

Image-Based Indoor Topological Navigation with Collision Avoidance for Pepper Robot

Documentation of pepper_navigation repository
https://github.com/suuman/pepper_navigation.git

Generated by Doxygen 1.8.17

1 Pepper_navigation	1
1.1 Dependencies	1
1.2 Build Instructions	1
1.3 Run Code	2
1.4 References	3
2 navmain	4
2.1 Dependencies	4
2.2 Usage	4
2.2.1 Libraries	4
2.2.2 Executables	5
3 navmain/maingui	6
4 depthnav	7
4.1 Free-space navigation using depth image	7
4.2 Dependencies	7
4.3 Usage	7
5 linenav	8
5.1 Image-based navigation using line segments	8
5.2 Dependencies	8
5.3 Usage	8
6 Mapping	9
6.0.1 Selection of Reference Images Based on the Line Segment Matching	9
6.1 Build	9
6.2 Usage	9
6.3 Creating Topological Map for Navigation	10
7 pepper_qi	11
7.1 Interface with naoqi library	11
7.2 Dependencies	11
8 qgv	12
8.1 Interactive Qt GraphViz display	12
8.2 Dependencies	12
8.3 Usage	12
9 Sample data for testing	13
9.1 offlinetest/	13
9.2 tmap/	13
10 Namespace Index	14
10.1 Namespace List	14
11 Hierarchical Index	15
11.1 Class Hierarchy	15

12 Class Index	17
12.1 Class List	17
13 File Index	19
13.1 File List	19
14 Namespace Documentation	21
14.1 AL Namespace Reference	21
14.2 AL::Math Namespace Reference	21
14.2.1 Function Documentation	21
14.2.1.1 distance()	22
14.2.1.2 distanceSquared()	23
14.2.1.3 pinv()	23
14.2.1.4 pose2dDiff()	24
14.2.1.5 pose2DInverse() [1/2]	24
14.2.1.6 pose2DInverse() [2/2]	24
14.2.1.7 pose2dInvertInPlace()	25
14.3 astar Namespace Reference	25
14.3.1 Typedef Documentation	26
14.3.1.1 cost	26
14.3.1.2 edge	26
14.3.1.3 edge_descriptor	26
14.3.1.4 mygraph_t	26
14.3.1.5 vertex	26
14.3.1.6 WeightMap	26
14.3.2 Variable Documentation	27
14.3.2.1 node	27
14.4 depthimagescanner Namespace Reference	27
14.5 tgraph Namespace Reference	27
14.6 Ui Namespace Reference	27
15 Class Documentation	28
15.1 astar::astar Class Reference	28
15.1.1 Detailed Description	28
15.1.2 Constructor & Destructor Documentation	28
15.1.2.1 astar()	28
15.1.3 Member Function Documentation	29
15.1.3.1 creategraph()	29
15.1.3.2 getweight()	29
15.1.3.3 searchinggraph()	29
15.1.3.4 searchinggraphrandom()	30
15.1.3.5 setedges()	30
15.1.3.6 setlocationherustics()	31
15.1.3.7 writegraph2file()	31
15.2 astar::astar_goal_visitor< Vertex > Class Template Reference	31

15.2.1 Detailed Description	32
15.2.2 Constructor & Destructor Documentation	32
15.2.2.1 astar_goal_visitor()	32
15.2.3 Member Function Documentation	32
15.2.3.1 examine_vertex()	33
15.3 tgraph::branch Struct Reference	33
15.3.1 Detailed Description	33
15.3.2 Member Data Documentation	33
15.3.2.1 diflag	33
15.3.2.2 eid	33
15.3.2.3 fold	34
15.3.2.4 indx	34
15.3.2.5 sid	34
15.4 CompareL Struct Reference	34
15.4.1 Detailed Description	34
15.4.2 Member Function Documentation	34
15.4.2.1 operator>()()	35
15.5 CompareS Struct Reference	35
15.5.1 Detailed Description	35
15.5.2 Member Function Documentation	35
15.5.2.1 operator>()()	35
15.6 depthimagescanner::DepthImageScanner Class Reference	35
15.6.1 Detailed Description	36
15.6.2 Constructor & Destructor Documentation	36
15.6.2.1 DepthImageScanner()	36
15.6.2.2 ~DepthImageScanner()	36
15.6.3 Member Function Documentation	36
15.6.3.1 convert_msg()	36
15.6.3.2 readlasersonar()	37
15.6.3.3 set_output_frame()	37
15.6.3.4 set_range_limits()	37
15.6.3.5 set_scan_height()	38
15.6.3.6 setdepthasD()	38
15.6.3.7 setdepthasUC()	38
15.6.3.8 setlmsname()	38
15.7 depthimagescanner::DepthTraits< T > Struct Template Reference	39
15.7.1 Detailed Description	39
15.8 depthimagescanner::DepthTraits< float > Struct Reference	39
15.8.1 Detailed Description	39
15.8.2 Member Function Documentation	39
15.8.2.1 fromMeters()	40
15.8.2.2 initializeBuffer()	40
15.8.2.3 toMeters()	40
15.8.2.4 valid()	40

15.9 depthimagescanner::DepthTraits< uint16_t > Struct Reference	40
15.9.1 Detailed Description	41
15.9.2 Member Function Documentation	41
15.9.2.1 fromMeters()	41
15.9.2.2 initializeBuffer()	41
15.9.2.3 toMeters()	41
15.9.2.4 valid()	41
15.10 dispNav Class Reference	42
15.10.1 Detailed Description	42
15.10.2 Constructor & Destructor Documentation	43
15.10.2.1 dispNav()	43
15.10.3 Member Function Documentation	43
15.10.3.1 closedisp()	43
15.10.3.2 displImages() [1/4]	43
15.10.3.3 displImages() [2/4]	43
15.10.3.4 displImages() [3/4]	44
15.10.3.5 displImages() [4/4]	44
15.10.3.6 displayimage()	45
15.10.3.7 dispNavigation() [1/2]	45
15.10.3.8 dispNavigation() [2/2]	46
15.10.3.9 saveimage()	46
15.10.3.10 setDisptime()	46
15.10.3.11 setKeyImages()	47
15.10.3.12 setpseudocolour()	47
15.10.3.13 showfeat()	47
15.10.4 Member Data Documentation	47
15.10.4.1 size	48
15.10.4.2 sz	48
15.11 astar::distance_heuristic< Graph, CostType, LocMap > Class Template Reference	48
15.11.1 Detailed Description	49
15.11.2 Member Typedef Documentation	49
15.11.2.1 Vertex	49
15.11.3 Constructor & Destructor Documentation	49
15.11.3.1 distance_heuristic()	49
15.11.4 Member Function Documentation	49
15.11.4.1 operator()()	50
15.12 tgraph::edge Struct Reference	50
15.12.1 Detailed Description	50
15.12.2 Member Data Documentation	50
15.12.2.1 eid	50
15.12.2.2 fold	50
15.12.2.3 sid	51
15.13 EdgeChains Struct Reference	51
15.13.1 Detailed Description	51

15.13.2 Member Data Documentation	51
15.13.2.1 numOfEdges	51
15.13.2.2 sld	51
15.13.2.3 xCors	52
15.13.2.4 yCors	52
15.14 EDLineDetector Class Reference	52
15.14.1 Detailed Description	53
15.14.2 Constructor & Destructor Documentation	53
15.14.2.1 EDLineDetector() [1/2]	53
15.14.2.2 EDLineDetector() [2/2]	53
15.14.2.3 ~EDLineDetector()	53
15.14.3 Member Function Documentation	53
15.14.3.1 EdgeDrawing()	54
15.14.3.2 EDline() [1/2]	54
15.14.3.3 EDline() [2/2]	54
15.14.4 Member Data Documentation	54
15.14.4.1 dxImg_	54
15.14.4.2 dyImg_	54
15.14.4.3 glmgWO_	55
15.14.4.4 imageHeight	55
15.14.4.5 imageWidth	55
15.14.4.6 lineDirection_	55
15.14.4.7 lineEndpoints_	55
15.14.4.8 lineEquations_	55
15.14.4.9 lines_	56
15.14.4.10 lineSalienc_	56
15.15 EDLineParam Struct Reference	56
15.15.1 Detailed Description	56
15.15.2 Member Data Documentation	56
15.15.2.1 anchorThreshold	56
15.15.2.2 gradientThreshold	57
15.15.2.3 ksize	57
15.15.2.4 lineFitErrThreshold	57
15.15.2.5 minLineLen	57
15.15.2.6 scanIntervals	57
15.15.2.7 sigma	57
15.16 astar::found_goal Struct Reference	58
15.16.1 Detailed Description	58
15.17 freespacenavigation Class Reference	58
15.17.1 Detailed Description	58
15.17.2 Constructor & Destructor Documentation	58
15.17.2.1 freespacenavigation() [1/2]	58
15.17.2.2 freespacenavigation() [2/2]	59
15.17.3 Member Function Documentation	59

15.17.3.1	getH()	59
15.17.3.2	getvelfreespace()	59
15.17.3.3	setbasevel()	60
15.17.3.4	setcurrentpose()	60
15.17.3.5	setinitialpose()	60
15.18	astar::graph_writer< Name, LocMap > Class Template Reference	61
15.18.1	Detailed Description	61
15.18.2	Constructor & Destructor Documentation	61
15.18.2.1	graph_writer()	61
15.18.3	Member Function Documentation	62
15.18.3.1	operator()()	62
15.19	astar::heuristic< Graph, CostType, LocMap > Class Template Reference	62
15.19.1	Detailed Description	63
15.19.2	Member Typedef Documentation	63
15.19.2.1	Vertex	63
15.19.3	Constructor & Destructor Documentation	63
15.19.3.1	heuristic()	63
15.19.4	Member Function Documentation	63
15.19.4.1	operator()()	64
15.20	kimRead Class Reference	64
15.20.1	Detailed Description	65
15.20.2	Constructor & Destructor Documentation	65
15.20.2.1	kimRead() [1/5]	65
15.20.2.2	kimRead() [2/5]	65
15.20.2.3	kimRead() [3/5]	66
15.20.2.4	kimRead() [4/5]	66
15.20.2.5	kimRead() [5/5]	66
15.20.2.6	~kimRead()	67
15.20.3	Member Function Documentation	67
15.20.3.1	addKeyImages()	67
15.20.3.2	currindex()	67
15.20.3.3	getkimfold()	67
15.20.3.4	getnextKeyImage() [1/2]	68
15.20.3.5	getnextKeyImage() [2/2]	68
15.20.3.6	getStartIndex()	68
15.20.3.7	insertKeyImages() [1/3]	69
15.20.3.8	insertKeyImages() [2/3]	69
15.20.3.9	insertKeyImages() [3/3]	69
15.20.3.10	insetKeyImage()	70
15.20.3.11	isLast()	70
15.20.3.12	kimsize()	70
15.20.3.13	removeBKeyImages()	70
15.20.3.14	removeKeyImages()	71
15.20.3.15	resetCurrentIndex()	71

