 8.1 AbstractDetector.h File Reference

```
#include "FeatureList.h"  
#include "Parameters.h"
```

Include dependency graph for AbstractDetector.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::AbstractDetector](#)

*Base class for keypoint detectors.*

## Namespaces

- [mpeg7cdvs](#)

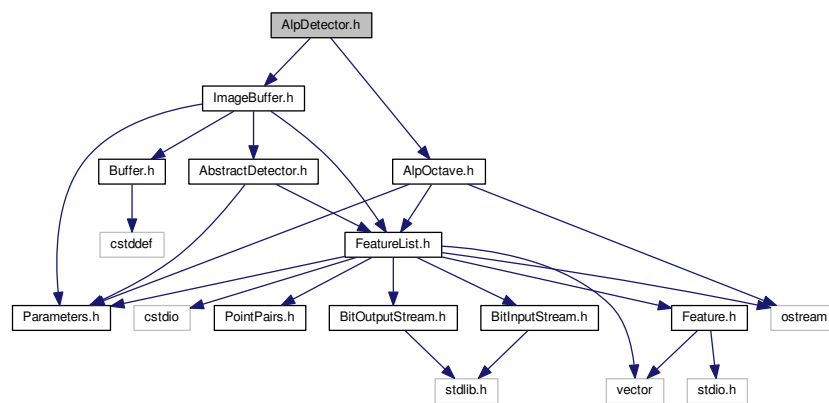
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## 8.2 AlpDetector.h File Reference

```
#include "ImageBuffer.h"
```

```
#include "AlpOctave.h"
```

Include dependency graph for AlpDetector.h:



## Data Structures

- class [mpeg7cdvs::AlpDetector](#)

*Implementation of the ALP keypoint detector.*

## Namespaces

- [mpeg7cdvs](#)

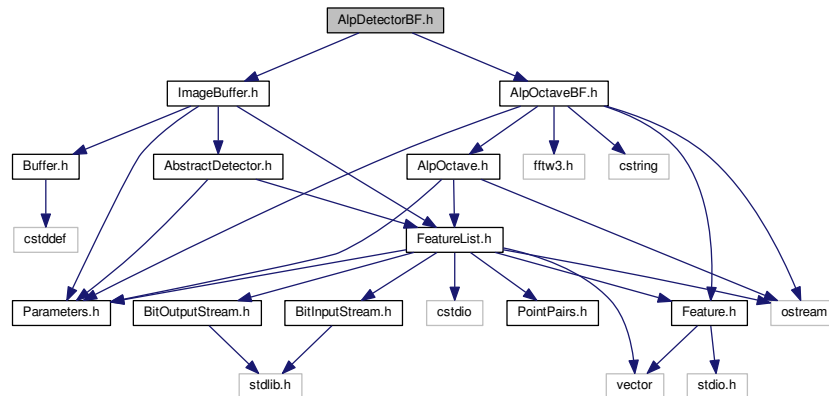
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## 8.3 AlpDetectorBF.h File Reference

```
#include "ImageBuffer.h"
```

```
#include "AlpOctaveBF.h"
```

Include dependency graph for AlpDetectorBF.h:



## Data Structures

- class [mpeg7cdvs::AlpDetectorBF](#)

*implementation of the ALP keypoint detector by employing block-based processing and frequency domain filtering.*

## Namespaces

- [mpeg7cdvs](#)

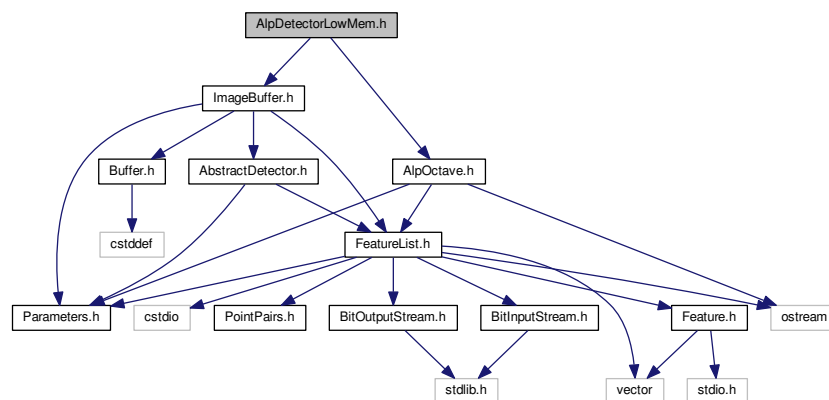
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsiInterface.h](#)) are included.*

## 8.4 AlpDetectorLowMem.h File Reference

```
#include "ImageBuffer.h"
```

```
#include "AlpOctave.h"
```

Include dependency graph for AlpDetectorLowMem.h:



## Data Structures

- class [mpeg7cdvs::AlpDetectorLowMem](#)  
*Low memory implementation of the ALP keypoint detector.*

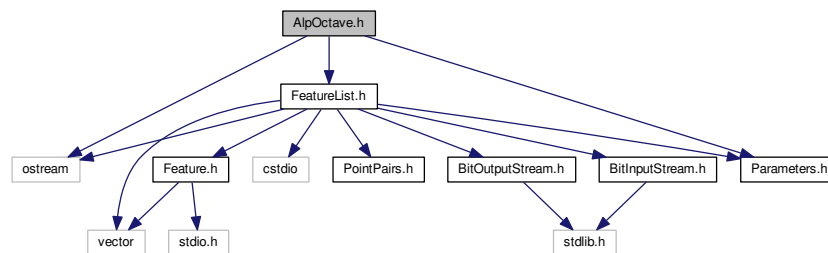
## Namespaces

- [mpeg7cdvs](#)  
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvslInterface.h](#)) are included.*

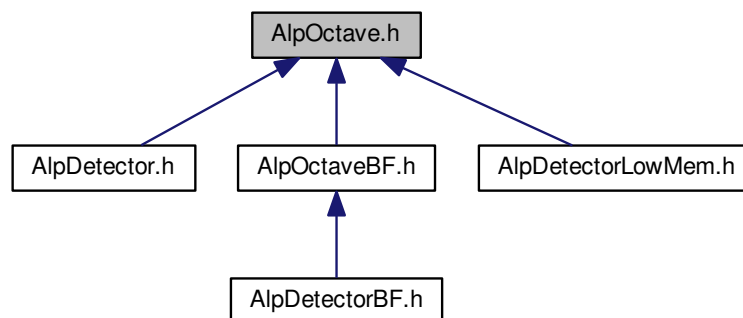
## 8.5 AlpOctave.h File Reference

```
#include "FeatureList.h"
#include "Parameters.h"
#include <ostream>
```

Include dependency graph for AlpOctave.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::FeatureAlp](#)

*Definition of keypoint for Alp.*

- class `mpeg7cdvs::Filter`

*A class containing a separable Gaussian filter kernel.*

- class `mpeg7cdvs::AlpOctave`

*A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.*

## Namespaces

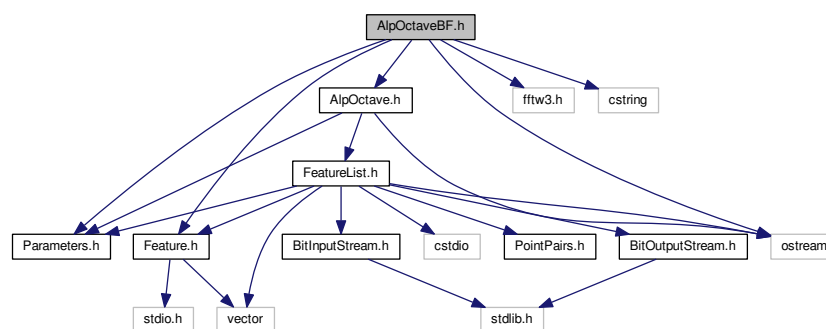
- `mpeg7cdvs`

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

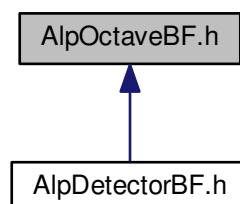
## 8.6 AlpOctaveBF.h File Reference

```
#include "Feature.h"
#include "Parameters.h"
#include <ostream>
#include "AlpOctave.h"
#include "fftw3.h"
#include <cstring>
```

Include dependency graph for AlpOctaveBF.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::FrequencyFilter](#)
- class [mpeg7cdvs::FeatureAlpBF](#)  
*Inherits all member variables declared in [FeatureAlp](#) and adds two new member variables.*
- class [mpeg7cdvs::AlpOctaveBF](#)  
*A container class for a single octave of an image, at a given scale, used to detect and extract ALP key points.*

## Namespaces

- [mpeg7cdvs](#)  
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvslInterface.h](#)) are included.*

## Macros

- `#define PADDING_REPEAT`
- `#define BLOCK_WIDTH 96 /*The Block Size*/`
- `#define MAX_FILTER_WIDTH 33 /*The Maximum Width of the Filters*/`
- `#define PAD_WIDTH 16 /*The pad width*/`
- `#define FBLOCK_WIDTH 128`
- `#define FHWIDTH 65`
- `#define BLOCK_SIZE 9216`
- `#define MODE_LEFT 1`
- `#define MODE_TOP 2`
- `#define MODE_RIGHT 4`
- `#define MODE_BOTTOM 8`
- `#define MODE_FINISH 16`
- `#define LOG_OF_2 0.693147180559945f`
- `#define PI 3.141592653589793f`
- `#define PI2 6.283185307179586f`
- `#define HALF_PI 1.5707963267948965`
- `#define QUTER_PI 0.78539816339744831f`
- `#define THR_QUTER_PI 2.3561944902f`
- `#define BPO_P_PI2 1.273239544735162686151f`
- `#define EPSILON_F 1.19209290E-07F`
- `#define EPSILON_D 2.220446049250313e-16`
- `#define MIN(x, y) (((x)<(y))?(x):(y))`
- `#define MAX(x, y) (((x)>(y))?(x):(y))`
- `#define SHIFT_LEFT(x, n) (((n)>=0)?((x)<<(n)):((x)>>-(n)))`
- `#define FILTER_LOG 0`
- `#define FILTER_GAUSSIAN 1`

## Typedefs

- typedef float [mpeg7cdvs::input\\_type](#)
- typedef float [mpeg7cdvs::gs\\_type](#)
- typedef float [mpeg7cdvs::element\\_type](#)
- typedef float [mpeg7cdvs::filter\\_type](#)



### 8.6.1 Macro Definition Documentation

8.6.1.1 `#define BLOCK_SIZE 9216`

8.6.1.2 `#define BLOCK_WIDTH 96 /*The Block Size*/`

8.6.1.3 `#define BPO_P_PI2 1.273239544735162686151f`

8.6.1.4 `#define EPSILON_D 2.220446049250313e-16`

8.6.1.5 `#define EPSILON_F 1.19209290E-07F`

8.6.1.6 `#define FBLOCK_WIDTH 128`

8.6.1.7 `#define FHWIDTH 65`

8.6.1.8 `#define FILTER_GAUSSIAN 1`

8.6.1.9 `#define FILTER_LOG 0`

8.6.1.10 `#define HALF_PI 1.5707963267948965`

8.6.1.11 `#define LOG_OF_2 0.693147180559945f`

8.6.1.12 `#define MAX( x, y ) (((x)>(y))?(x):(y))`

8.6.1.13 `#define MAX_FILTER_WIDTH 33 /*The Maximum Width of the Filters*/`

8.6.1.14 `#define MIN( x, y ) (((x)<(y))?(x):(y))`

8.6.1.15 `#define MODE_BOTTOM 8`

8.6.1.16 `#define MODE_FINISH 16`

8.6.1.17 `#define MODE_LEFT 1`

8.6.1.18 `#define MODE_RIGHT 4`

8.6.1.19 `#define MODE_TOP 2`

8.6.1.20 `#define PAD_WIDTH 16 /*The pad width*/`

8.6.1.21 `#define PADDING_REPEAT`

8.6.1.22 `#define PI 3.141592653589793f`

8.6.1.23 `#define PI2 6.283185307179586f`

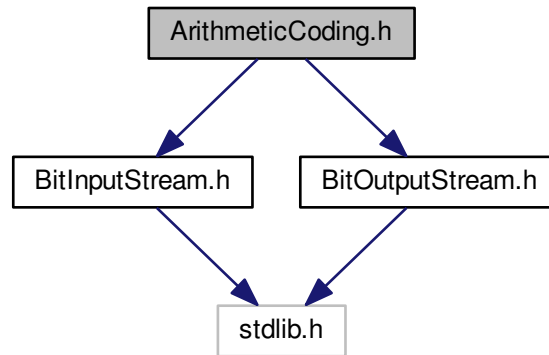
8.6.1.24 `#define QUTER_PI 0.78539816339744831f`

8.6.1.25 `#define SHIFT_LEFT( x, n ) (((n)>=0)?((x)<<(n)):((x)>>-(n)))`

8.6.1.26 `#define THR_QUTER_PI 2.3561944902f`

## 8.7 ArithmeticCoding.h File Reference

```
#include "BitInputStream.h"
#include "BitOutputStream.h"
Include dependency graph for ArithmeticCoding.h:
```



### Data Structures

- class [mpeg7cdvs::AC\\_model](#)  
*Arithmetic Coding model to be used when encoding or decoding a symbol.*
- class [mpeg7cdvs::AC\\_encoder](#)  
*The encoder using the Arithmetic Coding model.*
- class [mpeg7cdvs::AC\\_decoder](#)  
*The decoder using the Arithmetic Coder Model.*

### Namespaces

- [mpeg7cdvs](#)  
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

### Typedefs

- typedef unsigned long long [mpeg7cdvs::CODE\\_VALUE](#)

### Variables

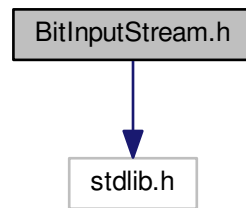
- static const int [mpeg7cdvs::PRECISION](#) = 64
- static const int [mpeg7cdvs::CODE\\_VALUE\\_BITS](#) = 18
- static const int [mpeg7cdvs::FREQUENCY\\_BITS](#) = CODE\_VALUE\_BITS - 2
- static const CODE\_VALUE [mpeg7cdvs::MAX\\_CODE](#) = (1 << CODE\_VALUE\_BITS) - 1
- static const CODE\_VALUE [mpeg7cdvs::MAX\\_FREQ](#) = (1 << FREQUENCY\_BITS) - 1
- static const CODE\_VALUE [mpeg7cdvs::TOP\\_VALUE](#) = MAX\_CODE

- static const CODE\_VALUE [mpeg7cdvs::ONE\\_FOURTH](#) = 1 << (CODE\_VALUE\_BITS - 2)
- static const CODE\_VALUE [mpeg7cdvs::ONE\\_HALF](#) = 2 \* ONE\_FOURTH
- static const CODE\_VALUE [mpeg7cdvs::THREE\\_FOURTHS](#) = 3 \* ONE\_FOURTH

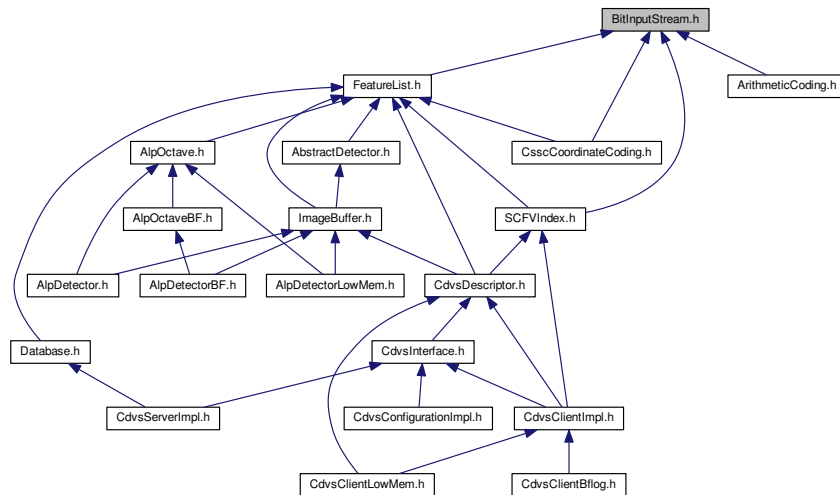
## 8.8 BitInputStream.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for BitInputStream.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- class [mpeg7cdvs::BitInputStream](#)  
This class represents an input stream of bits.