15.20.3.16 resetIndex()	71
15.20.3.17 setCurrIndex()	71
15.20.3.18 setKeyImageIndex()	72
15.20.3.19 setKimbasefold()	72
15.20.3.20 setnodes()	72
15.20.3.21 showkim()	73
15.21 LineChains Struct Reference	73
15.21.1 Detailed Description	73
15.21.2 Member Data Documentation	73
15.21.2.1 numOfLines	73
15.21.2.2 sld	74
15.21.2.3 xCors	74
15.21.2.4 yCors	74
15.22 LineDescriptor Class Reference	74
15.22.1 Detailed Description	75
15.22.2 Member Enumeration Documentation	75
15.22.2.1 anonymous enum	75
15.22.3 Constructor & Destructor Documentation	75
15.22.3.1 LineDescriptor() [1/2]	75
15.22.3.2 LineDescriptor() [2/2]	75
15.22.3.3 ~LineDescriptor()	76
15.22.4 Member Function Documentation	76
15.22.4.1 computeBinaryLineDesc()	76
15.22.4.2 findLineDesc()	76
15.22.4.3 GetLineDescriptor()	76
15.22.4.4 LineMatchingBinary()	76
15.22.4.5 MatchLineByDescriptor()	77
15.22.4.6 OctaveKeyLines()	77
15.22.5 Member Data Documentation	77
15.22.5.1 bDistThreshold	77
15.22.5.2 LowestThreshold	77
15.22.5.3 NNDRThreshold	77
15.23 linematch Class Reference	78
15.23.1 Detailed Description	78
15.23.2 Constructor & Destructor Documentation	78
15.23.2.1 linematch()	78
15.23.3 Member Function Documentation	78
15.23.3.1 findCommonIndex()	78
15.23.3.2 findCommonIndex2()	79
15.23.3.3 getlinedesc()	79
15.23.3.4 matchlines()	79
15.23.3.5 matchlinesbinary()	80
15.23.3.6 readlinedesc()	80
15.24 linenavigation Class Reference	80

15.24.1 Detailed Description	81
15.24.2 Constructor & Destructor Documentation	82
15.24.2.1 linenavigation() [1/2]	82
15.24.2.2 linenavigation() [2/2]	82
15.24.3 Member Function Documentation	82
15.24.3.1 getdiffN2NN()	82
15.24.3.2 getRotVel() [1/2]	83
15.24.3.3 getRotVel() [2/2]	83
15.24.3.4 getRotVel2() [1/2]	83
15.24.3.5 getRotVel2() [2/2]	83
15.24.3.6 initiallocalisationLines() [1/2]	84
15.24.3.7 initiallocalisationLines() [2/2]	84
15.24.3.8 initlocalisation()	85
15.24.3.9 setDisplay()	85
15.24.3.10 setInitialKeyImages()	85
15.24.3.11 setK()	86
15.24.3.12 SetKeyImages()	86
15.24.3.13 step()	86
15.24.3.14 SwitchKeyImages()	87
15.24.3.15 SwitchtoNewKeyImages()	87
15.24.4 Member Data Documentation	87
15.24.4.1 linesno	87
15.24.4.2 linesnoN	88
15.24.4.3 linesnoNN	88
15.24.4.4 nextimg	88
15.24.4.5 nextnextimg	88
15.24.4.6 previmg	88
15.25 astar::location Struct Reference	88
15.25.1 Detailed Description	89
15.25.2 Member Data Documentation	89
15.25.2.1 x	89
15.25.2.2 y	89
15.26 Matrix< T > Class Template Reference	89
15.26.1 Detailed Description	89
15.26.2 Constructor & Destructor Documentation	90
15.26.2.1 Matrix()	90
15.26.2.2 ~Matrix()	90
15.26.3 Member Function Documentation	90
15.26.3.1 operator[]()	90
15.26.3.2 SetZero()	90
15.27 MyService Class Reference	91
15.27.1 Detailed Description	91
15.27.2 Constructor & Destructor Documentation	91
15.27.2.1 MyService()	91

15.27.3 Member Function Documentation	91
15.27.3.1 myCallback()	91
15.27.3.2 myCallback1()	91
15.27.3.3 myCallback2()	92
15.28 navigation Class Reference	92
15.28.1 Detailed Description	93
15.28.2 Constructor & Destructor Documentation	93
15.28.2.1 navigation()	93
15.28.3 Member Function Documentation	93
15.28.3.1 getinitdisp()	93
15.28.3.2 getRotVel()	93
15.28.3.3 getturninginkim()	94
15.28.3.4 initlocalisation() [1/2]	94
15.28.3.5 initlocalisation() [2/2]	94
15.28.3.6 step()	95
15.28.3.7 SwitchtoNewKeyImages()	95
15.28.3.8 usecollisionavoidance() [1/2]	95
15.28.3.9 usecollisionavoidance() [2/2]	95
15.28.4 Member Data Documentation	96
15.28.4.1 errfile	96
15.28.4.2 featfile	96
15.28.4.3 jacfile	96
15.28.4.4 velfile	96
15.29 NavInside Class Reference	96
15.29.1 Detailed Description	97
15.29.2 Constructor & Destructor Documentation	97
15.29.2.1 NavInside()	97
15.29.2.2 ~NavInside()	97
15.29.3 Member Function Documentation	97
15.29.3.1 setGraph()	98
15.29.3.2 setKimfoldoffline()	98
15.29.3.3 setPepperRobotPointer()	98
15.29.3.4 setPointers()	98
15.29.3.5 startRobotNavigation()	99
15.30 NavWindow Class Reference	99
15.30.1 Detailed Description	100
15.30.2 Constructor & Destructor Documentation	100
15.30.2.1 NavWindow()	100
15.30.2.2 ~NavWindow()	100
15.30.3 Member Function Documentation	100
15.30.3.1 drawGraph()	101
15.30.3.2 setKimfoldoffline()	101
15.30.3.3 setPepperRobotPointer()	101
15.30.3.4 setPointers()	102

15.31	tgraph::node Struct Reference	102
15.31.1	Detailed Description	102
15.31.2	Member Data Documentation	102
15.31.2.1	nodeid	102
15.31.2.2	nodename	103
15.32	Node Struct Reference	103
15.32.1	Detailed Description	103
15.32.2	Member Data Documentation	103
15.32.2.1	leftLineID	103
15.32.2.2	rightLineID	103
15.33	OctaveLine Struct Reference	104
15.33.1	Detailed Description	104
15.33.2	Member Data Documentation	104
15.33.2.1	lineIDInOctave	104
15.33.2.2	lineIDInScaleLineVec	104
15.33.2.3	lineLength	104
15.33.2.4	octaveCount	105
15.34	OctaveSingleLine Struct Reference	105
15.34.1	Detailed Description	105
15.34.2	Member Data Documentation	105
15.34.2.1	bdescriptor	105
15.34.2.2	descriptor	106
15.34.2.3	direction	106
15.34.2.4	endPointX	106
15.34.2.5	endPointY	106
15.34.2.6	ePointInOctaveX	106
15.34.2.7	ePointInOctaveY	106
15.34.2.8	lineLength	107
15.34.2.9	numOfPixels	107
15.34.2.10	octaveCount	107
15.34.2.11	salience	107
15.34.2.12	sPointInOctaveX	107
15.34.2.13	sPointInOctaveY	107
15.34.2.14	startPointX	108
15.34.2.15	startPointY	108
15.35	PairwiseLineMatching Class Reference	108
15.35.1	Detailed Description	108
15.35.2	Constructor & Destructor Documentation	108
15.35.2.1	PairwiseLineMatching()	108
15.35.3	Member Function Documentation	108
15.35.3.1	LineMatching()	109
15.36	pepperInterface Class Reference	109
15.36.1	Detailed Description	110
15.36.2	Constructor & Destructor Documentation	110

15.36.2.1 pepperInterface()	110
15.36.2.2 ~pepperInterface()	110
15.36.3 Member Function Documentation	110
15.36.3.1 adjusthead()	110
15.36.3.2 getCurrDepthImage()	110
15.36.3.3 getCurrImage()	111
15.36.3.4 getid()	111
15.36.3.5 getK()	111
15.36.3.6 getKd()	111
15.36.3.7 getmode()	111
15.36.3.8 getOdometryReading()	111
15.36.3.9 initPosture()	112
15.36.3.10 openCamera()	112
15.36.3.11 openDepthCamera()	112
15.36.3.12 rotate180()	112
15.36.3.13 setBaseVelocities() [1/2]	112
15.36.3.14 setBaseVelocities() [2/2]	112
15.36.3.15 startBaseMotionController()	113
15.36.3.16 startPepper()	113
15.36.3.17 wait()	113
15.37 pepperNavigation Class Reference	113
15.37.1 Detailed Description	114
15.37.2 Constructor & Destructor Documentation	114
15.37.2.1 pepperNavigation()	114
15.37.3 Member Function Documentation	114
15.37.3.1 closedisp()	114
15.37.3.2 continueNav()	115
15.37.3.3 getH()	115
15.37.3.4 getinitdisplacemt()	115
15.37.3.5 getturninginkim()	116
15.37.3.6 getvel freespace()	116
15.37.3.7 localise()	116
15.37.3.8 navigate()	116
15.37.3.9 setbasevel()	117
15.37.3.10 setcurrentpose()	117
15.37.3.11 setfreespacenav()	117
15.37.3.12 setinitialpose()	118
15.37.3.13 setofileid()	118
15.37.3.14 setPath()	118
15.38 pepperOffline Class Reference	119
15.38.1 Detailed Description	120
15.38.2 Constructor & Destructor Documentation	120
15.38.2.1 pepperOffline()	120
15.38.3 Member Function Documentation	120

15.38.3.1	adjusthead()	120
15.38.3.2	getCurrDepthImage()	121
15.38.3.3	getCurrImage()	121
15.38.3.4	getid()	121
15.38.3.5	getK()	121
15.38.3.6	getKd()	122
15.38.3.7	getmode()	122
15.38.3.8	getOdometryReading()	122
15.38.3.9	initPosture()	122
15.38.3.10	openCamera()	122
15.38.3.11	openDepthCamera()	123
15.38.3.12	rotate180()	123
15.38.3.13	setBaseVelocities() [1/2]	123
15.38.3.14	setBaseVelocities() [2/2]	123
15.38.3.15	startBaseMotionController()	124
15.38.3.16	startPepper()	124
15.38.3.17	wait()	124
15.39	pepperRobot Class Reference	125
15.39.1	Detailed Description	126
15.39.2	Constructor & Destructor Documentation	126
15.39.2.1	pepperRobot()	126
15.39.3	Member Function Documentation	126
15.39.3.1	adjusthead()	127
15.39.3.2	getCurrDepthImage()	127
15.39.3.3	getCurrImage()	127
15.39.3.4	getid()	127
15.39.3.5	getK()	128
15.39.3.6	getKd()	128
15.39.3.7	getmode()	128
15.39.3.8	getOdometryReading()	129
15.39.3.9	initPosture()	129
15.39.3.10	openCamera()	129
15.39.3.11	openDepthCamera()	129
15.39.3.12	rotate180()	130
15.39.3.13	setBaseVelocities() [1/2]	130
15.39.3.14	setBaseVelocities() [2/2]	130
15.39.3.15	startBaseMotionController()	131
15.39.3.16	startPepper()	131
15.39.3.17	wait()	131
15.40	pepperRobotVirtual Class Reference	132
15.40.1	Detailed Description	133
15.40.2	Constructor & Destructor Documentation	133
15.40.2.1	pepperRobotVirtual()	133
15.40.3	Member Function Documentation	134