### Namespaces

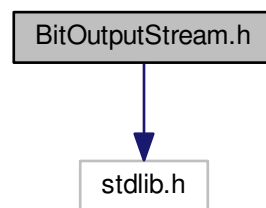
- [mpeg7cdvs](#)

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.

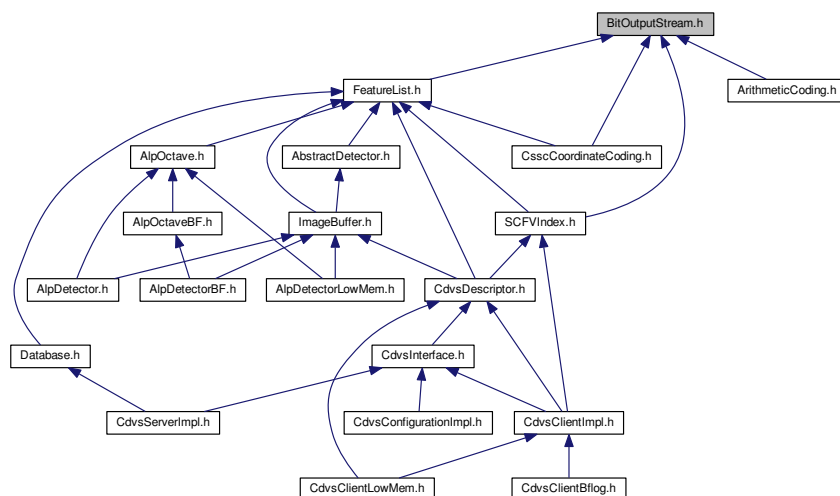
## 8.9 BitOutputStream.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for BitOutputStream.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- class [mpeg7cdvs::BitOutputStream](#)

*This class represents an output stream of bits.*

### Namespaces

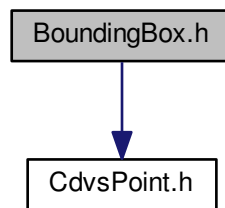
- [mpeg7cdvs](#)

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.

## 8.10 BoundingBox.h File Reference

```
#include "CdvPoint.h"
```

Include dependency graph for BoundingBox.h:



### Data Structures

- class [Polygon](#)

*A base class for all polygons;.*

- class [BoundingBox](#)

*A class containing four points which identify the object in the image.*

### Typedefs

- typedef [mpeg7cdvs::CDVSPPOINT](#) [point](#)
- typedef float [real\\_t](#)

#### 8.10.1 Typedef Documentation

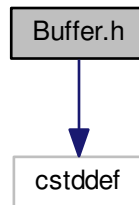
8.10.1.1 typedef [mpeg7cdvs::CDVSPPOINT](#) [point](#)

8.10.1.2 typedef float [real\\_t](#)

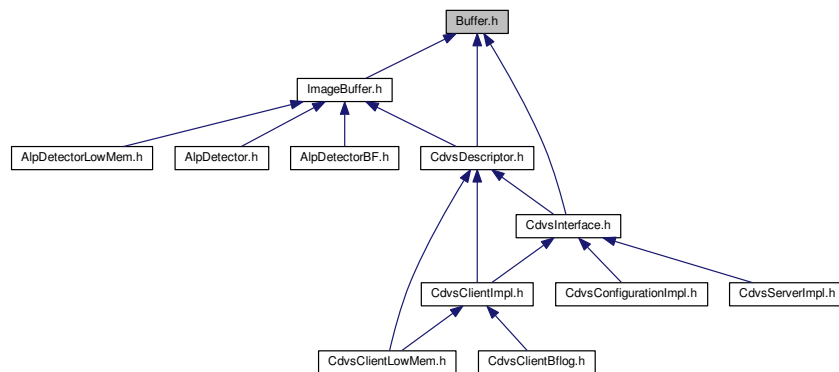
## 8.11 Buffer.h File Reference

```
#include <cstdint>
```

Include dependency graph for Buffer.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::Buffer](#)

*A container class for a byte array, intended to replace all `malloc()` and `new()` instructions in the main code.*

## Namespaces

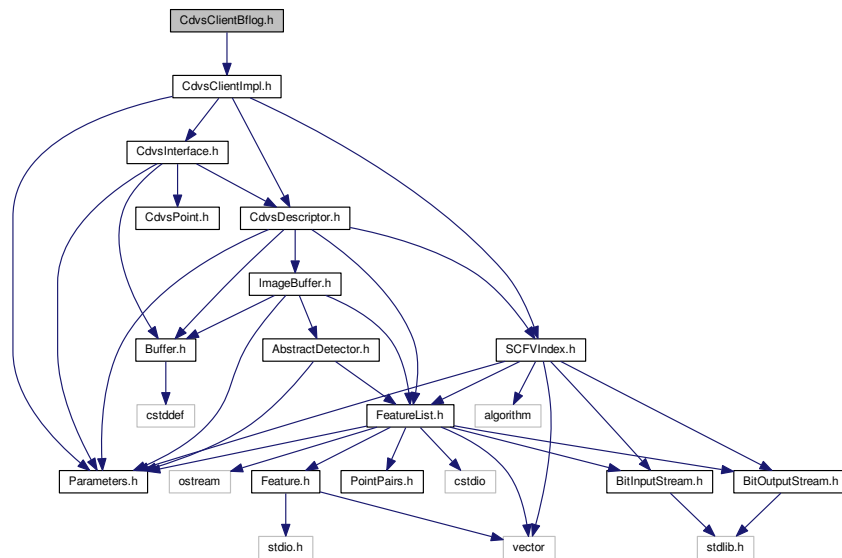
- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## 8.12 CdvClientBflog.h File Reference

```
#include "CdvClientImpl.h"
```

Include dependency graph for CdvsClientBflog.h:



## Data Structures

- class [mpeg7cdvs::CdvsClientBflog](#)

*Block-based frequency domain Laplacian of Gaussian implementation of the high level interface to the client-side functionality of the CDVS Library.*

## Namespaces

- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## Functions

- void [libcdvsbflog\\_is\\_present](#) (void)

### 8.12.1 Function Documentation

8.12.1.1 void [libcdvsbflog\\_is\\_present](#) ( void )

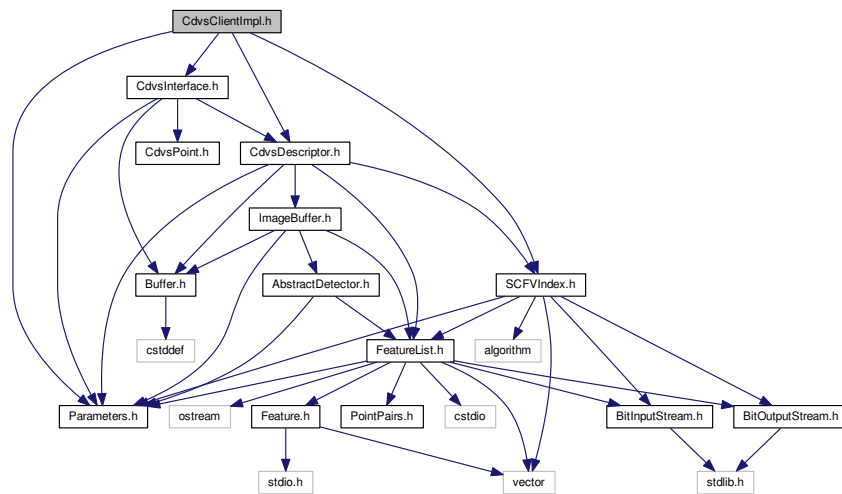
## 8.13 CdvsClientImpl.h File Reference

```

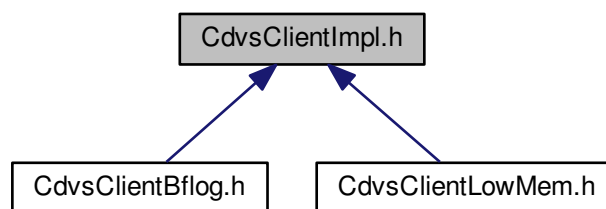
#include "CdvsInterface.h"
#include "Parameters.h"
#include "SCFVIndex.h"
#include "CdvsDescriptor.h"

```

Include dependency graph for CdvsClientImpl.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::CdvsClientImpl](#)

*Main implementation of the high level interface to the client-side functionality of the CDVS Library.*

## Namespaces

- [mpeg7cdvs](#)

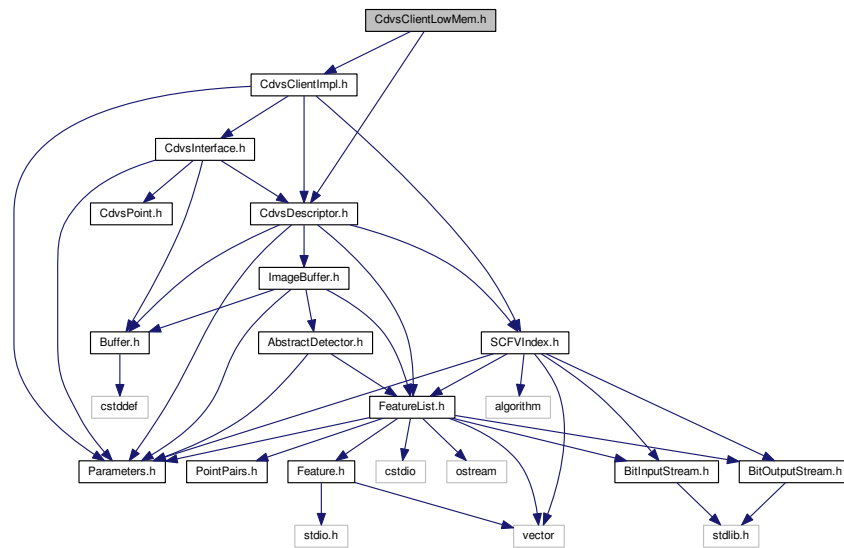
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## 8.14 CdvsClientLowMem.h File Reference

```
#include "CdvsClientImpl.h"
#include "CdvsDescriptor.h"
```



Include dependency graph for CdvsClientLowMem.h:



## Data Structures

- class [mpeg7cdvs::CdvsClientLowMem](#)

*Low Memory implementation of the high level interface to the client-side functionality of the CDVS Library.*

## Namespaces

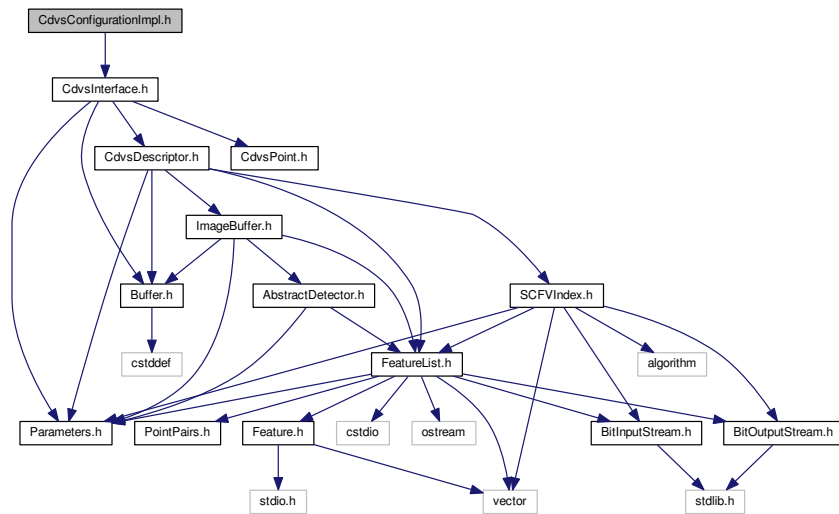
- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## 8.15 CdvsConfigurationImpl.h File Reference

```
#include "CdvsInterface.h"
```

Include dependency graph for `CdvsConfigurationImpl.h`:



## Data Structures

- class [mpeg7cdvs::CdvsConfigurationImpl](#)

*Interface to all configuration parameters for clients and servers.*

## Namespaces

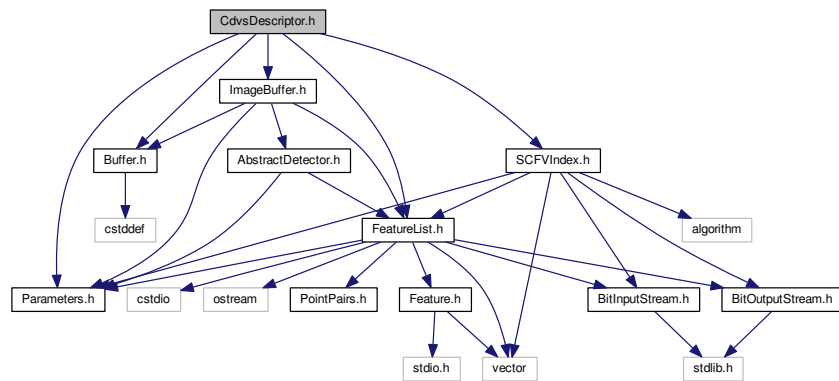
- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

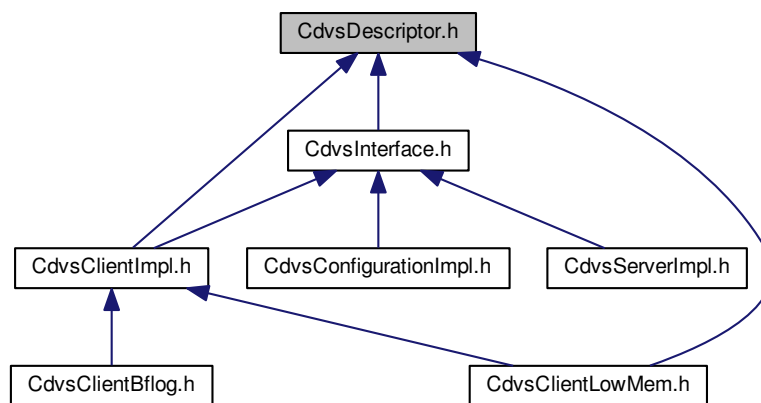
## 8.16 CdvsDescriptor.h File Reference

```
#include "Parameters.h"
#include "Buffer.h"
#include "ImageBuffer.h"
#include "FeatureList.h"
#include "SCFVIndex.h"
```

Include dependency graph for CdvsDescriptor.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::CdvsDescriptor](#)

*Helper class to read/write/check CDVS descriptors according to the syntax defined in ISO/IEC 15938-13.*

## Namespaces

- [mpeg7cdvs](#)

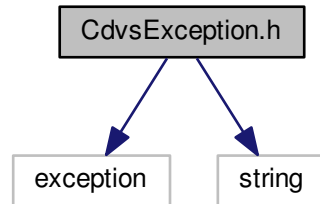
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## 8.17 CdvsException.h File Reference

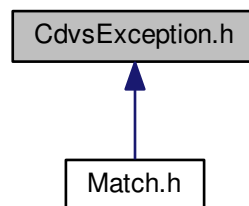
```
#include <exception>
```

```
#include <string>
```

Include dependency graph for CdvsException.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::CdvsException](#)  
*Class defining a specific exception for CDVS.*

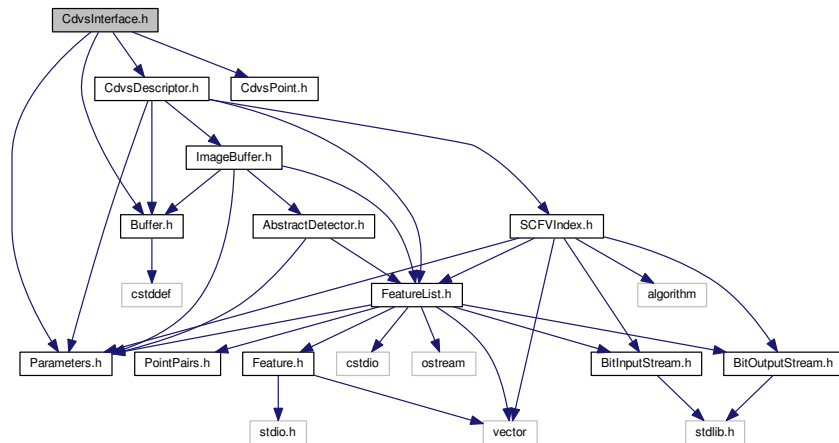
## Namespaces

- [mpeg7cdvs](#)  
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

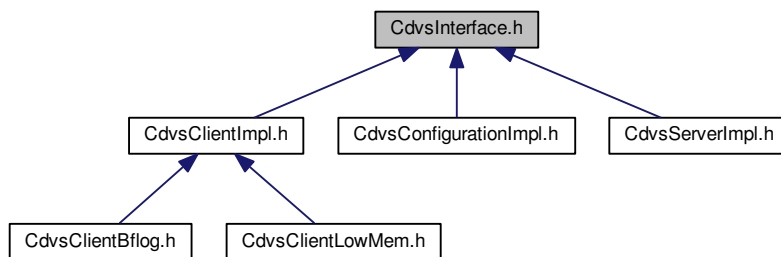
## 8.18 CdvsInterface.h File Reference

```
#include "Parameters.h"  
#include "CdvsDescriptor.h"  
#include "CdvsPoint.h"  
#include "Buffer.h"
```

Include dependency graph for CdvInterface.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

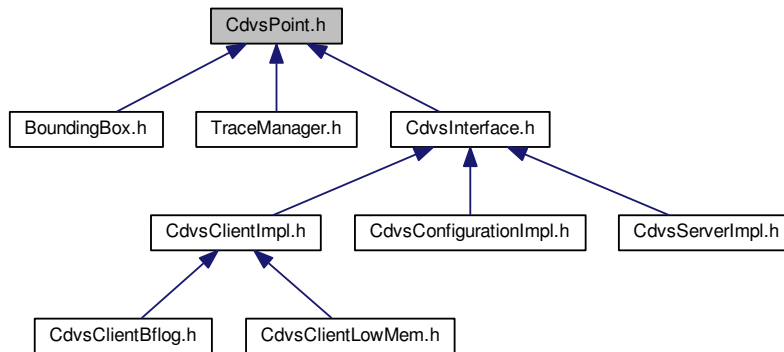
- class `mpeg7cdvs::CdvsConfiguration`  
*Interface to all configuration parameters for clients and servers.*
- class `mpeg7cdvs::CdvsClient`  
*Interface to the client-side functionality of the CDVS Library.*
- class `mpeg7cdvs::CdvsServer`  
*Interface to the server-side functionality of the CDVS Library.*

## Namespaces

- [mpeg7cdvs](#)  
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## 8.19 CdvsPoint.h File Reference

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [mpeg7cdvs::CDVSPPOINT](#)  
A structure containing the x and y coordinate of a point in the image.
- struct [mpeg7cdvs::RetrievalData](#)  
A structure containing the output of a retrieval operation.

### Namespaces

- [mpeg7cdvs](#)  
Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.

### Enumerations

- enum { [mpeg7cdvs::MATCH\\_TYPE\\_DEFAULT](#) = 0, [mpeg7cdvs::MATCH\\_TYPE\\_BOTH](#) = 1, [mpeg7cdvs::MATCH\\_TYPE\\_LOCAL](#) = 2, [mpeg7cdvs::MATCH\\_TYPE\\_GLOBAL](#) = 3 }
- Type of matching.

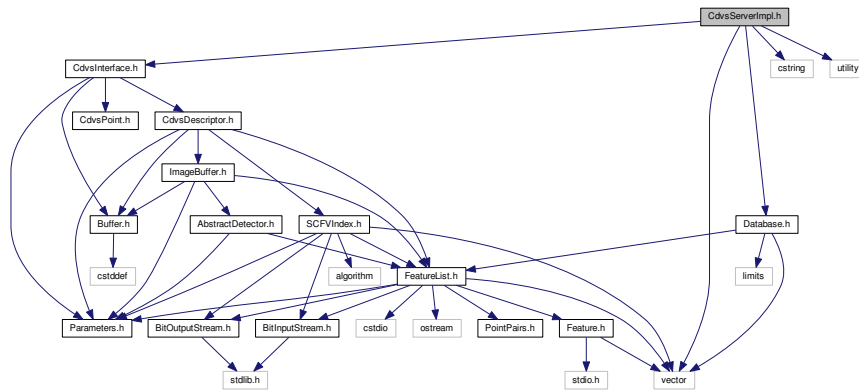
## 8.20 CdvsServerImpl.h File Reference

```

#include "CdvsInterface.h"
#include "Database.h"
#include <cstring>
#include <vector>
#include <utility>

```

Include dependency graph for CdvsServerImpl.h:



## Data Structures

- class [mpeg7cdvs::CdvsServerImpl](#)

*Implementation of the high level interface to the server-side functionality of the CDVS Library.*

## Namespaces

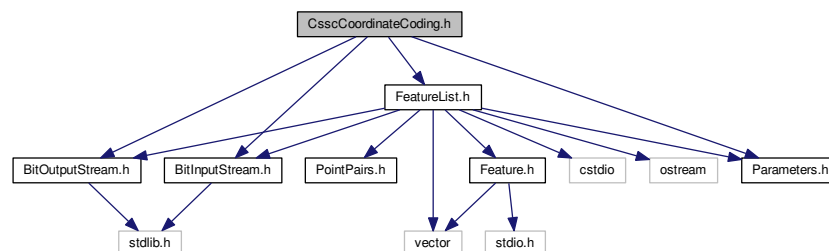
- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## 8.21 CsscCoordinateCoding.h File Reference

```
#include "BitOutputStream.h"
#include "BitInputStream.h"
#include "FeatureList.h"
#include "Parameters.h"
```

Include dependency graph for CsscCoordinateCoding.h:



## Data Structures

- class [mpeg7cdvs::CsscCoordinateCoding](#)

Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.

- struct [mpeg7cdvs::CsscCoordinateCoding::CircularSumContext](#)

Basic structure for Cssc.

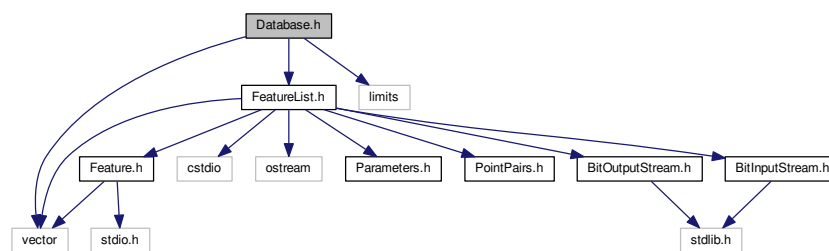
## Namespaces

- [mpeg7cdvs](#)

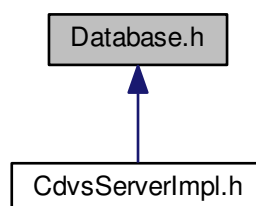
Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.

## 8.22 Database.h File Reference

```
#include "FeatureList.h"
#include <vector>
#include <limits>
Include dependency graph for Database.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::Database](#)

The image database implementation containing helper methods for image retrieval.



## Namespaces

- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvsInterface.h](#)) are included.*

## Typedefs

- typedef std::vector< unsigned int > [mpeg7cdvs::recallGraphNode\\_t](#)
- typedef std::vector  
< recallGraphNode\_t > [mpeg7cdvs::recallGraph\\_t](#)

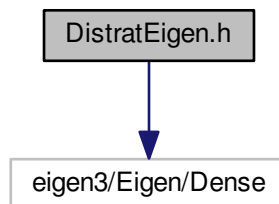
## Variables

- static const size\_t [mpeg7cdvs::NOT\\_FOUND](#) = std::numeric\_limits<size\_t>::max()

## 8.23 DistratEigen.h File Reference

```
#include <eigen3/Eigen/Dense>
```

Include dependency graph for DistratEigen.h:



## Data Structures

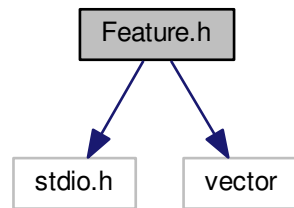
- class [DistratEigen](#)

*Simplified and faster version of DISTRAT, based on the Eigen library.*

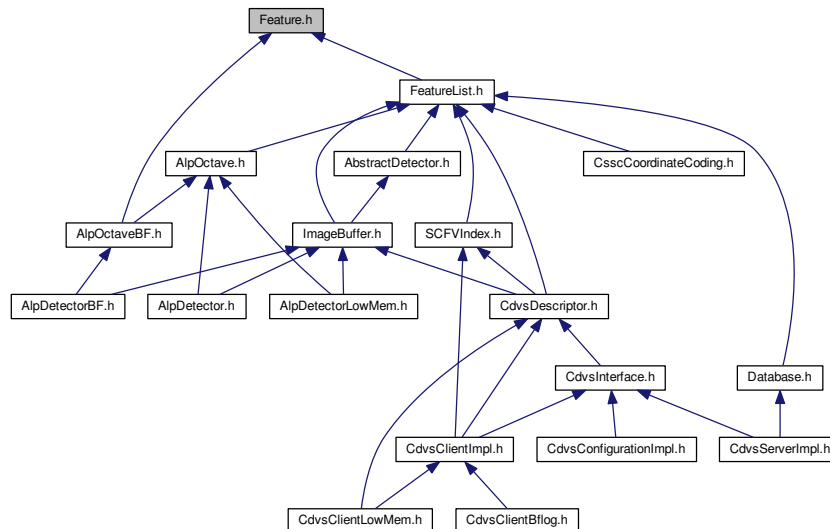
## 8.24 Feature.h File Reference

```
#include <stdio.h>
#include <vector>
```

Include dependency graph for Feature.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::Feature](#)

*Container class for the features of a single point (storing coordinates, scale, orientation, peak and descriptor of a point).*

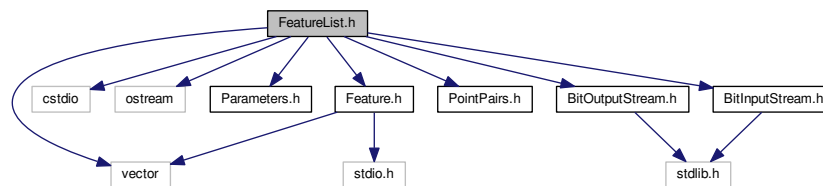
## Namespaces

- [mpeg7cdvs](#)

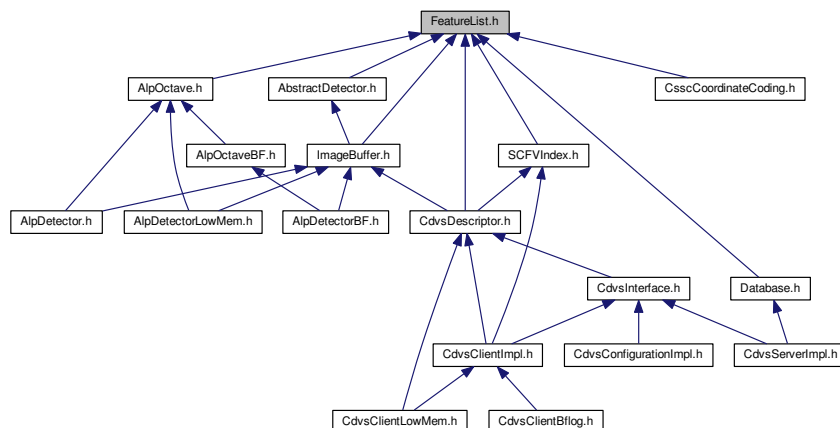
*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## 8.25 FeatureList.h File Reference

```
#include <vector>
#include <cstdio>
#include <ostream>
#include "Parameters.h"
#include "Feature.h"
#include "PointPairs.h"
#include "BitOutputStream.h"
#include "BitInputStream.h"
Include dependency graph for FeatureList.h:
```



This graph shows which files directly or indirectly include this file:



## Data Structures

- class `mpeg7cdvs::FeatureList`  
Container class for all features of an image.
- class `mpeg7cdvs::CompressedFeatureList`  
Container class for all compressed features of an image.

## Namespaces

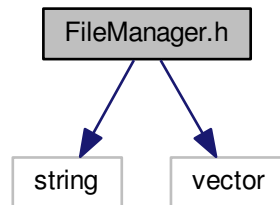
- `mpeg7cdvs`  
Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular `CdvInterface.h`) are included.

## 8.26 FileManager.h File Reference

```
#include <string>
```

```
#include <vector>
```

Include dependency graph for FileManager.h:



### Data Structures

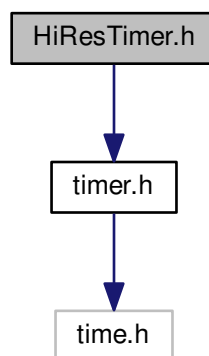
- class [FileManager](#)

*Helper class to manage lists of file names.*

## 8.27 HiResTimer.h File Reference

```
#include "timer.h"
```

Include dependency graph for HiResTimer.h:



### Data Structures

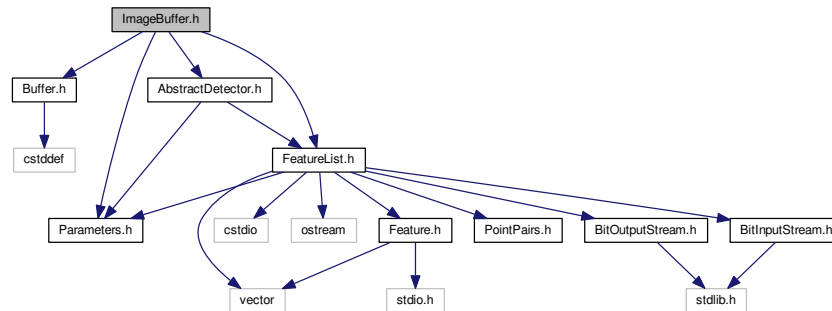
- class [HiResTimer](#)

*C++ wrapper class for the C functions implementing the high resolution timer.*

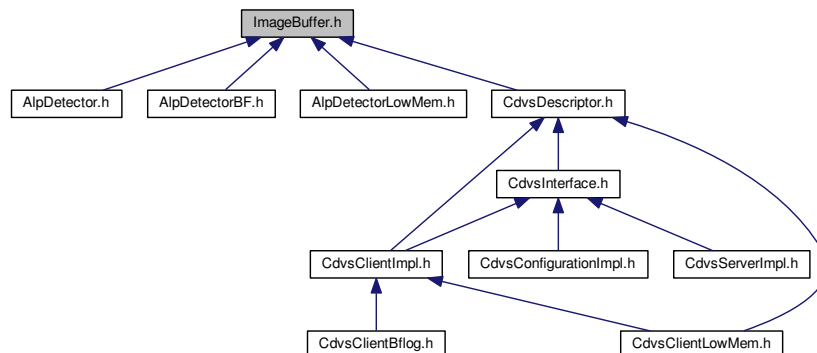
## 8.28 ImageBuffer.h File Reference

```
#include "Buffer.h"
#include "Parameters.h"
#include "FeatureList.h"
#include "AbstractDetector.h"
```

Include dependency graph for ImageBuffer.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- class [mpeg7cdvs::ImageBuffer](#)

*A container class for a bidimensional image; it's the base class of all keypoint detector classes.*

### Namespaces

- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## 8.29 main.main File Reference

## 8.30 map.h File Reference

### Data Structures

- struct [QRA](#)

*Structure containing the name of a query image and the list of names of matching reference pictures.*

### Macros

- #define [MAX\\_MATCHES](#) 500
- #define [MAX\\_FILENAME\\_LENGTH](#) 128

### Typedefs

- typedef char [FILENAME](#) [[MAX\\_FILENAME\\_LENGTH](#)]

### Functions

- double [mean\\_average\\_precision](#) ([QRA](#) \*\*truth, [QRA](#) \*\*results, int n)
- double [success\\_rate\\_for\\_top\\_match](#) ([QRA](#) \*\*truth, [QRA](#) \*\*results, int n)
- double [success\\_rate\\_for\\_top\\_match\\_2](#) ([QRA](#) \*\*truth, [QRA](#) \*\*results, int nt, int nr)
- int [read\\_qra\\_list](#) (char \*file, [QRA](#) \*\*list, int max\_n)
- void [write\\_qra\\_list](#) (char \*file, [QRA](#) \*\*list, int n)
- void [alloc\\_qra\\_list](#) ([QRA](#) \*\*list, int n)
- void [free\\_qra\\_list](#) ([QRA](#) \*\*list, int n)

#### 8.30.1 Macro Definition Documentation

8.30.1.1 #define [MAX\\_FILENAME\\_LENGTH](#) 128

8.30.1.2 #define [MAX\\_MATCHES](#) 500

#### 8.30.2 Typedef Documentation

8.30.2.1 typedef char [FILENAME](#)[[MAX\\_FILENAME\\_LENGTH](#)]

#### 8.30.3 Function Documentation

8.30.3.1 void [alloc\\_qra\\_list](#) ( [QRA](#) \*\* list, int n )

8.30.3.2 void [free\\_qra\\_list](#) ( [QRA](#) \*\* list, int n )

8.30.3.3 double [mean\\_average\\_precision](#) ( [QRA](#) \*\* truth, [QRA](#) \*\* results, int n )

8.30.3.4 int [read\\_qra\\_list](#) ( char \* file, [QRA](#) \*\* list, int max\_n )

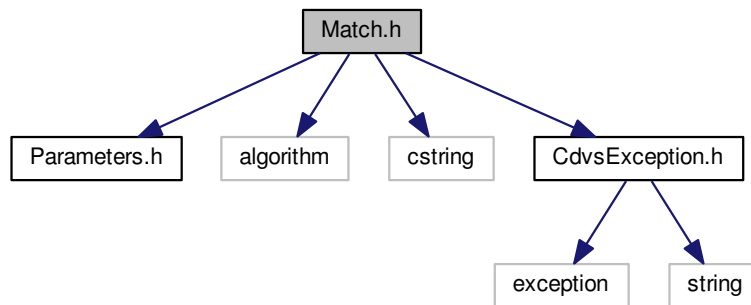
8.30.3.5 double [success\\_rate\\_for\\_top\\_match](#) ( [QRA](#) \*\* truth, [QRA](#) \*\* results, int n )

8.30.3.6 double [success\\_rate\\_for\\_top\\_match\\_2](#) ( [QRA](#) \*\* truth, [QRA](#) \*\* results, int nt, int nr )

8.30.3.7 void [write\\_qra\\_list](#) ( char \* file, [QRA](#) \*\* list, int n )

## 8.31 Match.h File Reference