15.40.3.1	adjusthead()	134
15.40.3.2	getCurrDepthImage()	134
15.40.3.3	getCurrImage()	134
15.40.3.4	getid()	135
15.40.3.5	getK()	135
15.40.3.6	getKd()	135
15.40.3.7	getmode()	136
15.40.3.8	getOdometryReading()	136
15.40.3.9	initPosture()	136
15.40.3.10	openCamera()	136
15.40.3.11	openDepthCamera()	137
15.40.3.12	rotate180()	137
15.40.3.13	setBaseVelocities() [1/2]	137
15.40.3.14	setBaseVelocities() [2/2]	138
15.40.3.15	startBaseMotionController()	138
15.40.3.16	startPepper()	138
15.40.3.17	wait()	139
15.41	pepperServices Class Reference	139
15.41.1	Detailed Description	139
15.41.2	Constructor & Destructor Documentation	139
15.41.2.1	pepperServices()	139
15.41.3	Member Function Documentation	140
15.41.3.1	armsCallback()	140
15.41.3.2	moveCallback()	140
15.41.4	Member Data Documentation	140
15.41.4.1	eventraised	140
15.42	Pixel Struct Reference	141
15.42.1	Detailed Description	141
15.42.2	Member Data Documentation	141
15.42.2.1	x	141
15.42.2.2	y	141
15.43	AL::Math::Pose2D Struct Reference	141
15.43.1	Detailed Description	143
15.43.2	Constructor & Destructor Documentation	143
15.43.2.1	Pose2D() [1/4]	143
15.43.2.2	Pose2D() [2/4]	143
15.43.2.3	Pose2D() [3/4]	144
15.43.2.4	Pose2D() [4/4]	144
15.43.3	Member Function Documentation	145
15.43.3.1	diff()	145
15.43.3.2	distance()	145
15.43.3.3	distanceSquared()	145
15.43.3.4	fromPolarCoordinates()	146
15.43.3.5	getAngle()	146

15.43.3.6 inverse()	146
15.43.3.7 isNear()	147
15.43.3.8 norm()	147
15.43.3.9 normalize()	147
15.43.3.10 operator!==()	147
15.43.3.11 operator*() [1/2]	148
15.43.3.12 operator*() [2/2]	148
15.43.3.13 operator*==() [1/2]	148
15.43.3.14 operator*==() [2/2]	149
15.43.3.15 operator+() [1/2]	149
15.43.3.16 operator+() [2/2]	149
15.43.3.17 operator+=()	149
15.43.3.18 operator-() [1/2]	150
15.43.3.19 operator-() [2/2]	150
15.43.3.20 operator-==()	150
15.43.3.21 operator/()	150
15.43.3.22 operator/==()	151
15.43.3.23 operator==()	151
15.43.3.24 toVector() [1/2]	151
15.43.3.25 toVector() [2/2]	152
15.43.3.26 writeToVector()	152
15.43.4 Member Data Documentation	152
15.43.4.1 theta	152
15.43.4.2 x	152
15.43.4.3 y	152
15.44 SingleLine Struct Reference	153
15.44.1 Detailed Description	153
15.44.2 Member Data Documentation	153
15.44.2.1 bdescriptor	153
15.44.2.2 darkSideGrayValue	153
15.44.2.3 descriptor	154
15.44.2.4 direction	154
15.44.2.5 endPointX	154
15.44.2.6 endPointY	154
15.44.2.7 gradientMagnitude	154
15.44.2.8 lightSideGrayValue	154
15.44.2.9 lineLength	155
15.44.2.10 linePointX	155
15.44.2.11 linePointY	155
15.44.2.12 numOfPixels	155
15.44.2.13 rho	155
15.44.2.14 startPointX	155
15.44.2.15 startPointY	156
15.44.2.16 theta	156

15.44.2.17 width	156
15.45 tgraph::topmapprocessor Class Reference	156
15.45.1 Detailed Description	157
15.45.2 Constructor & Destructor Documentation	157
15.45.2.1 topmapprocessor()	157
15.45.3 Member Function Documentation	157
15.45.3.1 getbranchindx()	157
15.45.3.2 getdirflag()	157
15.45.3.3 getindexofcurredge()	158
15.45.3.4 getobsavoiodflag()	158
15.45.3.5 getweightofcurredge()	158
15.45.3.6 kimshow()	158
15.45.3.7 localizeintopomap()	159
15.45.3.8 process()	159
15.45.3.9 setNavigationPtr()	159
15.45.3.10 setpathkim() [1/2]	160
15.45.3.11 setpathkim() [2/2]	160
15.46 tgraph::topograph Struct Reference	160
15.46.1 Detailed Description	161
15.46.2 Member Data Documentation	161
15.46.2.1 directionname	161
15.46.2.2 nbranch	161
15.46.2.3 node	161
15.46.2.4 tag	161
15.47 vpControl Class Reference	162
15.47.1 Detailed Description	162
15.47.2 Constructor & Destructor Documentation	162
15.47.2.1 vpControl()	162
15.47.2.2 ~vpControl()	162
15.47.3 Member Function Documentation	163
15.47.3.1 applyJointVelocity()	163
15.47.3.2 getJointValues()	163
15.47.3.3 printTime()	163
15.47.3.4 setDesJointVelocity()	163
15.47.3.5 setOneDesJointVelocity()	163
15.47.3.6 setTask()	164
15.47.3.7 start()	164
15.47.3.8 stop()	164
15.47.3.9 stopJoint()	164
15.48 vpNaoqiGrabber Class Reference	164
15.48.1 Detailed Description	166
15.48.2 Constructor & Destructor Documentation	166
15.48.2.1 vpNaoqiGrabber()	166
15.48.2.2 ~vpNaoqiGrabber()	166

15.48.3 Member Function Documentation	166
15.48.3.1 <code>acquire()</code> [1/2]	167
15.48.3.2 <code>acquire()</code> [2/2]	167
15.48.3.3 <code>acquiredepth()</code> [1/2]	167
15.48.3.4 <code>acquiredepth()</code> [2/2]	167
15.48.3.5 <code>acquirevoxel()</code>	168
15.48.3.6 <code>cleanup()</code>	168
15.48.3.7 <code>getCameraName()</code>	168
15.48.3.8 <code>getHeight()</code>	168
15.48.3.9 <code>getProxy()</code>	168
15.48.3.10 <code>getWidth()</code>	169
15.48.3.11 <code>open()</code>	169
15.48.3.12 <code>setCamera()</code>	169
15.48.3.13 <code>setCameraParameter()</code>	169
15.48.3.14 <code>setCameraResolution()</code>	170
15.48.3.15 <code>setFramerate()</code>	170
15.48.4 Member Data Documentation	171
15.48.4.1 <code>m_camerald</code>	171
15.48.4.2 <code>m_cameraMulti</code>	171
15.48.4.3 <code>m_cameraName</code>	171
15.48.4.4 <code>m_colorspace</code>	171
15.48.4.5 <code>m_colrspaceconvop</code>	171
15.48.4.6 <code>m_CV_imtype</code>	172
15.48.4.7 <code>m_fps</code>	172
15.48.4.8 <code>m_handle</code>	172
15.48.4.9 <code>m_height</code>	172
15.48.4.10 <code>m_img</code>	172
15.48.4.11 <code>m_isOpen</code>	173
15.48.4.12 <code>m_pepper</code>	173
15.48.4.13 <code>m_pMemory</code>	173
15.48.4.14 <code>m_pVideo</code>	173
15.48.4.15 <code>m_resolution</code>	173
15.48.4.16 <code>m_robotType</code>	174
15.48.4.17 <code>m_width</code>	174
15.49 <code>vpNaoqiRobot</code> Class Reference	174
15.49.1 Detailed Description	175
15.49.2 Member Enumeration Documentation	175
15.49.2.1 <code>RobotType</code>	175
15.49.3 Constructor & Destructor Documentation	176
15.49.3.1 <code>vpNaoqiRobot()</code>	176
15.49.3.2 <code>~vpNaoqiRobot()</code>	176
15.49.4 Member Function Documentation	176
15.49.4.1 <code>cleanup()</code>	176
15.49.4.2 <code>getAngles()</code> [1/2]	176

15.49.4.3	getAngles() [2/2]	177
15.49.4.4	getBodyNames()	177
15.49.4.5	getJointVelocity()	178
15.49.4.6	getLimits()	178
15.49.4.7	getMotionProxy()	178
15.49.4.8	getPosition() [1/2]	179
15.49.4.9	getPosition() [2/2]	179
15.49.4.10	getRobotName()	179
15.49.4.11	getRobotType()	179
15.49.4.12	moveTo()	179
15.49.4.13	open()	180
15.49.4.14	rotate180()	181
15.49.4.15	setBaseVelocity() [1/2]	181
15.49.4.16	setBaseVelocity() [2/2]	181
15.49.4.17	setCollisionProtection()	182
15.49.4.18	setExternalCollisionProtectionEnabled()	182
15.49.4.19	setPosition() [1/2]	182
15.49.4.20	setPosition() [2/2]	183
15.49.4.21	setStiffness() [1/3]	183
15.49.4.22	setStiffness() [2/3]	183
15.49.4.23	setStiffness() [3/3]	184
15.49.4.24	setVelocity()	184
15.49.4.25	startPepperControl()	184
15.49.4.26	stop() [1/2]	184
15.49.4.27	stop() [2/2]	185
15.49.4.28	stopBase()	185
15.49.4.29	stopPepperControl()	185
15.49.5	Member Data Documentation	185
15.49.5.1	m_collisionProtection	185
15.49.5.2	m_isOpen	186
15.49.5.3	m_pepper_control	186
15.49.5.4	m_pMemory	186
15.49.5.5	m_pMotion	186
15.49.5.6	m_robotName	186
15.49.5.7	m_robotType	187
15.50	astar::weight_writer< WeightMap > Class Template Reference	187
15.50.1	Detailed Description	187
15.50.2	Constructor & Destructor Documentation	187
15.50.2.1	weight_writer()	187
15.50.3	Member Function Documentation	187
15.50.3.1	operator()()	187