```
#include "Parameters.h"
#include <algorithm>
#include <cstring>
#include "CdvsException.h"
Include dependency graph for Match.h:
```

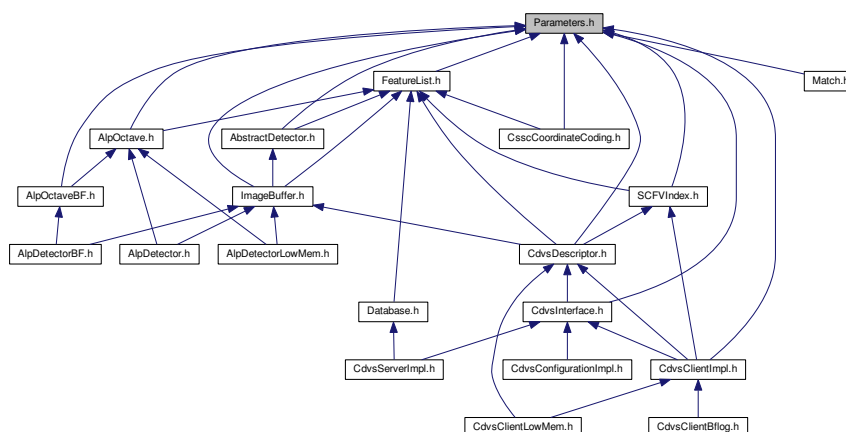


### Data Structures

- class [Match](#)  
Helper class to sort images according to a pair of values.

## 8.32 Parameters.h File Reference

This graph shows which files directly or indirectly include this file:



### Data Structures

- class [mpeg7cdvs::Parameters](#)

Container for all encoding/decoding parameters associated to each target bitrate defined by MPEG CDVS.

## Namespaces

- [mpeg7cdvs](#)

Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.

## Typedefs

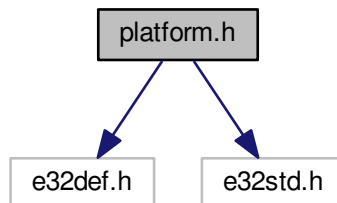
- typedef Parameters [mpeg7cdvs::ParameterSet](#) [Parameters::nModes]

## 8.33 platform.h File Reference

```
#include <e32def.h>
```

```
#include <e32std.h>
```

Include dependency graph for Symbian/platform.h:



## Macros

- #define [assert](#)(x) \_\_ASSERT\_DEBUG(x, User::Exit(255))

### 8.33.1 Macro Definition Documentation

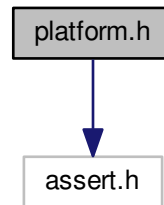
8.33.1.1 #define `assert( x ) __ASSERT_DEBUG(x, User::Exit(255))`



## 8.34 platform.h File Reference

```
#include <assert.h>
```

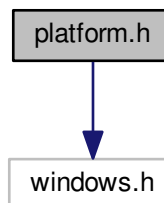
Include dependency graph for w2000/platform.h:



## 8.35 platform.h File Reference

```
#include <windows.h>
```

Include dependency graph for WCE/platform.h:



### Macros

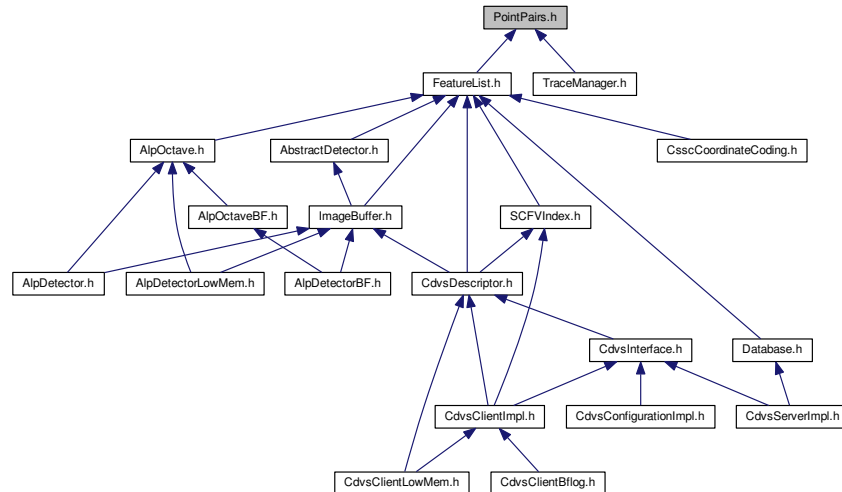
- #define `assert(x)` `ASSERT(x)`

### 8.35.1 Macro Definition Documentation

8.35.1.1 #define `assert( x )` `ASSERT(x)`

## 8.36 PointPairs.h File Reference

This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::PointPairs](#)

*Parameter class, used to pass around matched point coordinates.*

## Namespaces

- [mpeg7cdvs](#)

*Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.*

## Enumerations

- enum { [mpeg7cdvs::match\\_2way\\_INTERSECTION](#) = 0, [mpeg7cdvs::match\\_2way\\_DISJOINT1](#) = 1, [mpeg7cdvs::match\\_2way\\_DISJOINT2](#) = 2 }

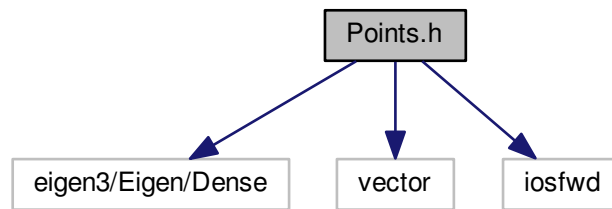
## 8.37 Points.h File Reference

```

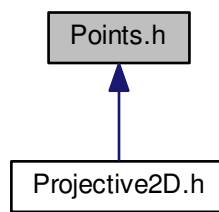
#include <eigen3/Eigen/Dense>
#include <vector>
#include <iosfwd>

```

Include dependency graph for Points.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [Ball2D](#)

## Typedefs

- typedef Eigen::Vector2f [Point2D](#)
- typedef Eigen::Vector3f [HomPoint2D](#)
- typedef Eigen::Matrix< float,  
Eigen::Dynamic,  
2, Eigen::RowMajor > [Point2DArray](#)
- typedef Eigen::Matrix< float,  
Eigen::Dynamic,  
3, Eigen::RowMajor > [HomPoint2DArray](#)

## Functions

- void [read](#) (std::istream &in, [Point2DArray](#) &points)
- std::istream & [operator>>](#) (std::istream &in, [Point2DArray](#) &points)
- [Point2D](#) [baricenter](#) ([Point2DArray](#) const &points)
- [Ball2D](#) [spread](#) ([Point2DArray](#) const &points)

- void `inhomogeneousPoints` (`Point2DArray` &inhPoints, `HomPoint2DArray` const &hPoints)
- `Point2DArray` `inhomogeneousPoints` (`HomPoint2DArray` const &hPoints)
- void `homogeneousPoints` (`HomPoint2DArray` &hPoints, `Point2DArray` const &inhPoints)
- `HomPoint2DArray` `homogeneousPoints` (`Point2DArray` const &inhPoints)
- void `preconditionTransform` (`Eigen::Matrix3f` &transform, `Point2DArray` const &points)
- `Eigen::Matrix3f` `preconditionTransform` (`Point2DArray` const &points)
- void `randomSubset` (`std::vector< int >` &indices, int nPoints, int subsetSize)

### 8.37.1 Typedef Documentation

8.37.1.1 `typedef Eigen::Vector3f HomPoint2D`

8.37.1.2 `typedef Eigen::Matrix<float, Eigen::Dynamic, 3, Eigen::RowMajor> HomPoint2DArray`

8.37.1.3 `typedef Eigen::Vector2f Point2D`

8.37.1.4 `typedef Eigen::Matrix<float, Eigen::Dynamic, 2, Eigen::RowMajor> Point2DArray`

### 8.37.2 Function Documentation

8.37.2.1 `Point2D` `baricenter` ( `Point2DArray` const & *points* ) `[inline]`

8.37.2.2 void `homogeneousPoints` ( `HomPoint2DArray` & *hPoints*, `Point2DArray` const & *inhPoints* )

Referenced by `homogeneousPoints()`.

8.37.2.3 `HomPoint2DArray` `homogeneousPoints` ( `Point2DArray` const & *inhPoints* ) `[inline]`

References `homogeneousPoints()`.

Here is the call graph for this function:



8.37.2.4 void `inhomogeneousPoints` ( `Point2DArray` & *inhPoints*, `HomPoint2DArray` const & *hPoints* )

Referenced by `inhomogeneousPoints()`.

8.37.2.5 `Point2DArray` `inhomogeneousPoints` ( `HomPoint2DArray` const & *hPoints* ) `[inline]`

References `inhomogeneousPoints()`.

Here is the call graph for this function:



8.37.2.6 `std::istream& operator>> ( std::istream & in, Point2DArray & points )`

8.37.2.7 `void preconditionTransform ( Eigen::Matrix3f & transform, Point2DArray const & points )`

Referenced by `preconditionTransform()`.

8.37.2.8 `Eigen::Matrix3f preconditionTransform ( Point2DArray const & points )` `[inline]`

References `preconditionTransform()`.

Here is the call graph for this function:



8.37.2.9 `void randomSubset ( std::vector< int > & indices, int nPoints, int subsetSize )`

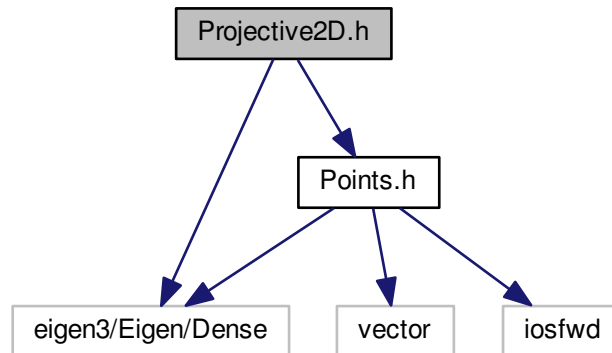
8.37.2.10 `void read ( std::istream & in, Point2DArray & points )`

8.37.2.11 `Ball2D spread ( Point2DArray const & points )`

## 8.38 Projective2D.h File Reference

```
#include "Points.h"
#include <eigen3/Eigen/Dense>
```

Include dependency graph for Projective2D.h:



## Data Structures

- class [Projective2D](#)

*Computes and performs homographies between two images.*

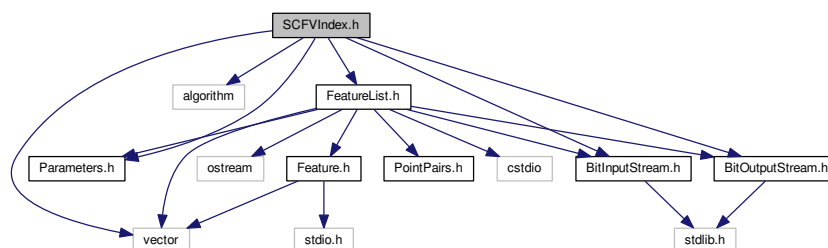
## 8.39 SCFVIndex.h File Reference

```

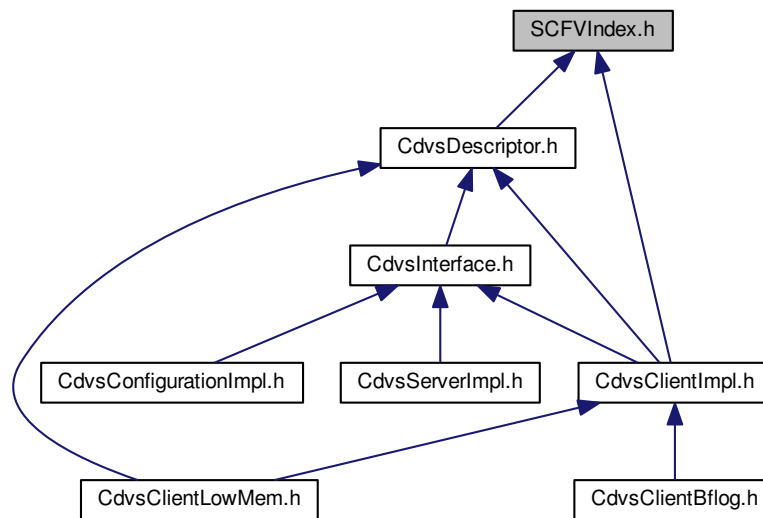
#include <vector>
#include <algorithm>
#include "FeatureList.h"
#include "Parameters.h"
#include "BitOutputStream.h"
#include "BitInputStream.h"

```

Include dependency graph for SCFVIndex.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- class [mpeg7cdvs::LookUpTable](#)  
A simple look up table implementation, to perform a bit count very quickly.
- class [mpeg7cdvs::SCFVSignature](#)  
Container class for a Scalable Fisher Vector binary signature; allows reading/writing from/to a bitstream, fetching/storing from/into a file, and comparing a signature with another.
- class [mpeg7cdvs::SCFVIndex](#)  
A class to manage an indexed list of SCFV signatures.
- class [mpeg7cdvs::SCFVFactory](#)  
A class to produce SCFV signatures.

## Namespaces

- [mpeg7cdvs](#)  
Namespace used to encapsulate all MPEG-7 CDVS declarations that are visible when the CDVS Library headers (in particular [CdvInterface.h](#)) are included.

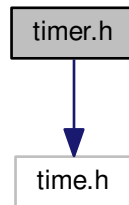
## Variables

- static const float [mpeg7cdvs::gama](#) = 0.3f
- static const int [mpeg7cdvs::num\\_bit\\_selection](#) = 24
- static const int [mpeg7cdvs::PCASiftLength](#) = 32  
number of principal components in the centroid space
- static const int [mpeg7cdvs::numberCentroids](#) = 512  
number of centroids of the codebook

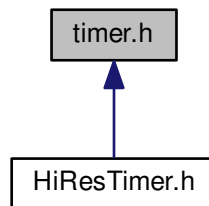
## 8.40 timer.h File Reference

```
#include <time.h>
```

Include dependency graph for timer.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [TM\\_STATE](#)  
*Structure used by the high precision timer.*
- union [TM\\_COUNTER](#)  
*Structure used by the high precision timer.*

### Functions

- void [start\\_timer](#) ([TM\\_STATE](#) \*state, [TM\\_COUNTER](#) \*start)
- void [stop\\_timer](#) ([TM\\_COUNTER](#) \*end)
- double [elapsed\\_time](#) ([TM\\_STATE](#) \*state, [TM\\_COUNTER](#) \*start, [TM\\_COUNTER](#) \*end)

#### 8.40.1 Function Documentation

8.40.1.1 `double elapsed_time ( TM\_STATE * state, TM\_COUNTER * start, TM\_COUNTER * end )`

Referenced by `HiResTimer::elapsed()`.



8.40.1.2 void start\_timer ( TM\_STATE \* state, TM\_COUNTER \* start )

Referenced by HiResTimer::start().

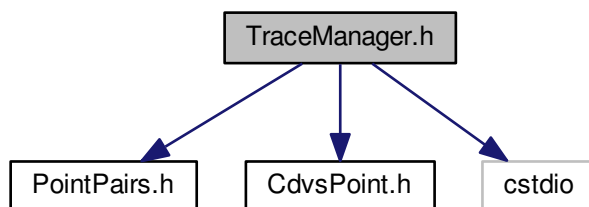
8.40.1.3 void stop\_timer ( TM\_COUNTER \* end )

Referenced by HiResTimer::stop().

## 8.41 TraceManager.h File Reference