16 File Documentation

188

16.1	cmake/BIAS_CMakeLists.txt File Reference	188
------	--	-----

16.2 cmake/libqgv_CMakeLists.txt File Reference	188
16.2.1 Function Documentation	188
16.2.1.1 find_package()	188
16.2.1.2 SET()	188
16.2.1.3 TARGET_LINK_LIBRARIES()	189
16.3 CMakeLists.txt File Reference	189
16.3.1 Function Documentation	189
16.3.1.1 cmake_minimum_required()	189
16.3.1.2 endif()	189
16.3.1.3 option()	189
16.3.2 Variable Documentation	190
16.3.2.1 lapack	190
16.4 depthnav/CMakeLists.txt File Reference	190
16.4.1 Function Documentation	190
16.4.1.1 cmake_minimum_required()	190
16.5 pepper_qi/CMakeLists.txt File Reference	190
16.5.1 Function Documentation	190
16.5.1.1 cmake_minimum_required()	190
16.6 data/README.md File Reference	191
16.7 depthnav/README.md File Reference	191
16.8 linenav/README.md File Reference	191
16.9 mapping/README.md File Reference	191
16.10 navmain/maingui/README.md File Reference	191
16.11 navmain/README.md File Reference	191
16.12 pepper_qi/README.md File Reference	191
16.13 qgv/README.md File Reference	191
16.14 README.md File Reference	191
16.15 depthnav/alpose2d.cpp File Reference	191
16.16 depthnav/alpose2d.h File Reference	192
16.16.1 Macro Definition Documentation	193
16.16.1.1 _LIBALMATH_ALMATH_TYPES_ALPOSE2D_H_	193
16.17 depthnav/depth_traits.h File Reference	194
16.18 depthnav/DepthImageScanner.cpp File Reference	195
16.19 depthnav/DepthImageScanner.h File Reference	195
16.20 depthnav/depthnav_Pepper.cpp File Reference	196
16.20.1 Function Documentation	196
16.20.1.1 getPepperRT()	196
16.20.1.2 main()	197
16.20.1.3 processdepth()	197
16.20.1.4 QI_REGISTER_MT_OBJECT()	197
16.20.2 Variable Documentation	197
16.20.2.1 eventraised	197
16.20.2.2 robo	197
16.21 depthnav/freespacenavigation.cpp File Reference	198

16.22 depthnav/freespacenavigation.h File Reference	198
16.23 depthnav/pepperlaser.h File Reference	199
16.24 linenav/dispnav.cpp File Reference	199
16.25 linenav/dispnav.h File Reference	200
16.26 linenav/EDLineDetector.cpp File Reference	201
16.26.1 Macro Definition Documentation	202
16.26.1.1 DownDir	202
16.26.1.2 Horizontal	202
16.26.1.3 LeftDir	202
16.26.1.4 RightDir	202
16.26.1.5 SkipEdgePoint	202
16.26.1.6 TryTime	203
16.26.1.7 UpDir	203
16.26.1.8 Vertical	203
16.26.2 Function Documentation	203
16.26.2.1 writeMat()	203
16.27 linenav/EDLineDetector.hh File Reference	204
16.27.1 Macro Definition Documentation	205
16.27.1.1 log_gamma	205
16.27.1.2 M_LN10	205
16.27.1.3 RELATIVE_ERROR_FACTOR	205
16.27.2 Typedef Documentation	205
16.27.2.1 PixelChain	205
16.28 linenav/kimread.cpp File Reference	206
16.29 linenav/kimread.h File Reference	206
16.30 linenav/LineDescriptor.cpp File Reference	207
16.30.1 Macro Definition Documentation	207
16.30.1.1 NO_OF_OCTAVES	208
16.30.1.2 SaliencyScale	208
16.31 linenav/LineDescriptor.hh File Reference	208
16.32 linenav/linematch.cpp File Reference	209
16.33 linenav/linematch.h File Reference	209
16.34 linenav/linenavigation.cpp File Reference	210
16.35 linenav/linenavigation.h File Reference	210
16.36 linenav/LineStructure.hh File Reference	211
16.36.1 Typedef Documentation	212
16.36.1.1 Lines_list	212
16.36.1.2 LinesVec	212
16.36.1.3 ScaleLines	213
16.37 linenav/PairwiseLineMatching.cpp File Reference	213
16.37.1 Macro Definition Documentation	214
16.37.1.1 AcceptableAngleHistogramDifference	214
16.37.1.2 AcceptableLengthVectorDifference	214
16.37.1.3 AngleDifferenceThreshold	214

16.37.1.4 DescriptorDifThreshold	214
16.37.1.5 Inf	214
16.37.1.6 IntersectionRationDifThreshold	214
16.37.1.7 LengthDifThreshold	215
16.37.1.8 ProjectionRationDifThreshold	215
16.37.1.9 RelativeAngleDifferenceThreshold	215
16.37.1.10 ResolutionScale	215
16.37.1.11 WeightOfMeanEigenVec	215
16.37.2 Function Documentation	215
16.37.2.1 normalize()	215
16.37.2.2 normL2()	216
16.38 linenav/PairwiseLineMatching.hh File Reference	216
16.38.1 Typedef Documentation	217
16.38.1.1 DISMAP	217
16.38.1.2 EigenMAP	217
16.38.1.3 Nodes_list	217
16.39 mapping/generate_configfile.m File Reference	217
16.39.1 Function Documentation	219
16.39.1.1 addpath()	219
16.39.1.2 branch()	219
16.39.1.3 DAMAGES()	220
16.39.1.4 elseif() [1/5]	220
16.39.1.5 elseif() [2/5]	220
16.39.1.6 elseif() [3/5]	220
16.39.1.7 elseif() [4/5]	220
16.39.1.8 elseif() [5/5]	220
16.39.1.9 fprintf() [1/4]	221
16.39.1.10 fprintf() [2/4]	221
16.39.1.11 fprintf() [3/4]	221
16.39.1.12 fprintf() [4/4]	221
16.39.1.13 if() [1/4]	221
16.39.1.14 if() [2/4]	221
16.39.1.15 if() [3/4]	222
16.39.1.16 if() [4/4]	222
16.39.1.17 matches()	222
16.39.1.18 system()	222
16.39.1.19 TORT()	222
16.39.2 Variable Documentation	222
16.39.2.1 all	222
16.39.2.2 bb	223
16.39.2.3 Belinda	223
16.39.2.4 bf	223
16.39.2.5 Cartman	223
16.39.2.6 clc	223

16.39.2.7 CONTRACT	224
16.39.2.8 d	224
16.39.2.9 df	224
16.39.2.10 DIRECT	224
16.39.2.11 dr	224
16.39.2.12 e	224
16.39.2.13 edgelist	225
16.39.2.14 edges	225
16.39.2.15 end	225
16.39.2.16 EXEMPLARY	225
16.39.2.17 flagx	225
16.39.2.18 Harvey	226
16.39.2.19 hd1	226
16.39.2.20 hd2	226
16.39.2.21 hd3	226
16.39.2.22 i	227
16.39.2.23 INCIDENTAL	227
16.39.2.24 INCLUDING	227
16.39.2.25 INDIRECT	227
16.39.2.26 j	227
16.39.2.27 kl	228
16.39.2.28 L	228
16.39.2.29 LIABILITY	228
16.39.2.30 linect	228
16.39.2.31 Manipulation	228
16.39.2.32 nodes	228
16.39.2.33 nodesname	229
16.39.2.34 Peter	229
16.39.2.35 Rob1	229
16.39.2.36 ROb2	229
16.39.2.37 Robotics	229
16.39.2.38 s	229
16.39.2.39 sf	230
16.39.2.40 SPECIAL	230
16.39.2.41 status	230
16.39.2.42 TO	230
16.39.2.43 WARRANTIES	230
16.40 mapping/getImages.m File Reference	230
16.40.1 Variable Documentation	231
16.40.1.1 function	231
16.41 mapping/select_ReferencelImages.m File Reference	231
16.41.1 Function Documentation	233
16.41.1.1 addpath() [1/2]	233
16.41.1.2 addpath() [2/2]	233

16.41.1.3 DAMAGES()	233
16.41.1.4 disp()	233
16.41.1.5 dlmwrite() [1/3]	234
16.41.1.6 dlmwrite() [2/3]	234
16.41.1.7 dlmwrite() [3/3]	234
16.41.1.8 if() [1/3]	234
16.41.1.9 if() [2/3]	234
16.41.1.10 if() [3/3]	234
16.41.1.11 imwrite()	235
16.41.1.12 length()	235
16.41.1.13 save() [1/3]	235
16.41.1.14 save() [2/3]	235
16.41.1.15 save() [3/3]	235
16.41.1.16 system()	235
16.41.1.17 TORT()	236
16.41.2 Variable Documentation	236
16.41.2.1 aaa	236
16.41.2.2 all	236
16.41.2.3 bf	236
16.41.2.4 clc	236
16.41.2.5 continue	236
16.41.2.6 CONTRACT	237
16.41.2.7 cr	237
16.41.2.8 ct	237
16.41.2.9 DIRECT	237
16.41.2.10 EXEMPLARY	237
16.41.2.11 folder	238
16.41.2.12 gim_t	238
16.41.2.13 im_t	238
16.41.2.14 imSize	238
16.41.2.15 INCIDENTAL	238
16.41.2.16 INCLUDING	238
16.41.2.17 INDIRECT	239
16.41.2.18 indxmatch	239
16.41.2.19 inliers	239
16.41.2.20 j	239
16.41.2.21 KFD	239
16.41.2.22 lc	239
16.41.2.23 LIABILITY	240
16.41.2.24 lines	240
16.41.2.25 linesmatch	240
16.41.2.26 linesprev	240
16.41.2.27 linest	240
16.41.2.28 m	240

16.41.2.29 matchedlines	241
16.41.2.30 matchindex	241
16.41.2.31 mp	241
16.41.2.32 pr	241
16.41.2.33 sfc	241
16.41.2.34 SPECIAL	241
16.41.2.35 status	242
16.41.2.36 thres	242
16.41.2.37 TO	242
16.41.2.38 WARRANTIES	242
16.42 navmain/maingui/navinside.cpp File Reference	242
16.43 navmain/maingui/navinside.h File Reference	243
16.44 navmain/maingui/navwindow.cpp File Reference	243
16.45 navmain/maingui/navwindow.h File Reference	244
16.46 navmain/nav_pepper_inside.cpp File Reference	244
16.46.1 Function Documentation	245
16.46.1.1 main()	245
16.46.1.2 usage()	245
16.47 navmain/nav_peppergui.cpp File Reference	246
16.47.1 Function Documentation	246
16.47.1.1 main()	246
16.47.1.2 usage()	246
16.48 navmain/nav_pepperoffline.cpp File Reference	247
16.48.1 Function Documentation	247
16.48.1.1 main()	247
16.48.1.2 usage()	247
16.49 navmain/nav_pepperonline.cpp File Reference	248
16.49.1 Function Documentation	248
16.49.1.1 currentDateTime()	248
16.49.1.2 main()	249
16.49.1.3 usage()	249
16.50 navmain/navigation.cpp File Reference	249
16.51 navmain/navigation.h File Reference	249
16.52 navmain/pepperevents.h File Reference	250
16.52.1 Function Documentation	251
16.52.1.1 QI_REGISTER_OBJECT()	251
16.53 navmain/pepperInterface.cpp File Reference	251
16.54 navmain/pepperInterface.h File Reference	252
16.55 navmain/peppernavigation.h File Reference	252
16.56 navmain/pepperOffline.cpp File Reference	253
16.57 navmain/pepperOffline.h File Reference	253
16.58 navmain/pepperRobot.cpp File Reference	254
16.59 navmain/pepperRobot.h File Reference	255
16.60 navmain/pepperRobotVirtual.cpp File Reference	255