```
#include "PointPairs.h"  
#include "CdvsPoint.h"  
#include <cstdio>
```

Include dependency graph for TraceManager.h:



### Data Structures

- class [TraceManager](#)  
*Helper class to manage trace files in text or XML format.*

# Index

- ~AC\_model
  - mpeg7cdvs::AC\_model, [27](#)
- ~AbstractDetector
  - mpeg7cdvs::AbstractDetector, [22](#)
- ~AlpDetector
  - mpeg7cdvs::AlpDetector, [31](#)
- ~AlpOctave
  - mpeg7cdvs::AlpOctave, [37](#)
- ~AlpOctaveBF
  - mpeg7cdvs::AlpOctaveBF, [43](#)
- ~BitInputStream
  - mpeg7cdvs::BitInputStream, [48](#)
- ~BitOutputStream
  - mpeg7cdvs::BitOutputStream, [51](#)
- ~Buffer
  - mpeg7cdvs::Buffer, [57](#)
- ~CdvsClient
  - mpeg7cdvs::CdvsClient, [59](#)
- ~CdvsClientBflog
  - mpeg7cdvs::CdvsClientBflog, [61](#)
- ~CdvsClientImpl
  - mpeg7cdvs::CdvsClientImpl, [63](#)
- ~CdvsClientLowMem
  - mpeg7cdvs::CdvsClientLowMem, [65](#)
- ~CdvsConfiguration
  - mpeg7cdvs::CdvsConfiguration, [67](#)
- ~CdvsConfigurationImpl
  - mpeg7cdvs::CdvsConfigurationImpl, [70](#)
- ~CdvsDescriptor
  - mpeg7cdvs::CdvsDescriptor, [73](#)
- ~CdvsException
  - mpeg7cdvs::CdvsException, [78](#)
- ~CdvsServer
  - mpeg7cdvs::CdvsServer, [80](#)
- ~CdvsServerImpl
  - mpeg7cdvs::CdvsServerImpl, [86](#)
- ~CompressedFeatureList
  - mpeg7cdvs::CompressedFeatureList, [93](#)
- ~CsscCoordinateCoding
  - mpeg7cdvs::CsscCoordinateCoding, [99](#)
- ~Database
  - mpeg7cdvs::Database, [103](#)
- ~DistratEigen
  - DistratEigen, [108](#)
- ~FileManager
  - FileManager, [125](#)
- ~FrequencyFilter
  - mpeg7cdvs::FrequencyFilter, [129](#)
- ~ImageBuffer
  - mpeg7cdvs::ImageBuffer, [134](#)
- ~Parameters
  - mpeg7cdvs::Parameters, [142](#)
- ~PointPairs
  - mpeg7cdvs::PointPairs, [147](#)
- ~TraceManager
  - TraceManager, [165](#)
- AC\_model
  - mpeg7cdvs::AC\_model, [27](#)
- AbstractDetector
  - mpeg7cdvs::AbstractDetector, [22](#)
- AbstractDetector.h, [167](#)
- adaptative
  - mpeg7cdvs::AC\_model, [27](#)
- addDescriptorToDB
  - mpeg7cdvs::CdvsServer, [80](#)
  - mpeg7cdvs::CdvsServerImpl, [86](#)
- addFeature
  - mpeg7cdvs::FeatureList, [119](#)
- addImage
  - mpeg7cdvs::Database, [103](#)
- AddImageSample
  - mpeg7cdvs::CsscCoordinateCoding, [99](#)
- addPair
  - mpeg7cdvs::PointPairs, [147](#)
- align
  - mpeg7cdvs::BitInputStream, [48](#)
  - mpeg7cdvs::BitOutputStream, [51](#)
- alloc\_qra\_list
  - map.h, [194](#)
- Allocate
  - mpeg7cdvs::AlpOctaveBF, [43](#)
- AlpDetector
  - mpeg7cdvs::AlpDetector, [31](#)
- AlpDetector.h, [168](#)
- AlpDetectorBF
  - mpeg7cdvs::AlpDetectorBF, [33](#)
- AlpDetectorBF.h, [168](#)
- AlpDetectorLowMem
  - mpeg7cdvs::AlpDetectorLowMem, [35](#)
- AlpDetectorLowMem.h, [169](#)
- AlpOctave
  - mpeg7cdvs::AlpOctave, [37](#)
- AlpOctave.h, [170](#)
- AlpOctaveBF
  - mpeg7cdvs::AlpOctaveBF, [43](#)
- AlpOctaveBF.h, [171](#)
- BLOCK\_SIZE, [173](#)
- BLOCK\_WIDTH, [173](#)

- BPO\_P\_PI2, 173
- EPSILON\_D, 173
- EPSILON\_F, 173
- FBLOCK\_WIDTH, 173
- FHWIDTH, 173
- FILTER\_GAUSSIAN, 173
- FILTER\_LOG, 173
- HALF\_PI, 173
- LOG\_OF\_2, 173
- MAX, 173
- MAX\_FILTER\_WIDTH, 173
- MIN, 173
- MODE\_BOTTOM, 173
- MODE\_FINISH, 173
- MODE\_LEFT, 173
- MODE\_RIGHT, 173
- MODE\_TOP, 173
- PAD\_WIDTH, 173
- PADDING\_REPEAT, 173
- PI, 173
- PI2, 173
- QUTER\_PI, 173
- SHIFT\_LEFT, 173
- THR\_QUTER\_PI, 173
- append
  - mpeg7cdvs::SCFVIndex, 157
- ArithmeticCoding.h, 174
- assert
  - Symbian/platform.h, 196
  - WCE/platform.h, 197
- assign
  - mpeg7cdvs::Buffer, 57
- available
  - mpeg7cdvs::BitInputStream, 48
  - mpeg7cdvs::BitOutputStream, 52
- BLOCK\_SIZE
  - AlpOctaveBF.h, 173
- BLOCK\_WIDTH
  - AlpOctaveBF.h, 173
- BPO\_P\_PI2
  - AlpOctaveBF.h, 173
- Ball2D, 46
  - Ball2D, 46
  - Ball2D, 46
  - center, 46
  - radius, 46
- baricenter
  - Points.h, 200
- BitInputStream
  - mpeg7cdvs::BitInputStream, 47
- BitInputStream.h, 175
- BitOutputStream
  - mpeg7cdvs::BitOutputStream, 51
- BitOutputStream.h, 176
- bits
  - mpeg7cdvs::AC\_decoder, 23
  - mpeg7cdvs::AC\_encoder, 26
- block\_buffer\_len
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_col\_n
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_col\_remain\_height
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_height
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_id
  - mpeg7cdvs::FeatureAlpBF, 116
- block\_map
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_row\_n
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_row\_remain\_width
  - mpeg7cdvs::AlpOctaveBF, 45
- block\_width
  - mpeg7cdvs::AlpOctaveBF, 45
- blockWidth
  - mpeg7cdvs::Parameters, 143
- BoundingBox, 54
  - BoundingBox, 54
  - BoundingBox, 54
  - cfind\_overlap, 54
  - find\_overlap, 55
  - getAddress, 55
  - read, 55
- BoundingBox.h, 177
  - point, 177
  - real\_t, 177
- Buffer
  - mpeg7cdvs::Buffer, 57
- buffer
  - mpeg7cdvs::CdvsDescriptor, 76
  - mpeg7cdvs::ImageBuffer, 137
- Buffer.h, 177
- CODE\_VALUE
  - mpeg7cdvs, 17
- CODE\_VALUE\_BITS
  - mpeg7cdvs, 18
- CONTEXT\_RANGE
  - mpeg7cdvs::CsscCoordinateCoding, 101
- CdvsClientBflog
  - mpeg7cdvs::CdvsClientBflog, 61
- CdvsClientBflog.h, 178
  - libcdvsbflog\_is\_present, 179
- cdvsClientFactory
  - mpeg7cdvs::CdvsClient, 59
- CdvsClientImpl
  - mpeg7cdvs::CdvsClientImpl, 63
- CdvsClientImpl.h, 179
- CdvsClientLowMem
  - mpeg7cdvs::CdvsClientLowMem, 65
- CdvsClientLowMem.h, 180
- cdvsConfigurationFactory
  - mpeg7cdvs::CdvsConfiguration, 67
- CdvsConfigurationImpl
  - mpeg7cdvs::CdvsConfigurationImpl, 70
- CdvsConfigurationImpl.h, 181

- CdvsDescriptor
  - mpeg7cdvs::CdvsDescriptor, 73
- CdvsDescriptor.h, 182
- CdvsException
  - mpeg7cdvs::CdvsException, 77
- CdvsException.h, 183
- CdvsInterface.h, 184
- CdvsPoint.h, 186
- cdvsServerFactory
  - mpeg7cdvs::CdvsServer, 80
- CdvsServerImpl
  - mpeg7cdvs::CdvsServerImpl, 86
- CdvsServerImpl.h, 186
- center
  - Ball2D, 46
- cfind\_overlap
  - BoundingBox, 54
- check
  - mpeg7cdvs::AC\_model, 27
  - mpeg7cdvs::CdvsDescriptor, 73
- chiSquarePercentile
  - mpeg7cdvs::Parameters, 143
- Clear
  - mpeg7cdvs::AlpOctaveBF, 43
- clear
  - mpeg7cdvs::AlpOctave, 38
  - mpeg7cdvs::Buffer, 57
  - mpeg7cdvs::CdvsDescriptor, 73
  - mpeg7cdvs::Database, 104
  - mpeg7cdvs::FeatureList, 119
  - mpeg7cdvs::SCFVIndex, 157
  - mpeg7cdvs::SCFVSignature, 161
- clearDB
  - mpeg7cdvs::CdvsServer, 81
  - mpeg7cdvs::CdvsServerImpl, 87
- close
  - mpeg7cdvs::BitInputStream, 48
  - mpeg7cdvs::BitOutputStream, 52
- close\_lib
  - mpeg7cdvs::FrequencyFilter, 129
- commitDB
  - mpeg7cdvs::CdvsServer, 81
  - mpeg7cdvs::CdvsServerImpl, 87
- compare
  - mpeg7cdvs::Buffer, 57
  - mpeg7cdvs::CsscCoordinateCoding, 99
  - mpeg7cdvs::SCFVSignature, 161
- compareCoordinates
  - mpeg7cdvs::FeatureList, 119
- compareDescriptors
  - mpeg7cdvs::FeatureList, 119
- compareKeypoints
  - mpeg7cdvs::FeatureList, 119
- compress
  - mpeg7cdvs::FeatureList, 121
- CompressedFeatureList
  - mpeg7cdvs::CompressedFeatureList, 93
  - mpeg7cdvs::FeatureList, 123
- compressedNumBits
  - mpeg7cdvs::SCFVSignature, 161
- computeDescriptor
  - mpeg7cdvs::AlpOctave, 38
  - mpeg7cdvs::AlpOctaveBF, 43
- computeDescriptor\_fast\_mode
  - mpeg7cdvs::AlpOctave, 38
- computeMaxPoints
  - mpeg7cdvs::FeatureList, 121
- consumed
  - mpeg7cdvs::BitInputStream, 48
- Convolution
  - mpeg7cdvs::FrequencyFilter, 129
- copyImageName
  - mpeg7cdvs::Database, 104
- count
  - mpeg7cdvs::AC\_model, 28
- counter
  - TM\_COUNTER, 164
- createDB
  - mpeg7cdvs::CdvsServer, 81
  - mpeg7cdvs::CdvsServerImpl, 87
- CsscCoordinateCoding
  - mpeg7cdvs::CsscCoordinateCoding, 99
- CsscCoordinateCoding.h, 187
- ctxTableIdx
  - mpeg7cdvs::Parameters, 143
- cur\_block\_id
  - mpeg7cdvs::AlpOctaveBF, 45
- curvRatio
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 114
- curvSigma
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 114
- data
  - mpeg7cdvs::Buffer, 57
- Database
  - mpeg7cdvs::Database, 103
- Database.h, 188
- debugLevel
  - mpeg7cdvs::Parameters, 143
- decode
  - mpeg7cdvs::CdvsDescriptor, 73
  - mpeg7cdvs::CdvsServer, 81
  - mpeg7cdvs::CdvsServerImpl, 87
- decode\_symbol
  - mpeg7cdvs::AC\_decoder, 23
- descLength
  - mpeg7cdvs::Parameters, 143
- descr
  - mpeg7cdvs::Feature, 111
- descrBytes
  - mpeg7cdvs::CompressedFeatureList, 93
- descrLength
  - mpeg7cdvs::Feature, 111
- detect
  - mpeg7cdvs::AbstractDetector, 22

- mpeg7cdvs::AlpDetector, 31
- mpeg7cdvs::AlpDetectorBF, 33
- mpeg7cdvs::AlpDetectorLowMem, 35
- mpeg7cdvs::AlpOctave, 38
- mpeg7cdvs::AlpOctaveBF, 43
- detectDuplicates
  - mpeg7cdvs::AlpOctave, 38
  - mpeg7cdvs::AlpOctaveBF, 44
- distL1
  - mpeg7cdvs::FeatureAlp, 114
- DistratEigen, 108
  - ~DistratEigen, 108
  - DistratEigen, 108
  - DistratEigen, 108
  - estimateInliers, 109
  - m\_GoFThreshold, 109
  - m\_bFitsGood, 109
  - m\_c, 109
- DistratEigen.h, 189
- done
  - mpeg7cdvs::AC\_decoder, 25
  - mpeg7cdvs::AC\_encoder, 26
  - mpeg7cdvs::AC\_model, 28
- EPSILON\_D
  - AlpOctaveBF.h, 173
- EPSILON\_F
  - AlpOctaveBF.h, 173
- elapsed
  - HiResTimer, 130
- elapsed\_time
  - timer.h, 204
- element\_type
  - mpeg7cdvs, 17
- empty
  - mpeg7cdvs::AlpOctave, 39
  - mpeg7cdvs::AlpOctaveBF, 44
  - mpeg7cdvs::Buffer, 57
- encode
  - mpeg7cdvs::CdvsClient, 60
  - mpeg7cdvs::CdvsClientBflog, 61
  - mpeg7cdvs::CdvsClientImpl, 63
  - mpeg7cdvs::CdvsClientLowMem, 65
  - mpeg7cdvs::CdvsDescriptor, 73
- encode\_symbol
  - mpeg7cdvs::AC\_encoder, 26
- EndTrainingMode
  - mpeg7cdvs::CsscCoordinateCoding, 99
- eof
  - mpeg7cdvs::BitInputStream, 48
  - mpeg7cdvs::BitOutputStream, 52
- equals
  - mpeg7cdvs::Buffer, 57
- estimateInliers
  - DistratEigen, 109
- exportVars
  - mpeg7cdvs::CsscCoordinateCoding, 100
- extract
  - mpeg7cdvs::AbstractDetector, 22
  - mpeg7cdvs::AlpDetector, 31
  - mpeg7cdvs::AlpDetectorBF, 34
  - mpeg7cdvs::AlpDetectorLowMem, 36
- extrema\_h
  - mpeg7cdvs::AlpOctaveBF, 45
- extrema\_sx
  - mpeg7cdvs::AlpOctaveBF, 45
- extrema\_sy
  - mpeg7cdvs::AlpOctaveBF, 45
- extrema\_w
  - mpeg7cdvs::AlpOctaveBF, 45
- f
  - mpeg7cdvs::LookUpTable, 138
- FBLOCK\_WIDTH
  - AlpOctaveBF.h, 173
- FHWIDTH
  - AlpOctaveBF.h, 173
- FILENAME
  - map.h, 194
- FILTER\_GAUSSIAN
  - AlpOctaveBF.h, 173
- FILTER\_LOG
  - AlpOctaveBF.h, 173
- FREQUENCY\_BITS
  - mpeg7cdvs, 18
- fScore
  - mpeg7cdvs::RetrievalData, 154
- fastInterpolate
  - mpeg7cdvs::ImageBuffer, 134
- fastScalarQuantize
  - mpeg7cdvs::ImageBuffer, 134
- fblock
  - mpeg7cdvs::FrequencyFilter, 129
- fblock\_flog
  - mpeg7cdvs::FrequencyFilter, 129
- Feature
  - mpeg7cdvs::Feature, 111
- Feature.h, 189
- featureInd
  - Match, 139
- FeatureList
  - mpeg7cdvs::FeatureList, 119
- FeatureList.h, 191
- featurelist
  - mpeg7cdvs::CdvsDescriptor, 76
- features
  - mpeg7cdvs::CompressedFeatureList, 97
  - mpeg7cdvs::FeatureList, 124
- fftplan
  - mpeg7cdvs::FrequencyFilter, 129
- FileManager, 124
  - ~FileManager, 125
  - FileManager, 125
  - FileManager, 125
  - getAbsolutePathname, 125
  - getQueryName, 125
  - getReferenceName, 126
  - getRelativePathname, 126

- readAnnotation, 126
- replaceExt, 126
- setAnnotationPath, 127
- setDatasetPath, 127
- FileManager.h, 192
- fill
  - mpeg7cdvs::Buffer, 58
- Filter
  - mpeg7cdvs::Filter, 128
- filter\_type
  - mpeg7cdvs, 17
- filters
  - mpeg7cdvs::AlpOctaveBF, 45
- find
  - mpeg7cdvs::Database, 104
- find\_overlap
  - BoundingBox, 55
- flush
  - mpeg7cdvs::BitOutputStream, 52
- free\_qra\_list
  - map.h, 194
- freq
  - TM\_STATE, 164
- FrequencyFilter
  - mpeg7cdvs::FrequencyFilter, 129
- FrequencyFiltering
  - mpeg7cdvs::AlpOctaveBF, 44
- fromBinary
  - mpeg7cdvs::CsscCoordinateCoding, 101
  - mpeg7cdvs::FeatureList, 121
- fromFile
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureList, 121
  - mpeg7cdvs::SCFVSignature, 162
- g\_factory
  - mpeg7cdvs::CdvsClientImpl, 64
- gScore
  - mpeg7cdvs::RetrievalData, 154
- gama
  - mpeg7cdvs, 18
- gdThreshold
  - mpeg7cdvs::Parameters, 143
- gdThresholdMixed
  - mpeg7cdvs::Parameters, 143
- generateFeatureList
  - mpeg7cdvs::CsscCoordinateCoding, 101
- generateHistogramMap
  - mpeg7cdvs::CsscCoordinateCoding, 101
- generateSCFV
  - mpeg7cdvs::SCFVFactory, 155
- generateWeight
  - mpeg7cdvs::SCFVIndex, 157
- getAbsolutePathname
  - FileManager, 125
- getAddress
  - BoundingBox, 55
- getAlpKeypoints
  - mpeg7cdvs::AlpOctave, 39
- getBlockInput
  - mpeg7cdvs::AlpOctaveBF, 44
- getDistance
  - mpeg7cdvs::CompressedFeatureList, 93
- getGlobalHasBitSelection
  - mpeg7cdvs::CdvsDescriptor, 74
- getGlobalHasVariance
  - mpeg7cdvs::CdvsDescriptor, 74
- getHeight
  - mpeg7cdvs::AlpOctave, 39
  - mpeg7cdvs::AlpOctaveBF, 44
- getHistogramCountSize
  - mpeg7cdvs::CdvsDescriptor, 74
- getHistogramMapSizeX
  - mpeg7cdvs::CdvsDescriptor, 74
- getHistogramMapSizeY
  - mpeg7cdvs::CdvsDescriptor, 74
- getImage
  - mpeg7cdvs::SCFVIndex, 158
- getImageId
  - mpeg7cdvs::CdvsServer, 82
  - mpeg7cdvs::CdvsServerImpl, 88
- getImageName
  - mpeg7cdvs::Database, 104
- getInlierWeight
  - mpeg7cdvs::PointPairs, 148
- getKeypoints
  - mpeg7cdvs::AlpOctave, 39
- getKeypoints\_fast\_mode
  - mpeg7cdvs::AlpOctave, 39
- getMode
  - mpeg7cdvs::CdvsConfiguration, 67
  - mpeg7cdvs::Database, 104
- getModelID
  - mpeg7cdvs::CdvsDescriptor, 74
  - mpeg7cdvs::Parameters, 142
- getNorm
  - mpeg7cdvs::SCFVSignature, 162
- getNumberOfLocalDescriptors
  - mpeg7cdvs::CdvsDescriptor, 74
- getOctave
  - mpeg7cdvs::AlpOctave, 40
  - mpeg7cdvs::AlpOctaveBF, 44
- getOriginalImageXResolution
  - mpeg7cdvs::CdvsDescriptor, 74
- getOriginalImageYResolution
  - mpeg7cdvs::CdvsDescriptor, 74
- getParameters
  - mpeg7cdvs::CdvsConfiguration, 67
  - mpeg7cdvs::CdvsConfigurationImpl, 70
- getPointer
  - mpeg7cdvs::BitInputStream, 48
  - mpeg7cdvs::BitOutputStream, 52
- getQueryName
  - FileManager, 125
- getRecallGraph
  - mpeg7cdvs::Database, 104
- getReferenceName

- FileManager, 126
- getRelativePathname
  - FileManager, 126
- getRelevanceBitsPresent
  - mpeg7cdvs::CdvsDescriptor, 75
- getRelevantPoints
  - mpeg7cdvs::FeatureList, 122
- getSize
  - mpeg7cdvs::BitInputStream, 48
  - mpeg7cdvs::BitOutputStream, 52
- getSymbol
  - mpeg7cdvs::AC\_model, 28
- getTotalWeight
  - mpeg7cdvs::PointPairs, 148
- getVersionID
  - mpeg7cdvs::CdvsDescriptor, 75
- getVisited
  - mpeg7cdvs::SCFVSignature, 162
- getWidth
  - mpeg7cdvs::AlpOctave, 40
  - mpeg7cdvs::AlpOctaveBF, 44
- global\_score
  - mpeg7cdvs::PointPairs, 148
- global\_threshold
  - mpeg7cdvs::PointPairs, 148
- gs\_type
  - mpeg7cdvs, 17
- h2
  - mpeg7cdvs::AlpOctaveBF, 45
- h\_mode
  - mpeg7cdvs::AlpOctaveBF, 45
- HALF\_PI
  - AlpOctaveBF.h, 173
- hasBitSelection
  - mpeg7cdvs::Parameters, 143
  - mpeg7cdvs::SCFVFactory, 156
  - mpeg7cdvs::SCFVSignature, 162
- hasLocalizationInliers
  - mpeg7cdvs::PointPairs, 148
- hasRecallGraph
  - mpeg7cdvs::Database, 105
- hasVar
  - mpeg7cdvs::Parameters, 143
  - mpeg7cdvs::SCFVSignature, 162
- hasVariance
  - mpeg7cdvs::SCFVFactory, 156
- height
  - mpeg7cdvs::AbstractDetector, 22
- HiResTimer, 130
  - elapsed, 130
  - start, 131
  - stop, 131
- HiResTimer.h, 192
- high
  - mpeg7cdvs::AC\_model, 28
- HomPoint2D
  - Points.h, 200
- HomPoint2DArray
  - Points.h, 200
- homogeneousPoints
  - Points.h, 200
- ifftplan
  - mpeg7cdvs::FrequencyFilter, 130
- ImageBuffer
  - mpeg7cdvs::ImageBuffer, 134
- ImageBuffer.h, 193
- imageHeight
  - mpeg7cdvs::CompressedFeatureList, 97
  - mpeg7cdvs::FeatureList, 124
- imageWidth
  - mpeg7cdvs::CompressedFeatureList, 97
  - mpeg7cdvs::FeatureList, 124
- imagefile
  - mpeg7cdvs::CompressedFeatureList, 97
- images
  - mpeg7cdvs::Database, 107
- index
  - mpeg7cdvs::RetrievalData, 154
- inhomogeneousPoints
  - Points.h, 200
- init
  - mpeg7cdvs::AC\_decoder, 25
  - mpeg7cdvs::AC\_encoder, 26
  - mpeg7cdvs::AC\_model, 29
  - mpeg7cdvs::AlpOctave, 40
  - mpeg7cdvs::SCFVFactory, 156
- inlierIndexes
  - mpeg7cdvs::PointPairs, 148
- inmat
  - mpeg7cdvs::FrequencyFilter, 130
- input\_type
  - mpeg7cdvs, 17
- inter
  - Polygon, 150
- isDescriptorInDB
  - mpeg7cdvs::CdvsServer, 82
  - mpeg7cdvs::CdvsServerImpl, 88
- isEnabled
  - TraceManager, 165
- isIdentity
  - Projective2D, 151
- iscale
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 114
- ix
  - mpeg7cdvs::FeatureAlp, 114
- iy
  - mpeg7cdvs::FeatureAlp, 114
- jumpTo
  - mpeg7cdvs::BitInputStream, 49
  - mpeg7cdvs::BitOutputStream, 52
- kernel
  - mpeg7cdvs::Filter, 128
- keypoints



- mpeg7cdvs::AlpOctave, [41](#)
  - mpeg7cdvs::AlpOctaveBF, [45](#)
- LOG\_OF\_2
  - AlpOctaveBF.h, [173](#)
- libcdvsbflog\_is\_present
  - CdvsClientBflog.h, [179](#)
- loadDB
  - mpeg7cdvs::CdvsServer, [82](#)
  - mpeg7cdvs::CdvsServerImpl, [88](#)
- loadHammingWeight
  - mpeg7cdvs::SCFVIndex, [158](#)
- local\_score
  - mpeg7cdvs::PointPairs, [149](#)
- local\_threshold
  - mpeg7cdvs::PointPairs, [149](#)
- locationBits
  - mpeg7cdvs::Parameters, [143](#)
- LookUpTable
  - mpeg7cdvs::LookUpTable, [138](#)
- low
  - mpeg7cdvs::AC\_model, [29](#)
- MATCH\_TYPE\_BOTH
  - mpeg7cdvs, [18](#)
- MATCH\_TYPE\_DEFAULT
  - mpeg7cdvs, [18](#)
- MATCH\_TYPE\_GLOBAL
  - mpeg7cdvs, [18](#)
- MATCH\_TYPE\_LOCAL
  - mpeg7cdvs, [18](#)
- m\_GoFThreshold
  - DistratEigen, [109](#)
- m\_bFitsGood
  - DistratEigen, [109](#)
- m\_c
  - DistratEigen, [109](#)
- m\_vWordBlock
  - mpeg7cdvs::SCFVSignature, [163](#)
- m\_vWordVarBlock
  - mpeg7cdvs::SCFVSignature, [163](#)
- MAX
  - AlpOctaveBF.h, [173](#)
- MAX\_CODE
  - mpeg7cdvs, [18](#)
- MAX\_FILENAME\_LENGTH
  - map.h, [194](#)
- MAX\_FILTER\_WIDTH
  - AlpOctaveBF.h, [173](#)
- MAX\_FREQ
  - mpeg7cdvs, [18](#)
- MAX\_MATCHES
  - map.h, [194](#)
- MAX\_NUM\_FEATURES
  - mpeg7cdvs::FeatureList, [124](#)
- MIN
  - AlpOctaveBF.h, [173](#)
- MODE\_BOTTOM
  - AlpOctaveBF.h, [173](#)
- MODE\_FINISH
  - AlpOctaveBF.h, [173](#)
- MODE\_LEFT
  - AlpOctaveBF.h, [173](#)
- MODE\_RIGHT
  - AlpOctaveBF.h, [173](#)
- MODE\_TOP
  - AlpOctaveBF.h, [173](#)
- main.main, [193](#)
- makeHomography
  - Projective2D, [151](#), [152](#)
- map.h, [194](#)
  - alloc\_qra\_list, [194](#)
  - FILENAME, [194](#)
  - free\_qra\_list, [194](#)
  - MAX\_FILENAME\_LENGTH, [194](#)
  - MAX\_MATCHES, [194](#)
  - mean\_average\_precision, [194](#)
  - read\_qra\_list, [194](#)
  - success\_rate\_for\_top\_match, [194](#)
  - success\_rate\_for\_top\_match\_2, [194](#)
  - write\_qra\_list, [194](#)
- Match, [138](#)
  - featureInd, [139](#)
  - otherFeatureInd, [139](#)
  - sortMatchByWeight, [139](#)
  - sortMatchPredicate, [139](#)
  - weight, [139](#)
- match
  - mpeg7cdvs::CdvsServer, [82](#), [83](#)
  - mpeg7cdvs::CdvsServerImpl, [88](#), [89](#)
- Match.h, [195](#)
- match\_2way\_DISJOINT1
  - mpeg7cdvs, [18](#)
- match\_2way\_DISJOINT2
  - mpeg7cdvs, [18](#)
- match\_2way\_INTERSECTION
  - mpeg7cdvs, [18](#)
- match\_dirs
  - mpeg7cdvs::PointPairs, [149](#)
- matchCompressedDescriptors\_oneWay
  - mpeg7cdvs::Database, [105](#)
- matchCompressedDescriptors\_twoWay
  - mpeg7cdvs::Database, [105](#)
- matchDescriptors\_oneWay
  - mpeg7cdvs::CompressedFeatureList, [95](#)
- matchDescriptors\_twoWay
  - mpeg7cdvs::CompressedFeatureList, [95](#)
- matchImages
  - mpeg7cdvs::SCFVIndex, [158](#)
- matchImages\_bitselection
  - mpeg7cdvs::SCFVIndex, [158](#)
- matchPair
  - TraceManager, [165](#)
- matchResults
  - TraceManager, [166](#)
- matches
  - QRA, [154](#)



- maxsize
  - mpeg7cdvs::Filter, 128
- mean\_average\_precision
  - map.h, 194
- merge
  - mpeg7cdvs::Database, 106
- minNumInliers
  - mpeg7cdvs::Parameters, 144
- mode
  - mpeg7cdvs::FeatureAlpBF, 116
- modeExt
  - mpeg7cdvs::Parameters, 144
- modeld
  - mpeg7cdvs::CdvsClientImpl, 64
  - mpeg7cdvs::Database, 107
- modelFit
  - Projective2D, 152
- moveByHomography
  - Projective2D, 152
- mpeg7cdvs, 15
  - CODE\_VALUE, 17
  - CODE\_VALUE\_BITS, 18
  - element\_type, 17
  - FREQUENCY\_BITS, 18
  - filter\_type, 17
  - gama, 18
  - gs\_type, 17
  - input\_type, 17
  - MATCH\_TYPE\_BOTH, 18
  - MATCH\_TYPE\_DEFAULT, 18
  - MATCH\_TYPE\_GLOBAL, 18
  - MATCH\_TYPE\_LOCAL, 18
  - MAX\_CODE, 18
  - MAX\_FREQ, 18
  - match\_2way\_DISJOINT1, 18
  - match\_2way\_DISJOINT2, 18
  - match\_2way\_INTERSECTION, 18
  - NOT\_FOUND, 18
  - num\_bit\_selection, 18
  - numberCentroids, 18
  - ONE\_FOURTH, 18
  - ONE\_HALF, 18
  - PCASiftLength, 18
  - PRECISION, 19
  - ParameterSet, 18
  - recallGraph\_t, 18
  - recallGraphNode\_t, 18
  - THREE\_FOURTHS, 19
  - TOP\_VALUE, 19
- mpeg7cdvs::AC\_decoder, 23
  - bits, 23
  - decode\_symbol, 23
  - done, 25
  - init, 25
- mpeg7cdvs::AC\_encoder, 25
  - bits, 26
  - done, 26
  - encode\_symbol, 26
  - init, 26
- mpeg7cdvs::AC\_model, 26
  - ~AC\_model, 27
  - AC\_model, 27
  - adaptative, 27
  - check, 27
  - count, 28
  - done, 28
  - getSymbol, 28
  - high, 28
  - init, 29
  - low, 29
  - print, 29
  - update, 29
- mpeg7cdvs::AbstractDetector, 21
  - ~AbstractDetector, 22
  - AbstractDetector, 22
  - detect, 22
  - extract, 22
  - height, 22
  - originalHeight, 22
  - originalWidth, 23
  - width, 23
- mpeg7cdvs::AlpDetector, 30
  - ~AlpDetector, 31
  - AlpDetector, 31
  - detect, 31
  - extract, 31
  - sortPdf, 32
- mpeg7cdvs::AlpDetectorBF, 32
  - AlpDetectorBF, 33
  - detect, 33
  - extract, 34
- mpeg7cdvs::AlpDetectorLowMem, 34
  - AlpDetectorLowMem, 35
  - detect, 35
  - extract, 36
- mpeg7cdvs::AlpOctave, 36
  - ~AlpOctave, 37
  - AlpOctave, 37
  - clear, 38
  - computeDescriptor, 38
  - computeDescriptor\_fast\_mode, 38
  - detect, 38
  - detectDuplicates, 38
  - empty, 39
  - getAlpKeypoints, 39
  - getHeight, 39
  - getKeypoints, 39
  - getKeypoints\_fast\_mode, 39
  - getOctave, 40
  - getWidth, 40
  - init, 40
  - keypoints, 41
  - processFirst, 41
  - processNext, 41
- mpeg7cdvs::AlpOctaveBF, 41
  - ~AlpOctaveBF, 43

- Allocate, 43
- AlpOctaveBF, 43
- block\_buffer\_len, 45
- block\_col\_n, 45
- block\_col\_remain\_height, 45