16.61 navmain/pepperRobotVirtual.h File Reference	256
16.62 navmain/topograph_astar.h File Reference	256
16.63 navmain/topograph_processor.h File Reference	258
16.64 pepper_qi/apps/main.cpp File Reference	259
16.64.1 Function Documentation	259
16.64.1.1 main()	259
16.65 pepper_qi/include/vpControl.hpp File Reference	259
16.65.1 Function Documentation	260
16.65.1.1 QI_REGISTER_OBJECT()	260
16.66 pepper_qi/include/vpNaoqiGrabber.h File Reference	261
16.67 pepper_qi/include/vpNaoqiRobot.h File Reference	261
16.67.1 Macro Definition Documentation	262
16.67.1.1 BOOST_SIGNALS_NO_DEPRECATION_WARNING	262
16.68 pepper_qi/src/vpControl.cpp File Reference	262
16.68.1 Macro Definition Documentation	263
16.68.1.1 DELTAQMIN	263
16.68.1.2 FLAGACC	263
16.68.1.3 FLAGCTE	263
16.68.1.4 FLAGDEC	263
16.68.1.5 FLAGSTO	263
16.68.1.6 OFFSET_BUTEE	264
16.69 pepper_qi/src/vpNaoqiGrabber.cpp File Reference	264
16.70 pepper_qi/src/vpNaoqiRobot.cpp File Reference	264

Chapter 1

Pepper_navigation

This repository contains the source code accompanying our paper Image-Based Indoor Topological Navigation with Collision Avoidance for Resource-Constrained Mobile Robots as discussed here [here](#) [1].

The code is open source. The codes are provided “as-is” without any warranty. Before using the code, you agree to use the code at your own risk. The authors are not responsible or liable for any damages incurred using this code.

1.1 Dependencies

1. [MATLAB](#) (for mapping, has to be downloaded separately),
2. [OpenCV](#) (available in ppa of Ubuntu or has to be built from [source](#) with [extra modules](#)),
3. [Boost](#) (available in ppa of Ubuntu or has to be built from [source](#)),
4. [naoqi-libqi](#) and [naoqi-libqicore](#) (ROS Packages available for Kinetic and Melodic),
5. [ARPACK++](#), [ARPACK](#), [BLAS](#), [LAPACK](#), and [SuperLU](#) (for line segment matching, available in ppa of Ubuntu),
6. [Qt5](#) ([Core](#), [Widgets](#), [Test](#)) (for GUI, available in ppa of Ubuntu),
7. [Graphviz](#) ([GVC](#), [CGRAPH](#), [CDT](#)) (for visualising topological graph, available in ppa of Ubuntu),
8. [Pepper Virtual Machine](#) and [VirtualBox](#) (to compile binaries that can run onboard on the Pepper Robot, has to be downloaded separately),
9. [qgv](#) and [pepper_qi](#) (custom libraries that are shipped with this repository).

1.2 Build Instructions

1. Get Source codes from the repository.

```
$ git clone https://github.com/suuman/pepper\_navigation.git
```

2. Install required dependencies (if required).

```
$ sudo apt install libopencv-dev libopencv-contrib-dev libboost-all-dev  
$ sudo apt install ros-melodic-naoqi-libqi ros-melodic-naoqi-libqicore  
$ sudo apt install libarpack++2-dev libarpack2-dev libblas-dev liblapack-dev libsuperlu-dev  
  
$ sudo apt install qt5-default  
$ sudo apt install libcddt5 libcgraph6 libcgcv6 libgraphviz-dev
```

3. Build executables required for mapping.

```
$ cd pepper_navigation/mapping
$ ./build_linematching.sh
```

The above command will build executables **detectinesED** and **matchlines** to `./linematching` folder and is equivalent to the following commands

```
$ cd linematching/Linematching_iso && mkdir build $$ cd build
$ cmake .. && make -j8
```

4. Build Navigation code

```
$ cd pepper_navigation
$ ./compile.sh
```

The above command will build executables in the `build/` directory and is equivalent to the following commands

```
$ mkdir build $$ cd build
$ cmake .. && make -j8
```

If the CMake options and Path has to be changed from the default

```
$ ccmake .. or cmake-gui ..
Configure and Generate Makefile.
$ make -j8
```

Following executables will be built

1. **peppernav_gui** => GUI version of the topological navigation of the Pepper robot. It has to be executed on the external PC connecting to the Pepper robot remotely.
2. **peppernav_inside** => Topological navigation of the Pepper robot. It is capable of running onboard on the Pepper robot.
3. **peppernavigation** => Navigation of the Pepper robot along the single reference image list. It is capable of running onboard on Pepper robot.
4. **peppernavigationoff** => Offline localisation mode from the image sequence. It does not require a robot (Images read from the folder).

Note: To run the executables **peppernav_inside** and **peppernavigation** onboard on the Pepper robot, they must be compiled in the **Pepper Virtual Machine**. For details, please refer to the **Reference Manual** in `../docs` folder.

1.3 Run Code

Example Scripts: `run_onboard.sh`, `run_remotePC.sh`, and `run_offline.sh`.

The navigation executables require two arguments:

- a. Path of topological map (e.g. `../data/tmap`), and
- b. IP address of the robot (e.g. `172.19.226.236` to run online with robot) or path of the image sequence (e.g. `../data/offlinetest`) to run offline with image sequence (localisation mode only).

For details, please refer to **docs/Pepper_Navigation_Reference_Manual.pdf** and **README.MD** at sub-folders of this repository.

1.4 References

- [1] Bista SR, Ward B, Corke P. Image-Based Indoor Topological Navigation with Collision Avoidance for Resource-Constrained Mobile Robots. *Journal of Intelligent & Robotic Systems*. 102,55(2021). <http://dx.doi.org/10.1007/s10846-021-01390-6>
- [2] Bista SR, Giordano PR, Chaumette F. Appearance-based indoor navigation by IBVS using line segments. *IEEE Robotics and Automation Letters*. 2016 Jan 26;1(1):423-30. <https://doi.org/10.1109/LRA.2016.2521907>

Chapter 2

navmain

Top-level interface for the navigation.

2.1 Dependencies

- `OpenCV`,
- `Boost`,
- `naoqi_libqi`, `naoqi_libqicore` and `pepper_qi` (Please refer to `../pepper_qi` folder for details),
- `Qt5`, `qgv` and `Graphviz` (Please refer to `../qgv` folder for details).

2.2 Usage

2.2.1 Libraries

1. **`navigation`** (`navigation.h`, `navigation.cpp`) => Top-level interface for image-based navigation using line segments (`../linenav`).
2. **`peppernavigation`** (`peppernavigation.h`) => Top-level Interface for the Pepper robot navigation. Top-level functions for `free-space navigation` and `image-based navigation`.
3. **`pepperInterface`** (`pepperInterface.h`, `pepperInterface.cpp`) => Virtual Class with top-level virtual functions for interface with the Pepper robot or just image sequences.
 - **`pepperRobot`** (`pepperRobot.h`, `pepperRobot.cpp`) => Derived Class to interface with the Pepper robot.
 - **`pepperRobotVirtual`** (`pepperRobotVirtual.h`, `pepperRobotVirtual.cpp`) => Derived Class to test **`pepperRobot`** functionalities via Virtual Pepper where images are read from the folder.
 - **`pepperOffline`** (`pepperOffline.h`, `pepperOffline.cpp`) => Derived Class to interface in offline mode i.e. read images from the folder and perform image-based localisation only (Pepper robot not used).
4. **`topmapprocessor`** (`topograph_processor.h`) => Top-level interface for processing topological map (reads topological graph and reference images from disk for navigation).
5. **`astar`** (`topograph_astar.h`) => Performs A* search in the graph. Modified code from `Boost Graph Library example : astar-cities`.
6. **`pepperServices`** (`pepperevents.h`) => Subscribe to the Pepper robot's internal events related to move and collision.

2.2.2 Executables

1. ***peppernav_gui*** (*nav_peppergui.cpp*) => Topological navigation of the Pepper robot with GUI. The code has to be executed on the external PC connecting to the Pepper robot remotely.
2. ***peppernav_inside*** (*nav_pepper_inside.cpp*) => Topological navigation of the Pepper robot. This code is capable to run onboard on the Pepper robot.
3. ***peppernavigation*** (*nav_pepperonline.cpp*) => Image-based navigation of the Pepper robot along a single sequence of reference images.
4. ***peppernavigationoff*** (*nav_pepperoffline.cpp*) => Image-based localisation along the reference image list. This is the offline mode that uses image sequence and does not require the robot.

Note: The executables *peppernav_inside* and *peppernavigation* are capable of running onboard on the Pepper robot provided that they are compiled in the **Pepper Virtual Machine** [1]. For details, please refer to the **Reference Manual** in *../docs* folder.

[1] G. Suddrey, A. Jacobson and B. Ward. "Enabling a pepper robot to provide automated and interactive tours of a robotics laboratory." arXiv preprint [arXiv:1804.03288](https://arxiv.org/abs/1804.03288) (2018). *ACRA 2018 Proceedings*. https://bitbucket.org/pepper_qut/virtual-machine.git.

Chapter 3

navmain/maingui

- Interface for Topological Navigation.
- The GUI version of topological navigation runs on a remote PC. The code communicates with the Pepper robot remotely via the `naoqi` interface.
- The Non-GUI version of topological navigation is capable of running onboard on the Pepper robot if the code is compiled in the `Pepper Virtual Machine`.

Topological Navigation

navwindow (`navwindow.h`, `navwindow.cpp`, `navwindow.ui`) => Navigation with GUI control. Requires Qt5 and qgv libraries.

navinside (`navinside.h`, `navinside.cpp`) => Navigation without GUI. This version can be used to run navigation onboard on the Pepper robot.

Please refer to `../..CMakeLists.txt`.

window_QT and files_Qt

Uses OpenCV for displaying the image in QT UI.

Code taken from `OpenCV highgui module repository`.

Usage

Please refer to the `Reference Manual` in `../docs` folder.

Chapter 4

depthnav

4.1 Free-space navigation using depth image

Navigation in the drivable free-space using a 2D occupancy grid map obtained from the depth image.

4.2 Dependencies

- `OpenCV`,
- `Boost`,
- `naoqi_libqi`, `naoqi_libqicore` and `pepper_qi` (Please refer to `../pepper_qi` folder for details).

4.3 Usage

I. Standalone usage

Refer `depthnav_Pepper.cpp`

```
$ mkdir build $$ cd build
$ cmake .. && make -j8
$ ./pepper_fsnave --ip < Pepper_IP >
```

II. As library

1. To use the depthnav library, please refer to class **`freespacenavigation`** (`freespacenavigation.h`, `freespacenavigation.cpp`).
2. Other classes and functions:
 - **`depthimagescanner`** (`DepthImageScanner.h`, `DepthImageScanner.cpp`) => Creates a 2D grid map from the depth image.
 - **`depth_traits`** (`depth_traits.h`) => Template function to process the depth image obtained from `ros depthimage_to_laserscan` library.
 - **`pepperlaser`** (`pepperlaser.h`) => Defines the Pepper robot's Laser Memory Keys.
 - **`alpose2d`** (`alpose2d.h`, `alpose2d.cpp`) => `libalmath` Pose2D library used to process odometry data from the Pepper robot.

Chapter 5

linenav

5.1 Image-based navigation using line segments

https://github.com/suuman/line_navigation_offline.git

Bista SR, Giordano PR, Chaumette F. Appearance-based indoor navigation by IBVS using line segments. IEEE Robotics and Automation Letters. 2016 Jan 26;1(1):423-30.

5.2 Dependencies

1. `OpenCV`,
2. `Line matching code` is based on <http://www.mip.informatik.uni-kiel.de/tiki-download%20file.php?fileId=1965> (offline now but available in OpenCV),
1. `BIAS library` is based on <http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=BIAS> (offline now),
2. `ARPACK++`, `ARPACK`, `BLAS`, `LAPACK`, `SuperLU`.

5.3 Usage

1. To use the original line detection and matching code based on the legacy BIAS library (shipped with this repository [here](#)), use the codes inside the `edlbd/` folder.
If you do not want to use the legacy BIAS library, the line detection and matching based on OpenCV is used.
Please refer to `../CMakeLists.txt`.
2. To use **`linenavigation`** as a library, please refer to `linenavigation.h` and `linenavigation.cpp`.
3. To display the image-based localisation via **`dispnav`** class, please refer to `dispnav.h` and `dispnav.cpp`.
4. For the usage of **`linenavigation`** and **`dispnav`**, please refer to `navigation.h` and `navigation.cpp` in `../navmain/` directory.

Chapter 6

Mapping

6.0.1 Selection of Reference Images Based on the Line Segment Matching

Bista SR, Giordano PR, Chaumette F. Appearance-based indoor navigation by IBVS using line segments. IEEE Robotics and Automation Letters. 2016 Jan 26;1(1):423-30.

<https://github.com/suuman/selectKeyImagesLines.git>

6.1 Build

1. Get Source codes from the repository.

```
$ git clone https://github.com/suuman/selectKeyImagesLines.git
```

2. Build executables required for mapping.

```
$ ./build_linematching.sh
```

The above command will build executables **detectinesED** and **matchlines** to *./linematching* folder and is equivalent to the following commands:

```
$ cd linematching/Linematching_iso && mkdir build $$ cd build  
$ cmake .. && make -j8
```

6.2 Usage

The code for the mapping is in MATLAB. Make sure the line detection and matching codes have been properly built and executables have been placed in the correct folder.

For selecting the reference images, we need to provide the path of the image sequence folder and the folder to store the reference images.

1. Open selectRefImages.m
2. Set the path of the image sequence. e.g imseq = '../roboroom/imgs_acquired'
3. Set the path to store the reference images. e.g refimpath='../roboroom/ref_imgs'
4. Run selectRefImages.m

The reference image folder will contain

1. Reference Images.
2. The text files with the detected line segments and their descriptors. There is one .txt file for each reference image.

6.3 Creating Topological Map for Navigation

Bista SR, Ward B, Corke P. Image-Based Indoor Topological Navigation with Collision Avoidance for Resource-Constrained Mobile Robots. Journal of Intelligent & Robotic Systems. 102,55(2021).

To build a topological map from reference images, please refer to the [Reference Manual](#) in `../docs` folder.

`genConfigFile.m` generates *conf.txt* that contains the link between the reference images among the adjacent edges.

Chapter 7

pepper_qi

7.1 Interface with **naoqi** library

This source code has been taken and modified for the Pepper robot.

The original code is available from

- https://github.com/lagadic/visp_aoqi/tree/libqi
- https://github.com/lagadic/pepper_control/tree/libqi

Reference of the original code

E. Marchand, F. Spindler, F. Chaumette. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics and Automation Magazine, Special Issue on “Software Packages for Vision-Based Control of Motion”, P. Oh, D. Burschka (Eds.), 12(4):40-52, December 2005. DOI:10.1109/MRA.2005.1577023. inria-00351899. <https://visp.inria.fr/>.

This modified code supports image acquisition from the depth camera.

7.2 Dependencies

- OpenCV,
- Boost,
- **aoqi_libqi** (ROS library),
- **aoqi_libqicore** (ROS library).

Note:

- The thirdparty folder consists of source code of **aoqi_libqi** and **aoqi_libqicore** that has been modified to work with Boost version 1.65.1.
- The latest code of **libqi** and **qicore** can be obtained from
<https://github.com/aldebaran/libqi>
<https://github.com/aldebaran/libqicore>

Chapter 8

qgv

8.1 Interactive Qt GraphViz display

Interactive Qt GraphViz display library used to display the topological graph in [the GUI version of the source code](#).

Taken from <https://github.com/nbergont/qgv>

8.2 Dependencies

1. `Qt5 (Core, Widgets, Test)`
2. `Graphviz (GVC, CGRAPH, CDT)`

8.3 Usage

Please refer to `libqgv_CMakeLists.txt` in the `../cmake` folder for the usage of the qgv library in this project.

Chapter 9

Sample data for testing

9.1 offlinetest/

Contains data for offline testing (mapping and localisation).

1. *offlinetest/imgs* => contains image sequence obtained from the Pepper robot.
2. *offlinetest/kfls* => contains reference images obtained from *select_ReferenceImages.m* located in the *../mapping* folder.

9.2 tmap/

Contains topological map required for navigation.

tmap/conf.txt => configuration file that defines the topological map.
This file is generated from *generate_configfile.m* located in the *../mapping* folder.

Chapter 10

Namespace Index

10.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AL	21
AL::Math	21
astar	25
depthimagescanner	27
tgraph	27
Ui	27

Chapter 11

Hierarchical Index

11.1 Class Hierarchy

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

astar::astar	28
astar_heuristic	
astar::distance_heuristic< Graph, CostType, LocMap >	48
astar::heuristic< Graph, CostType, LocMap >	62
tgraph::branch	33
CompareL	34
CompareS	35
default_astar_visitor	
astar::astar_goal_visitor< Vertex >	31
depthimagescanner::DepthImageScanner	35
depthimagescanner::DepthTraits< T >	39
depthimagescanner::DepthTraits< float >	39
depthimagescanner::DepthTraits< uint16_t >	40
dispNav	42
tgraph::edge	50
EdgeChains	51
EDLineDetector	52
EDLineParam	56
astar::found_goal	58
freespacenavigation	58
astar::graph_writer< Name, LocMap >	61
kimRead	64
LineChains	73
LineDescriptor	74
linematch	78
linenavigation	80
astar::location	88
Matrix< T >	89
MyService	91
navigation	92
NavInside	96
tgraph::node	102
Node	103
OctaveLine	104
OctaveSingleLine	105
PairwiseLineMatching	108
pepperInterface	109
pepperOffline	119

pepperRobot	125
pepperRobotVirtual	132
pepperNavigation	113
pepperServices	139
Pixel	141
AL::Math::Pose2D	141
QMainWindow	
NavWindow	99
SingleLine	153
tgraph::topmapprocessor	156
tgraph::topograph	160
vpControl	162
vpNaoqiGrabber	164
vpNaoqiRobot	174
astar::weight_writer< WeightMap >	187

Chapter 12

Class Index

12.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

astar::astar	
The astar class	28
astar::astar_goal_visitor< Vertex >	31
tgraph::branch	33
CompareL	34
CompareS	35
depthimagescanner::DepthImageScanner	35
depthimagescanner::DepthTraits< T >	39
depthimagescanner::DepthTraits< float >	39
depthimagescanner::DepthTraits< uint16_t >	40
dispNav	
The dispNav class	42
astar::distance_heuristic< Graph, CostType, LocMap >	48
tgraph::edge	50
EdgeChains	51
EDLineDetector	52
EDLineParam	56
astar::found_goal	58
freespacenavigation	58
astar::graph_writer< Name, LocMap >	61
astar::heuristic< Graph, CostType, LocMap >	62
kimRead	64
LineChains	73
LineDescriptor	74
linematch	78
linenavigation	80
astar::location	88
Matrix< T >	89
MyService	91
navigation	
The navigation class	92
NavInside	96
NavWindow	99
tgraph::node	
The node struct each node has nodeid starting from 0. nodename : higher level id for node	102
Node	103
OctaveLine	104
OctaveSingleLine	105

PairwiseLineMatching	108
pepperInterface	109
pepperNavigation	
The pepperNavigation class	113
pepperOffline	119
pepperRobot	125
pepperRobotVirtual	132
pepperServices	
The pepperServices class	139
Pixel	141
AL::Math::Pose2D	
A pose in a 2-dimentional space	141
SingleLine	153
tgraph::topmapprocessor	
The topmapprocessor class	156
tgraph::topograph	
The topograph struct	160
vpControl	162
vpNaoqiGrabber	164
vpNaoqiRobot	174
astar::weight_writer< WeightMap >	187

Chapter 13

File Index

13.1 File List

Here is a list of all files with brief descriptions:

depthnav/alpose2d.cpp	191
depthnav/alpose2d.h	192
depthnav/depth_traits.h	194
depthnav/DepthImageScanner.cpp	195
depthnav/DepthImageScanner.h	195
depthnav/depthnav_Pepper.cpp	196
depthnav/freespacenavigation.cpp	198
depthnav/freespacenavigation.h	198
depthnav/pepperlaser.h	199
linenav/dispnave.cpp	199
linenav/dispnave.h	200
linenav/EDLineDetector.cpp	201
linenav/EDLineDetector.hh	204
linenav/kimread.cpp	206
linenav/kimread.h	206
linenav/LineDescriptor.cpp	207
linenav/LineDescriptor.hh	208
linenav/linematch.cpp	209
linenav/linematch.h	209
linenav/linenavigation.cpp	210
linenav/linenavigation.h	210
linenav/LineStructure.hh	211
linenav/PairwiseLineMatching.cpp	213
linenav/PairwiseLineMatching.hh	216
mapping/generate_configfile.m	217
mapping/getImages.m	230
mapping/select_ReferencelImages.m	231
navmain/nav_pepper_inside.cpp	244
navmain/nav_peppergui.cpp	246
navmain/nav_pepperoffline.cpp	247
navmain/nav_pepperonline.cpp	248
navmain/navigation.cpp	249
navmain/navigation.h	249
navmain/pepperevents.h	250
navmain/pepperInterface.cpp	251
navmain/pepperInterface.h	252
navmain/peppernavigation.h	252
navmain/pepperOffline.cpp	253

navmain/pepperOffline.h	253
navmain/pepperRobot.cpp	254
navmain/pepperRobot.h	255
navmain/pepperRobotVirtual.cpp	255
navmain/pepperRobotVirtual.h	256
navmain/topograph_astar.h	256
navmain/topograph_processor.h	258
navmain/maingui/navinside.cpp	242
navmain/maingui/navinside.h	243
navmain/maingui/navwindow.cpp	243
navmain/maingui/navwindow.h	244
pepper_qi/apps/main.cpp	259
pepper_qi/include/vpControl.hpp	259
pepper_qi/include/vpNaoqiGrabber.h	261
pepper_qi/include/vpNaoqiRobot.h	261
pepper_qi/src/vpControl.cpp	262
pepper_qi/src/vpNaoqiGrabber.cpp	264
pepper_qi/src/vpNaoqiRobot.cpp	264

Chapter 14

Namespace Documentation

14.1 AL Namespace Reference

Namespaces

- [Math](#)

14.2 AL::Math Namespace Reference

Classes

- struct [Pose2D](#)
A pose in a 2-dimentional space.