- block\_height, 45
- block\_map, 45
- block\_row\_n, 45
- block\_row\_remain\_width, 45
- block\_width, 45
- Clear, 43
- computeDescriptor, 43
- cur\_block\_id, 45
- detect, 43
- detectDuplicates, 44
- empty, 44
- extrema\_h, 45
- extrema\_sx, 45
- extrema\_sy, 45
- extrema\_w, 45
- filters, 45
- FrequencyFiltering, 44
- getBlockInput, 44
- getHeight, 44
- getOctave, 44
- getWidth, 44
- h2, 45
- h\_mode, 45
- keypoints, 45
- padWidth, 45
- processOctave, 44
- SpatialFiltering, 44
- sx, 45
- sy, 45
- v\_mode, 45
- w2, 45
- windowSize, 45
- mpeg7cdvs::BitInputStream, 46
  - ~BitInputStream, 48
  - align, 48
  - available, 48
  - BitInputStream, 47
  - close, 48
  - consumed, 48
  - eof, 48
  - getPointer, 48
  - getSize, 48
  - jumpTo, 49
  - open, 49
  - read, 49
  - reset, 50
  - skip, 50
- mpeg7cdvs::BitOutputStream, 50
  - ~BitOutputStream, 51
  - align, 51
  - available, 52
  - BitOutputStream, 51
  - close, 52
  - eof, 52
  - flush, 52
  - getPointer, 52
  - getSize, 52
  - jumpTo, 52
  - open, 53
  - produced, 53
  - reset, 53
  - skip, 53
  - write, 53, 54
- mpeg7cdvs::Buffer, 55
  - ~Buffer, 57
  - assign, 57
  - Buffer, 57
  - clear, 57
  - compare, 57
  - data, 57
  - empty, 57
  - equals, 57
  - fill, 58
  - operator=, 58
  - operator==, 58
  - read, 58
  - resize, 58
  - size, 58
  - swap, 58
  - write, 58
- mpeg7cdvs::CDVSPPOINT, 78
  - x, 78
  - y, 78
- mpeg7cdvs::CdvsClient, 58
  - ~CdvsClient, 59
  - cdvsClientFactory, 59
  - encode, 60
- mpeg7cdvs::CdvsClientBflog, 60
  - ~CdvsClientBflog, 61
  - CdvsClientBflog, 61
  - encode, 61
- mpeg7cdvs::CdvsClientImpl, 62
  - ~CdvsClientImpl, 63
  - CdvsClientImpl, 63
  - encode, 63
  - g\_factory, 64
  - modelId, 64
  - params, 64
- mpeg7cdvs::CdvsClientLowMem, 64
  - ~CdvsClientLowMem, 65
  - CdvsClientLowMem, 65
  - encode, 65
- mpeg7cdvs::CdvsConfiguration, 66
  - ~CdvsConfiguration, 67
  - cdvsConfigurationFactory, 67
  - getMode, 67
  - getParameters, 67
  - setParameters, 69
- mpeg7cdvs::CdvsConfigurationImpl, 69
  - ~CdvsConfigurationImpl, 70
  - CdvsConfigurationImpl, 70

- getParameters, 70
- setParameters, 71
- mpeg7cdvs::CdvsDescriptor, 71
  - ~CdvsDescriptor, 73
  - buffer, 76
  - CdvsDescriptor, 73
  - check, 73
  - clear, 73
  - decode, 73
  - encode, 73
  - featurelist, 76
  - getGlobalHasBitSelection, 74
  - getGlobalHasVariance, 74
  - getHistogramCountSize, 74
  - getHistogramMapSizeX, 74
  - getHistogramMapSizeY, 74
  - getModelID, 74
  - getNumberOfLocalDescriptors, 74
  - getOriginalImageXResolution, 74
  - getOriginalImageYResolution, 74
  - getRelevanceBitsPresent, 75
  - getVersionID, 75
  - print, 75
  - scfvSignature, 76
  - setGlobalHasBitSelection, 75
  - setGlobalHasVariance, 75
  - setHistogramCountSize, 75
  - setHistogramMapSizeX, 75
  - setHistogramMapSizeY, 75
  - setModelID, 75
  - setNumberOfLocalDescriptors, 75
  - setOriginalImageXResolution, 75
  - setOriginalImageYResolution, 76
  - setRelevanceBitsPresent, 76
  - setVersionID, 76
- mpeg7cdvs::CdvsException, 76
  - ~CdvsException, 78
  - CdvsException, 77
  - what, 78
- mpeg7cdvs::CdvsServer, 78
  - ~CdvsServer, 80
  - addDescriptorToDB, 80
  - cdvsServerFactory, 80
  - clearDB, 81
  - commitDB, 81
  - createDB, 81
  - decode, 81
  - getImageld, 82
  - isDescriptorInDB, 82
  - loadDB, 82
  - match, 82, 83
  - replaceDescriptorInDB, 83
  - retrieve, 84
  - sizeofDB, 84
  - storeDB, 84
- mpeg7cdvs::CdvsServerImpl, 85
  - ~CdvsServerImpl, 86
  - addDescriptorToDB, 86
  - CdvsServerImpl, 86
  - clearDB, 87
  - commitDB, 87
  - createDB, 87
  - decode, 87
  - getImageld, 88
  - isDescriptorInDB, 88
  - loadDB, 88
  - match, 88, 89
  - replaceDescriptorInDB, 89
  - retrieve, 90
  - sizeofDB, 90
  - storeDB, 90
- mpeg7cdvs::CompressedFeatureList, 91
  - ~CompressedFeatureList, 93
  - CompressedFeatureList, 93
  - descrBytes, 93
  - features, 97
  - getDistance, 93
  - imageHeight, 97
  - imageWidth, 97
  - imagefile, 97
  - matchDescriptors\_oneWay, 95
  - matchDescriptors\_twoWay, 95
  - nDescLength, 97
  - nFeatures, 95
  - numFeatures, 97
  - operator=, 95
  - originalHeight, 97
  - originalWidth, 97
  - print, 96
  - read, 96
  - readFromFile, 96
  - setFilename, 96
  - swap, 96
  - write, 96
  - writeToFile, 97
  - Xcoord, 98
  - Ycoord, 98
- mpeg7cdvs::CsscCoordinateCoding, 98
  - ~CsscCoordinateCoding, 99
  - AddImageSample, 99
  - CONTEXT\_RANGE, 101
  - compare, 99
  - CsscCoordinateCoding, 99
  - EndTrainingMode, 99
  - exportVars, 100
  - fromBinary, 101
  - generateFeatureList, 101
  - generateHistogramMap, 101
  - readSeparateContext, 101
  - StartTrainingMode, 101
  - toBinary, 101
  - writeSeparateContext, 101
- mpeg7cdvs::CsscCoordinateCoding::CircularSum-Context, 91
  - vCount, 91
  - vInitialMap, 91

- vMap, 91
- mpeg7cdvs::Database, 102
  - ~Database, 103
  - addImage, 103
  - clear, 104
  - copyImageName, 104
  - Database, 103
  - find, 104
  - getImageName, 104
  - getMode, 104
  - getRecallGraph, 104
  - hasRecallGraph, 105
  - images, 107
  - matchCompressedDescriptors\_oneWay, 105
  - matchCompressedDescriptors\_twoWay, 105
  - merge, 106
  - modelId, 107
  - read, 106
  - readFromFile, 106
  - readHeader, 106
  - recallGraph, 107
  - replaceImage, 106
  - size, 107
  - write, 107
  - writeToFile, 107
- mpeg7cdvs::Feature, 109
  - curvRatio, 111
  - curvSigma, 111
  - descr, 111
  - descrLength, 111
  - Feature, 111
  - fromFile, 111
  - iscale, 111
  - octave, 111
  - orientation, 111
  - pdf, 111
  - peak, 112
  - qdescr, 112
  - relevance, 112
  - scale, 112
  - spatialIndex, 112
  - toFile, 111
  - x, 112
  - y, 112
- mpeg7cdvs::FeatureAlp, 112
  - curvRatio, 114
  - curvSigma, 114
  - distL1, 114
  - iscale, 114
  - ix, 114
  - iy, 114
  - octave, 114
  - operator<<, 114
  - orientation, 115
  - pdf, 115
  - peak, 115
  - sigma, 115
  - spatialIndex, 115
  - x, 115
  - y, 115
- mpeg7cdvs::FeatureAlpBF, 115
  - block\_id, 116
  - mode, 116
- mpeg7cdvs::FeatureList, 117
  - addFeature, 119
  - clear, 119
  - compareCoordinates, 119
  - compareDescriptors, 119
  - compareKeypoints, 119
  - compress, 121
  - CompressedFeatureList, 123
  - computeMaxPoints, 121
  - FeatureList, 119
  - features, 124
  - fromBinary, 121
  - fromFile, 121
  - getRelevantPoints, 122
  - imageHeight, 124
  - imageWidth, 124
  - MAX\_NUM\_FEATURES, 124
  - nFeatures, 122
  - originalHeight, 124
  - originalWidth, 124
  - print, 122
  - qdescr\_size, 124
  - select, 122
  - selectFirst, 122
  - selectFromTo, 122
  - setRelevantPoints, 122
  - setResolution, 123
  - sortRelevance, 123
  - sortSpatialIndex, 123
  - toBinary, 123
  - toFile, 123
- mpeg7cdvs::Filter, 127
  - Filter, 128
  - kernel, 128
  - maxsize, 128
  - ntaps, 128
  - print, 128
  - sigma, 128
- mpeg7cdvs::FrequencyFilter, 128
  - ~FrequencyFilter, 129
  - close\_lib, 129
  - Convolution, 129
  - fblock, 129
  - fblock\_flog, 129
  - fftplan, 129
  - FrequencyFilter, 129
  - ifftplan, 130
  - inmat, 130
  - open\_lib, 129
- mpeg7cdvs::ImageBuffer, 132
  - ~ImageBuffer, 134
  - buffer, 137
  - fastInterpolate, 134

- fastScalarQuantize, 134
- ImageBuffer, 134
- print, 134
- printDescr, 134
- printHeader, 135
- read, 135
- resample, 135
- resampleIfGreater, 135
- resize, 136
- scalarQuantize, 136
- sortPdfPredicate, 136
- sortPredicate, 136
- swap, 137
- writeBMP, 137
- writeRawData, 137
- mpeg7cdvs::LookUpTable, 137
  - f, 138
  - LookUpTable, 138
- mpeg7cdvs::Parameters, 140
  - ~Parameters, 142
  - blockWidth, 143
  - chiSquarePercentile, 143
  - ctxTableIdx, 143
  - debugLevel, 143
  - descLength, 143
  - gdThreshold, 143
  - gdThresholdMixed, 143
  - getModelID, 142
  - hasBitSelection, 143
  - hasVar, 143
  - locationBits, 143
  - minNumInliers, 144
  - modeExt, 144
  - nBits, 144
  - nModes, 144
  - numRelevantPoints, 144
  - numberOfElementGroups, 144
  - Parameters, 142
  - queryExpansionLoops, 144
  - ransacNumTests, 144
  - ransacThreshold, 144
  - ratioThreshold, 144
  - readAll, 142
  - readParameters, 142
  - resizeMaxSize, 144
  - retrievalLoops, 144
  - retrievalMaxPoints, 145
  - scfvThreshold, 145
  - selectMaxPoints, 145
  - wmMixed, 145
  - wmMixed2Way, 145
  - wmRetrieval, 145
  - wmRetrieval2Way, 145
  - wmThreshold, 145
  - wmThreshold2Way, 145
- mpeg7cdvs::PointPairs, 145
  - ~PointPairs, 147
  - addPair, 147
  - getInlierWeight, 148
  - getTotalWeight, 148
  - global\_score, 148
  - global\_threshold, 148
  - hasLocalizationInliers, 148
  - inlierIndexes, 148
  - local\_score, 149
  - local\_threshold, 149
  - match\_dirs, 149
  - nInliers, 149
  - nMatched, 149
  - operator=, 148
  - PointPairs, 147
  - score, 149
  - size, 149
  - toFullResolution, 148
  - weights, 149
  - x1, 149
  - x2, 149
  - y1, 149
  - y2, 149
- mpeg7cdvs::RetrievalData, 154
  - fScore, 154
  - gScore, 154
  - index, 154
  - nInliers, 155
  - nMatched, 155
- mpeg7cdvs::SCFVFactory, 155
  - generateSCFV, 155
  - hasBitSelection, 156
  - hasVariance, 156
  - init, 156
  - SCFVFactory, 155
- mpeg7cdvs::SCFVIndex, 156
  - append, 157
  - clear, 157
  - generateWeight, 157
  - getImage, 158
  - loadHammingWeight, 158
  - matchImages, 158
  - matchImages\_bitselection, 158
  - numberImages, 159
  - query, 159
  - query\_bitselection, 159
  - read, 159
  - replace, 159
  - reserve, 159
  - resize, 160
  - SCFVIndex, 157
  - write, 160
- mpeg7cdvs::SCFVSignature, 160
  - clear, 161
  - compare, 161
  - compressedNumBits, 161
  - fromFile, 162
  - getNorm, 162
  - getVisited, 162
  - hasBitSelection, 162

- hasVar, 162
- m\_vWordBlock, 163
- m\_vWordVarBlock, 163
- print, 162
- read, 162
- SCFVSignature, 161
- setNorm, 162
- setVisited, 162
- size, 162
- table\_bit\_selection, 163
- toFile, 163
- write, 163
- n\_matches
  - QRA, 154
- nBits
  - mpeg7cdvs::Parameters, 144
- nDescLength
  - mpeg7cdvs::CompressedFeatureList, 97
- nFeatures
  - mpeg7cdvs::CompressedFeatureList, 95
  - mpeg7cdvs::FeatureList, 122
- nInliers
  - mpeg7cdvs::PointPairs, 149
  - mpeg7cdvs::RetrievalData, 155
- nMatched
  - mpeg7cdvs::PointPairs, 149
  - mpeg7cdvs::RetrievalData, 155
- nModes
  - mpeg7cdvs::Parameters, 144
- NOT\_FOUND
  - mpeg7cdvs, 18
- ntaps
  - mpeg7cdvs::Filter, 128
- num\_bit\_selection
  - mpeg7cdvs, 18
- numFeatures
  - mpeg7cdvs::CompressedFeatureList, 97
- numRelevantPoints
  - mpeg7cdvs::Parameters, 144
- numberCentroids
  - mpeg7cdvs, 18
- numberImages
  - mpeg7cdvs::SCFVIndex, 159
- numberOfElementGroups
  - mpeg7cdvs::Parameters, 144
- ONE\_FOURTH
  - mpeg7cdvs, 18
- ONE\_HALF
  - mpeg7cdvs, 18
- octave
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 114
- open
  - mpeg7cdvs::BitInputStream, 49
  - mpeg7cdvs::BitOutputStream, 53
- open\_lib
  - mpeg7cdvs::FrequencyFilter, 129
- openTxt
  - TraceManager, 166
- openXml
  - TraceManager, 166
- operator<<
  - mpeg7cdvs::FeatureAlp, 114
- operator>>
  - Points.h, 201
- operator=