Functions

- float [distanceSquared](#) (const [Pose2D](#) &pPos1, const [Pose2D](#) &pPos2)
Compute the squared distance between two [Pose2D](#).
- float [distance](#) (const [Pose2D](#) &pPos1, const [Pose2D](#) &pPos2)
Compute the distance between two [Pose2D](#).
- void [pose2DInverse](#) (const [Pose2D](#) &pPos, [Pose2D](#) &pRes)
Compute the inverse of a [Pose2D](#).
- void [pose2dInvertInPlace](#) ([Pose2D](#) &pPos)
Inverse the given [Pose2D](#) in place:
- [Pose2D pose2dDiff](#) (const [Pose2D](#) &pPos1, const [Pose2D](#) &pPos2)
Compute the [Pose2D](#) between the actual [Pose2D](#) and the one give in argument result:
- [Pose2D pose2DInverse](#) (const [Pose2D](#) &pPos)
Compute the inverse of a [Pose2D](#).
- [Pose2D pinv](#) (const [Pose2D](#) &pPos)
Alternative name for inverse: return the pose2d inverse of the given [Pose2D](#).

14.2.1 Function Documentation

14.2.1.1 distance()

```
float AL::Math::distance (
    const Pose2D & pPos1,
    const Pose2D & pPos2 )
```

Compute the distance between two [Pose2D](#).

$$\sqrt{(pPos1.x - pPos2.x)^2 + (pPos1.y - pPos2.y)^2}$$

Parameters

<i>pPos1</i>	the first Pose2D
<i>pPos2</i>	the second Pose2D

Returns

the float distance between the two [Pose2D](#)

Definition at line 174 of file `alpose2d.cpp`.

14.2.1.2 distanceSquared()

```
float AL::Math::distanceSquared (
    const Pose2D & pPos1,
    const Pose2D & pPos2 )
```

Compute the squared distance between two [Pose2D](#).

$$(pPos1.x - pPos2.x)^2 + (pPos1.y - pPos2.y)^2$$

Parameters

<i>pPos1</i>	the first Pose2D
<i>pPos2</i>	the second Pose2D

Returns

the float squared distance between the two [Pose2D](#)

Definition at line 167 of file `alpose2d.cpp`.

14.2.1.3 pinv()

```
Pose2D AL::Math::pinv (
    const Pose2D & pPos )
```

Alternative name for inverse: return the pose2d inverse of the given [Pose2D](#).

Parameters

<i>pPos</i>	the given Pose2D
-------------	----------------------------------

Definition at line 233 of file `alpose2d.cpp`.

14.2.1.4 pose2dDiff()

```
Pose2D AL::Math::pose2dDiff (
    const Pose2D & pPos1,
    const Pose2D & pPos2 )
```

Compute the [Pose2D](#) between the actual [Pose2D](#) and the one give in argument result:

$\text{inverse}(pPos1) * pPos2$

Parameters

<i>pPos1</i>	the first Pose2D
<i>pPos2</i>	the second Pose2D

Returns

the [Pose2D](#)

Definition at line 216 of file `alpose2d.cpp`.

14.2.1.5 pose2DInverse() [1/2]

```
Pose2D AL::Math::pose2DInverse (
    const Pose2D & pPos )
```

Compute the inverse of a [Pose2D](#).

Parameters

<i>pPos</i>	the initial Pose2D
-------------	------------------------------------

Returns

the inverse [Pose2D](#)

Definition at line 226 of file `alpose2d.cpp`.

14.2.1.6 pose2DInverse() [2/2]

```
void AL::Math::pose2DInverse (
    const Pose2D & pPos,
    Pose2D & pRes )
```

Compute the inverse of a [Pose2D](#).

Parameters

<i>pPos</i>	the initial Pose2D
<i>pRes</i>	the inverse Pose2D

Definition at line 196 of file `alpose2d.cpp`.

14.2.1.7 pose2dInvertInPlace()

```
void AL::Math::pose2dInvertInPlace (
    Pose2D & pPos )
```

Inverse the given [Pose2D](#) in place:

Parameters

<i>pPos</i>	the given Pose2D
-------------	----------------------------------

Definition at line 204 of file `alpose2d.cpp`.

14.3 astar Namespace Reference

Classes

- class [astar](#)
The astar class.
- class [astar_goal_visitor](#)
- class [distance_heuristic](#)
- struct [found_goal](#)
- class [graph_writer](#)
- class [heuristic](#)
- struct [location](#)
- class [weight_writer](#)

Typedefs

- typedef float [cost](#)
- typedef boost::adjacency_list< boost::listS, boost::vecS, boost::undirectedS, boost::no_property, boost::property< boost::edge_weight_t, [cost](#) > > [mygraph_t](#)
- typedef boost::property_map< [mygraph_t](#), boost::edge_weight_t >::type [WeightMap](#)
- typedef [mygraph_t](#)::vertex_descriptor [vertex](#)
- typedef [mygraph_t](#)::edge_descriptor [edge_descriptor](#)
- typedef std::pair< int, int > [edge](#)

Variables

- const typedef char * [node](#)

14.3.1 Typedef Documentation

14.3.1.1 cost

```
typedef float astar::cost
```

Definition at line 50 of file topograph_astar.h.

14.3.1.2 edge

```
typedef std::pair<int, int> astar::edge
```

Definition at line 150 of file topograph_astar.h.

14.3.1.3 edge_descriptor

```
typedef mygraph_t::edge_descriptor astar::edge_descriptor
```

Definition at line 149 of file topograph_astar.h.

14.3.1.4 mygraph_t

```
typedef boost::adjacency_list<boost::listS, boost::vecS, boost::undirectedS, boost::no_property, boost::property<boost::edge_weight_t, cost> > astar::mygraph_t
```

Definition at line 146 of file topograph_astar.h.

14.3.1.5 vertex

```
typedef mygraph_t::vertex_descriptor astar::vertex
```

Definition at line 148 of file topograph_astar.h.

14.3.1.6 WeightMap

```
typedef boost::property_map<mygraph_t, boost::edge_weight_t>::type astar::WeightMap
```

Definition at line 147 of file topograph_astar.h.

14.3.2 Variable Documentation

14.3.2.1 node

```
const typedef char* astar::node
```

Definition at line 51 of file topograph_astar.h.

14.4 depthimagescanner Namespace Reference

Classes

- class [DepthImageScanner](#)
- struct [DepthTraits](#)
- struct [DepthTraits< float >](#)
- struct [DepthTraits< uint16_t >](#)

14.5 tgraph Namespace Reference

Classes

- struct [branch](#)
- struct [edge](#)
- struct [node](#)
The node struct each node has nodeid starting from 0. nodename : higher level id for node.
- class [topmapprocessor](#)
The topmapprocessor class.
- struct [topograph](#)
The topograph struct.

14.6 Ui Namespace Reference

Chapter 15

Class Documentation

15.1 astar::astar Class Reference

The astar class.

```
#include <topograph_astar.h>
```

Public Member Functions

- [astar](#) (int num_nodes)
astar initilaise graph
- float [getweight](#) (int indx)
getweight return weight of the edge
- void [setlocationherustics](#) ([node](#) *nodename, [location](#) *locat)
setlocationherustics
- void [setedges](#) (int numedges, [edge](#) *edgearray, [cost](#) *wts)
setedges set edges of the graph
- void [creategraph](#) ()
creategraph creates topological graph from edges
- int [searchingraph](#) (int st, int gl, std::vector< int > &optpath)
searchingraph A search betwwn start node and destination node*
- int [searchingraphrandom](#) (int &st, int &gl, std::vector< int > &optpath)
searchingraphrandom A search with random start and random destination*
- void [writegraph2file](#) (std::string &fname)
writegraph2file wirte graph in file in .dot format

15.1.1 Detailed Description

The astar class.

Definition at line 156 of file topograph_astar.h.

15.1.2 Constructor & Destructor Documentation

15.1.2.1 astar()

```
astar::astar::astar (  
    int num_nodes ) [inline]
```

astar initilaise graph

Parameters

<i>num_nodes</i>	number of nodes
------------------	-----------------

Definition at line 225 of file topograph_astar.h.

15.1.3 Member Function Documentation

15.1.3.1 creategraph()

```
void astar::astar::creategraph ( ) [inline]
```

creategraph creates topological graph from edges

Definition at line 277 of file topograph_astar.h.

15.1.3.2 getweight()

```
float astar::astar::getweight (
    int indx ) [inline]
```

getweight return weight of the edge

Parameters

<i>indx</i>	index of edge
-------------	---------------

Returns

weight

Definition at line 240 of file topograph_astar.h.

15.1.3.3 searchingraph()

```
int astar::astar::searchingraph (
    int st,
    int gl,
    std::vector< int > & optpath ) [inline]
```

searchingraph A* search betwwn start node and destination node

Parameters

<i>st</i>	index of start node
<i>gl</i>	index of destination node
<i>optpath</i>	node list of the optimum path

Returns

cost of the optimum path

Definition at line 296 of file topograph_astar.h.

15.1.3.4 searchinggraphrandom()

```
int astar::astar::searchinggraphrandom (
    int & st,
    int & gl,
    std::vector< int > & optpath ) [inline]
```

searchinggraphrandom A* search with random start and random destination

Parameters

<i>st</i>	start node index => generated randomly
<i>gl</i>	end node index => generated randomly
<i>optpath</i>	node list of the optimum path

Returns

cost of the optimum path

Definition at line 315 of file topograph_astar.h.

15.1.3.5 setedges()

```
void astar::astar::setedges (
    int numedges,
    edge * edgearray,
    cost * wts ) [inline]
```

setedges set edges of the graph

Parameters

<i>numedges</i>	number of the edges
<i>edgearray</i>	edges information
<i>wts</i>	weights of edges

Definition at line 262 of file topograph_astar.h.

15.1.3.6 setlocationherustics()

```
void astar::astar::setlocationherustics (
    node * nodename,
    location * locat ) [inline]
```

setlocationherustics

Definition at line 247 of file topograph_astar.h.