  - mpeg7cdvs::Buffer, 58
  - mpeg7cdvs::CompressedFeatureList, 95
  - mpeg7cdvs::PointPairs, 148
- operator==
  - mpeg7cdvs::Buffer, 58
- orientation
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 115
- originalHeight
  - mpeg7cdvs::AbstractDetector, 22
  - mpeg7cdvs::CompressedFeatureList, 97
  - mpeg7cdvs::FeatureList, 124
- originalWidth
  - mpeg7cdvs::AbstractDetector, 23
  - mpeg7cdvs::CompressedFeatureList, 97
  - mpeg7cdvs::FeatureList, 124
- osxtmspec
  - TM\_COUNTER, 164
- otherFeatureInd
  - Match, 139
- PAD\_WIDTH
  - AlpOctaveBF.h, 173
- PADDING\_REPEAT
  - AlpOctaveBF.h, 173
- PCASiftLength
  - mpeg7cdvs, 18
- PI
  - AlpOctaveBF.h, 173
- PI2
  - AlpOctaveBF.h, 173
- PRECISION
  - mpeg7cdvs, 19
- padWidth
  - mpeg7cdvs::AlpOctaveBF, 45
- ParameterSet
  - mpeg7cdvs, 18
- Parameters
  - mpeg7cdvs::Parameters, 142
- Parameters.h, 195
- params
  - mpeg7cdvs::CdvsClientImpl, 64
- pdf
  - mpeg7cdvs::Feature, 111
  - mpeg7cdvs::FeatureAlp, 115
- peak
  - mpeg7cdvs::Feature, 112
  - mpeg7cdvs::FeatureAlp, 115
- platform.h, 196, 197
- point

- BoundingBox.h, 177
- Point2D
  - Points.h, 200
- Point2DArray
  - Points.h, 200
- PointPairs
  - mpeg7cdvs::PointPairs, 147
- PointPairs.h, 198
- Points.h, 198
  - baricenter, 200
  - HomPoint2D, 200
  - HomPoint2DArray, 200
  - homogeneousPoints, 200
  - inhomogeneousPoints, 200
  - operator>>, 201
  - Point2D, 200
  - Point2DArray, 200
  - preconditionTransform, 201
  - randomSubset, 201
  - read, 201
  - spread, 201
- Polygon, 150
  - inter, 150
  - Polygon, 150
- preconditionTransform
  - Points.h, 201
- print
  - mpeg7cdvs::AC\_model, 29
  - mpeg7cdvs::CdvsDescriptor, 75
  - mpeg7cdvs::CompressedFeatureList, 96
  - mpeg7cdvs::FeatureList, 122
  - mpeg7cdvs::Filter, 128
  - mpeg7cdvs::ImageBuffer, 134
  - mpeg7cdvs::SCFVSignature, 162
- printDescr
  - mpeg7cdvs::ImageBuffer, 134
- printHeader
  - mpeg7cdvs::ImageBuffer, 135
- processFirst
  - mpeg7cdvs::AlpOctave, 41
- processNext
  - mpeg7cdvs::AlpOctave, 41
- processOctave
  - mpeg7cdvs::AlpOctaveBF, 44
- produced
  - mpeg7cdvs::BitOutputStream, 53
- Projective2D, 150
  - isIdentity, 151
  - makeHomography, 151, 152
  - modelFit, 152
  - moveByHomography, 152
  - Projective2D, 151
  - Projective2D, 151
  - ransac, 153
- Projective2D.h, 201
- QRA, 153
  - matches, 154
  - n\_matches, 154
  - query, 154
- QUTER\_PI
  - AlpOctaveBF.h, 173
- qdescr
  - mpeg7cdvs::Feature, 112
- qdescr\_size
  - mpeg7cdvs::FeatureList, 124
- query
  - mpeg7cdvs::SCFVIndex, 159
  - QRA, 154
- query\_bitselection
  - mpeg7cdvs::SCFVIndex, 159
- queryExpansionLoops
  - mpeg7cdvs::Parameters, 144
- radius
  - Ball2D, 46
- randomSubset
  - Points.h, 201
- ransac
  - Projective2D, 153
- ransacNumTests
  - mpeg7cdvs::Parameters, 144
- ransacThreshold
  - mpeg7cdvs::Parameters, 144
- ratioThreshold
  - mpeg7cdvs::Parameters, 144
- read
  - BoundingBox, 55
  - mpeg7cdvs::BitInputStream, 49
  - mpeg7cdvs::Buffer, 58
  - mpeg7cdvs::CompressedFeatureList, 96
  - mpeg7cdvs::Database, 106
  - mpeg7cdvs::ImageBuffer, 135
  - mpeg7cdvs::SCFVIndex, 159
  - mpeg7cdvs::SCFVSignature, 162
  - Points.h, 201
- read\_qra\_list
  - map.h, 194
- readAll
  - mpeg7cdvs::Parameters, 142
- readAnnotation
  - FileManager, 126
- readFromFile
  - mpeg7cdvs::CompressedFeatureList, 96
  - mpeg7cdvs::Database, 106
- readHeader
  - mpeg7cdvs::Database, 106
- readParameters
  - mpeg7cdvs::Parameters, 142
- readSeparateContext
  - mpeg7cdvs::CsscCoordinateCoding, 101
- real\_t
  - BoundingBox.h, 177
- recallGraph
  - mpeg7cdvs::Database, 107
- recallGraph\_t
  - mpeg7cdvs, 18
- recallGraphNode\_t



- mpeg7cdvs, 18
- relevance
  - mpeg7cdvs::Feature, 112
- replace
  - mpeg7cdvs::SCFVIndex, 159
- replaceDescriptorInDB
  - mpeg7cdvs::CdvsServer, 83
  - mpeg7cdvs::CdvsServerImpl, 89
- replaceExt
  - FileManager, 126
- replacelImage
  - mpeg7cdvs::Database, 106
- resample
  - mpeg7cdvs::ImageBuffer, 135
- resampleIfGreater
  - mpeg7cdvs::ImageBuffer, 135
- reserve
  - mpeg7cdvs::SCFVIndex, 159
- reset
  - mpeg7cdvs::BitInputStream, 50
  - mpeg7cdvs::BitOutputStream, 53
- resize
  - mpeg7cdvs::Buffer, 58
  - mpeg7cdvs::ImageBuffer, 136
  - mpeg7cdvs::SCFVIndex, 160
- resizeMaxSize
  - mpeg7cdvs::Parameters, 144
- retrievalLoops
  - mpeg7cdvs::Parameters, 144
- retrievalMaxPoints
  - mpeg7cdvs::Parameters, 145
- retrieve
  - mpeg7cdvs::CdvsServer, 84
  - mpeg7cdvs::CdvsServerImpl, 90
- SCFVFactory
  - mpeg7cdvs::SCFVFactory, 155
- SCFVIndex
  - mpeg7cdvs::SCFVIndex, 157
- SCFVIndex.h, 202
- SCFVSignature
  - mpeg7cdvs::SCFVSignature, 161
- SHIFT\_LEFT
  - AlpOctaveBF.h, 173
- scalarQuantize
  - mpeg7cdvs::ImageBuffer, 136
- scale
  - mpeg7cdvs::Feature, 112
- scfvSignature
  - mpeg7cdvs::CdvsDescriptor, 76
- scfvThreshold
  - mpeg7cdvs::Parameters, 145
- score
  - mpeg7cdvs::PointPairs, 149
- select
  - mpeg7cdvs::FeatureList, 122
- selectFirst
  - mpeg7cdvs::FeatureList, 122
- selectFromTo
  - mpeg7cdvs::FeatureList, 122
- selectMaxPoints
  - mpeg7cdvs::Parameters, 145
- setAnnotationPath
  - FileManager, 127
- setDatasetPath
  - FileManager, 127
- setFilename
  - mpeg7cdvs::CompressedFeatureList, 96
- setGlobalHasBitSelection
  - mpeg7cdvs::CdvsDescriptor, 75
- setGlobalHasVariance
  - mpeg7cdvs::CdvsDescriptor, 75
- setHistogramCountSize
  - mpeg7cdvs::CdvsDescriptor, 75
- setHistogramMapSizeX
  - mpeg7cdvs::CdvsDescriptor, 75
- setHistogramMapSizeY
  - mpeg7cdvs::CdvsDescriptor, 75
- setModelID
  - mpeg7cdvs::CdvsDescriptor, 75
- setNorm
  - mpeg7cdvs::SCFVSignature, 162
- setNumberOfLocalDescriptors
  - mpeg7cdvs::CdvsDescriptor, 75
- setOriginalImageXResolution
  - mpeg7cdvs::CdvsDescriptor, 75
- setOriginalImageYResolution
  - mpeg7cdvs::CdvsDescriptor, 76
- setParameters
  - mpeg7cdvs::CdvsConfiguration, 69
  - mpeg7cdvs::CdvsConfigurationImpl, 71
- setRelevanceBitsPresent
  - mpeg7cdvs::CdvsDescriptor, 76
- setRelevantPoints
  - mpeg7cdvs::FeatureList, 122
- setResolution
  - mpeg7cdvs::FeatureList, 123
- setVersionID
  - mpeg7cdvs::CdvsDescriptor, 76
- setVisited
  - mpeg7cdvs::SCFVSignature, 162
- sigma
  - mpeg7cdvs::FeatureAlp, 115
  - mpeg7cdvs::Filter, 128
- size
  - mpeg7cdvs::Buffer, 58
  - mpeg7cdvs::Database, 107
  - mpeg7cdvs::PointPairs, 149
  - mpeg7cdvs::SCFVSignature, 162
- sizeofDB
  - mpeg7cdvs::CdvsServer, 84
  - mpeg7cdvs::CdvsServerImpl, 90
- skip
  - mpeg7cdvs::BitInputStream, 50
  - mpeg7cdvs::BitOutputStream, 53
- sortMatchByWeight
  - Match, 139



- sortMatchPredicate
  - Match, [139](#)
- sortPdf
  - mpeg7cdvs::AlpDetector, [32](#)
- sortPdfPredicate
  - mpeg7cdvs::ImageBuffer, [136](#)
- sortPredicate
  - mpeg7cdvs::ImageBuffer, [136](#)
- sortRelevance
  - mpeg7cdvs::FeatureList, [123](#)
- sortSpatialIndex
  - mpeg7cdvs::FeatureList, [123](#)
- SpatialFiltering
  - mpeg7cdvs::AlpOctaveBF, [44](#)
- spatialIndex
  - mpeg7cdvs::Feature, [112](#)
  - mpeg7cdvs::FeatureAlp, [115](#)
- spread
  - Points.h, [201](#)
- start
  - HiResTimer, [131](#)
  - TraceManager, [166](#)
- start\_timer
  - timer.h, [204](#)
- StartTrainingMode
  - mpeg7cdvs::CsscCoordinateCoding, [101](#)
- stop
  - HiResTimer, [131](#)
  - TraceManager, [166](#)
- stop\_timer
  - timer.h, [205](#)
- storeDB
  - mpeg7cdvs::CdvsServer, [84](#)
  - mpeg7cdvs::CdvsServerImpl, [90](#)
- success\_rate\_for\_top\_match
  - map.h, [194](#)
- success\_rate\_for\_top\_match\_2
  - map.h, [194](#)
- swap
  - mpeg7cdvs::Buffer, [58](#)
  - mpeg7cdvs::CompressedFeatureList, [96](#)
  - mpeg7cdvs::ImageBuffer, [137](#)
- sx
  - mpeg7cdvs::AlpOctaveBF, [45](#)
- sy
  - mpeg7cdvs::AlpOctaveBF, [45](#)
- Symbian/platform.h
  - assert, [196](#)
- THR\_QUTER\_PI
  - AlpOctaveBF.h, [173](#)
- THREE\_FOURTHS
  - mpeg7cdvs, [19](#)
- TM\_COUNTER, [163](#)
  - counter, [164](#)
  - osxtmspec, [164](#)
  - tmspec, [164](#)
  - tv\_nsec, [164](#)
  - tv\_sec, [164](#)
  - tv\_usec, [164](#)
  - TM\_STATE, [164](#)
  - freq, [164](#)
  - TOP\_VALUE
    - mpeg7cdvs, [19](#)
  - table\_bit\_selection
    - mpeg7cdvs::SCFVSignature, [163](#)
  - timer.h, [204](#)
    - elapsed\_time, [204](#)
    - start\_timer, [204](#)
    - stop\_timer, [205](#)
  - tmspec
    - TM\_COUNTER, [164](#)
  - toBinary
    - mpeg7cdvs::CsscCoordinateCoding, [101](#)
    - mpeg7cdvs::FeatureList, [123](#)
  - toFile
    - mpeg7cdvs::Feature, [111](#)
    - mpeg7cdvs::FeatureList, [123](#)
    - mpeg7cdvs::SCFVSignature, [163](#)
  - toFullResolution
    - mpeg7cdvs::PointPairs, [148](#)
  - TraceManager, [164](#)
    - ~TraceManager, [165](#)
    - isEnabled, [165](#)
    - matchPair, [165](#)
    - matchResults, [166](#)
    - openTxt, [166](#)
    - openXml, [166](#)
    - start, [166](#)
    - stop, [166](#)
    - TraceManager, [165](#)
    - TraceManager, [165](#)
  - TraceManager.h, [205](#)
  - tv\_nsec
    - TM\_COUNTER, [164](#)
  - tv\_sec
    - TM\_COUNTER, [164](#)
  - tv\_usec
    - TM\_COUNTER, [164](#)
  - update
    - mpeg7cdvs::AC\_model, [29](#)
  - v\_mode
    - mpeg7cdvs::AlpOctaveBF, [45](#)
  - vCount
    - mpeg7cdvs::CsscCoordinateCoding::CircularSum-Context, [91](#)
  - vInitialMap
    - mpeg7cdvs::CsscCoordinateCoding::CircularSum-Context, [91](#)
  - vMap
    - mpeg7cdvs::CsscCoordinateCoding::CircularSum-Context, [91](#)
  - w2
    - mpeg7cdvs::AlpOctaveBF, [45](#)
  - WCE/platform.h

- assert, [197](#)
- weight
  - Match, [139](#)
- weights
  - mpeg7cdvs::PointPairs, [149](#)
- what
  - mpeg7cdvs::CdvsException, [78](#)
- width
  - mpeg7cdvs::AbstractDetector, [23](#)
- windowSize
  - mpeg7cdvs::AlpOctaveBF, [45](#)
- wmMixed
  - mpeg7cdvs::Parameters, [145](#)
- wmMixed2Way
  - mpeg7cdvs::Parameters, [145](#)
- wmRetrieval
  - mpeg7cdvs::Parameters, [145](#)
- wmRetrieval2Way
  - mpeg7cdvs::Parameters, [145](#)
- wmThreshold
  - mpeg7cdvs::Parameters, [145](#)
- wmThreshold2Way
  - mpeg7cdvs::Parameters, [145](#)
- write
  - mpeg7cdvs::BitOutputStream, [53](#), [54](#)
  - mpeg7cdvs::Buffer, [58](#)
  - mpeg7cdvs::CompressedFeatureList, [96](#)
  - mpeg7cdvs::Database, [107](#)
  - mpeg7cdvs::SCFVIndex, [160](#)
  - mpeg7cdvs::SCFVSignature, [163](#)
- write\_qra\_list
  - map.h, [194](#)
- writeBMP
  - mpeg7cdvs::ImageBuffer, [137](#)
- writeRawData
  - mpeg7cdvs::ImageBuffer, [137](#)
- writeSeparateContext
  - mpeg7cdvs::CsscCoordinateCoding, [101](#)
- writeToFile
  - mpeg7cdvs::CompressedFeatureList, [97](#)
  - mpeg7cdvs::Database, [107](#)
- x
  - mpeg7cdvs::CDVSPPOINT, [78](#)
  - mpeg7cdvs::Feature, [112](#)
  - mpeg7cdvs::FeatureAlp, [115](#)
- x1
  - mpeg7cdvs::PointPairs, [149](#)
- x2
  - mpeg7cdvs::PointPairs, [149](#)
- Xcoord
  - mpeg7cdvs::CompressedFeatureList, [98](#)
- y
  - mpeg7cdvs::CDVSPPOINT, [78](#)
  - mpeg7cdvs::Feature, [112](#)
  - mpeg7cdvs::FeatureAlp, [115](#)
- y1
  - mpeg7cdvs::PointPairs, [149](#)
- y2
  - mpeg7cdvs::PointPairs, [149](#)
- Ycoord
  - mpeg7cdvs::CompressedFeatureList, [98](#)