15.1.3.7 writegraph2file()

```
void astar::astar::writegraph2file (
    std::string & fname ) [inline]
```

writegraph2file wirte graph in file in .dot format

Parameters

<i>fname</i>	file name
--------------	-----------

Definition at line 334 of file topograph_astar.h.

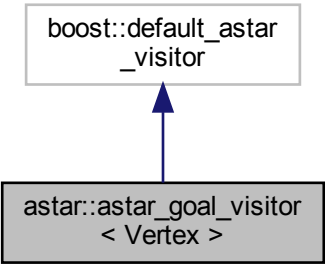
The documentation for this class was generated from the following file:

- [navmain/topograph_astar.h](#)

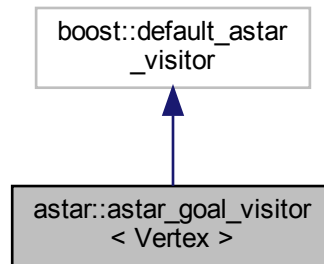
15.2 astar::astar_goal_visitor< Vertex > Class Template Reference

```
#include <topograph_astar.h>
```

Inheritance diagram for astar::astar_goal_visitor< Vertex >:



Collaboration diagram for `astar::astar_goal_visitor< Vertex >`:



Public Member Functions

- [astar_goal_visitor](#) (Vertex goal)
- `template<class Graph >`
void [examine_vertex](#) (Vertex u, Graph &g)

15.2.1 Detailed Description

```
template<class Vertex>
class astar::astar_goal_visitor< Vertex >
```

Definition at line 131 of file `topograph_astar.h`.

15.2.2 Constructor & Destructor Documentation

15.2.2.1 `astar_goal_visitor()`

```
template<class Vertex >
astar::astar_goal_visitor< Vertex >::astar_goal_visitor (
    Vertex goal ) [inline]
```

Definition at line 134 of file `topograph_astar.h`.

15.2.3 Member Function Documentation

15.2.3.1 examine_vertex()

```
template<class Vertex >
template<class Graph >
void astar::astar_goal_visitor< Vertex >::examine_vertex (
    Vertex u,
    Graph & g ) [inline]
```

Definition at line 136 of file topograph_astar.h.

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

- navmain/[topograph_astar.h](#)

15.3 tgraph::branch Struct Reference

```
#include <topograph_processor.h>
```

Public Attributes

- int [sid](#)
- int [eid](#)
- std::string [fold](#)
- int [indx](#)
- char [diflag](#)

15.3.1 Detailed Description

Definition at line 57 of file topograph_processor.h.

15.3.2 Member Data Documentation

15.3.2.1 diflag

```
char tgraph::branch::diflag
```

Definition at line 62 of file topograph_processor.h.

15.3.2.2 eid

```
int tgraph::branch::eid
```

Definition at line 59 of file topograph_processor.h.

15.3.2.3 fold

```
std::string tgraph::branch::fold
```

Definition at line 60 of file `topograph_processor.h`.

15.3.2.4 indx

```
int tgraph::branch::indx
```

Definition at line 61 of file `topograph_processor.h`.

15.3.2.5 sid

```
int tgraph::branch::sid
```

Definition at line 58 of file `topograph_processor.h`.

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

- [navmain/topograph_processor.h](#)

15.4 CompareL Struct Reference

```
#include <PairwiseLineMatching.hh>
```

Public Member Functions

- bool [operator\(\)](#) (const double &lhs, const double &rhs) const

15.4.1 Detailed Description

Definition at line 26 of file `PairwiseLineMatching.hh`.

15.4.2 Member Function Documentation

15.4.2.1 operator()

```
bool CompareL::operator() (
    const double & lhs,
    const double & rhs ) const [inline]
```

Definition at line 27 of file PairwiseLineMatching.hh.

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

- [linenav/PairwiseLineMatching.hh](#)

15.5 CompareS Struct Reference

```
#include <PairwiseLineMatching.hh>
```

Public Member Functions

- bool [operator\(\)](#) (const double &lhs, const double &rhs) const

15.5.1 Detailed Description

Definition at line 31 of file PairwiseLineMatching.hh.

15.5.2 Member Function Documentation

15.5.2.1 operator()

```
bool CompareS::operator() (
    const double & lhs,
    const double & rhs ) const [inline]
```

Definition at line 32 of file PairwiseLineMatching.hh.

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

- [linenav/PairwiseLineMatching.hh](#)

15.6 depthimagescanner::DepthImageScanner Class Reference

```
#include <DepthImageScanner.h>
```

Public Member Functions

- [DepthImageScanner](#) (cv::Mat &Kc)
- [~DepthImageScanner](#) ()
- void [setdepthasUC](#) ()
- void [readlasersonar](#) (qi::AnyObject mememoryproxy)
- void [setlmname](#) (int ct)
- void [setdepthasD](#) ()
- void [convert_msg](#) (cv::Mat &, cv::Mat &grid, std::vector< double > &od, std::vector< double > &wr, std::vector< double > &wrl)
- void [set_range_limits](#) (const float range_min, const float range_max)
- void [set_scan_height](#) (const int scan_height)
- void [set_output_frame](#) (const std::string output_frame_id)

15.6.1 Detailed Description

Definition at line 53 of file DepthImageScanner.h.

15.6.2 Constructor & Destructor Documentation

15.6.2.1 DepthImageScanner()

```
DepthImageScanner::DepthImageScanner (
    cv::Mat & Kc )
```

constructor @params Kc instrinsic matrix of the depth camera

Definition at line 44 of file DepthImageScanner.cpp.

15.6.2.2 ~DepthImageScanner()

```
DepthImageScanner::~DepthImageScanner ( )
```

destructor

Definition at line 89 of file DepthImageScanner.cpp.

15.6.3 Member Function Documentation

15.6.3.1 convert_msg()

```
void DepthImageScanner::convert_msg (
    cv::Mat & depth,
    cv::Mat & grid,
    std::vector< double > & od,
    std::vector< double > & wr,
    std::vector< double > & wrl )
```

Converts depth image to 2D occupancy grid.

Parameters

<i>depth</i>	depth image.
<i>grid</i>	output grid image
<i>od</i>	input robot's transformation from previous frame measured fro robot's odometry
<i>wr,wrl</i>	=> rotational velocity vector.

Definition at line 146 of file DepthImageScanner.cpp.

15.6.3.2 readlasersonar()

```
void DepthImageScanner::readlasersonar (
    qi::AnyObject memeoryproxy )
```

reads laser and sonar value

Parameters

<i>memeoryproxy</i>	memeory proxy object
---------------------	----------------------

Definition at line 141 of file DepthImageScanner.cpp.

15.6.3.3 set_output_frame()

```
void DepthImageScanner::set_output_frame (
    const std::string output_frame_id )
```

Sets the frame_id for the output LaserScan.

Output frame_id for the LaserScan. Will probably NOT be the same frame_id as the depth image. Example: For OpenNI cameras, this should be set to 'camera_depth_frame' while the camera uses 'camera_depth_optical_frame'.

Parameters

<i>output_frame_id</i>	Frame_id to use for the output sensor_msgs::LaserScan.
------------------------	--

Definition at line 218 of file DepthImageScanner.cpp.

15.6.3.4 set_range_limits()

```
void DepthImageScanner::set_range_limits (
    const float range_min,
    const float range_max )
```

Sets the minimum and maximum range for laser scan

`range_min` is used to determine how close of a value to allow through when multiple radii correspond to the same angular increment. `range_max` is used to set the output message.

Parameters

<code>range_min</code>	Minimum range to assign points to the laserscan, also minimum range to use points in the output scan.
<code>range_max</code>	Maximum range to use points in the output scan.

Definition at line 209 of file `DepthImageScanner.cpp`.

15.6.3.5 `set_scan_height()`

```
void DepthImageScanner::set_scan_height (
    const int scan_height )
```

Sets the number of image rows to use in the output `LaserScan`.

`scan_height` is the number of rows (pixels) to use in the output. This will provide `scan_height` number of radii for each angular increment. The output scan will output the closest radius that is still not smaller than `range_min`. This function can be used to vertically compress obstacles into a single `LaserScan`.

Parameters

<code>scan_height</code>	Number of pixels centered around the center of the image to compress into the <code>LaserScan</code> .
--------------------------	--

Definition at line 214 of file `DepthImageScanner.cpp`.

15.6.3.6 `setdepthasD()`

```
void DepthImageScanner::setdepthasD ( )
```

sets depth as double

Definition at line 82 of file `DepthImageScanner.cpp`.

15.6.3.7 `setdepthasUC()`

```
void DepthImageScanner::setdepthasUC ( )
```

sets depth as unsigned char

Definition at line 75 of file `DepthImageScanner.cpp`.

15.6.3.8 `setImname()`

```
void DepthImageScanner::setImname (
    int ct )
```

set image name for saving

Parameters

<i>ct</i>	: counter for file name
-----------	-------------------------

Definition at line 772 of file DepthImageScanner.cpp.

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

- depthnav/[DepthImageScanner.h](#)
- depthnav/[DepthImageScanner.cpp](#)

15.7 depthimagescanner::DepthTraits< T > Struct Template Reference

```
#include <depth_traits.h>
```

15.7.1 Detailed Description

```
template<typename T>  
struct depthimagescanner::DepthTraits< T >
```

Definition at line 43 of file depth_traits.h.

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

- depthnav/[depth_traits.h](#)

15.8 depthimagescanner::DepthTraits< float > Struct Reference

```
#include <depth_traits.h>
```

Static Public Member Functions

- static bool [valid](#) (float depth)
- static float [toMeters](#) (float depth)
- static float [fromMeters](#) (float depth)
- static void [initializeBuffer](#) (std::vector< uint8_t > &buffer)

15.8.1 Detailed Description

Definition at line 55 of file depth_traits.h.

15.8.2 Member Function Documentation

15.8.2.1 fromMeters()

```
static float depthimagescanner::DepthTraits< float >::fromMeters (
    float depth ) [inline], [static]
```

Definition at line 59 of file depth_traits.h.

15.8.2.2 initializeBuffer()

```
static void depthimagescanner::DepthTraits< float >::initializeBuffer (
    std::vector< uint8_t > & buffer ) [inline], [static]
```

Definition at line 61 of file depth_traits.h.

15.8.2.3 toMeters()

```
static float depthimagescanner::DepthTraits< float >::toMeters (
    float depth ) [inline], [static]
```

Definition at line 58 of file depth_traits.h.

15.8.2.4 valid()

```
static bool depthimagescanner::DepthTraits< float >::valid (
    float depth ) [inline], [static]
```

Definition at line 57 of file depth_traits.h.

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

- depthnav/[depth_traits.h](#)

15.9 depthimagescanner::DepthTraits< uint16_t > Struct Reference

```
#include <depth_traits.h>
```

Static Public Member Functions

- static bool [valid](#) (uint16_t depth)
- static float [toMeters](#) (uint16_t depth)
- static uint16_t [fromMeters](#) (float depth)
- static void [initializeBuffer](#) (std::vector< uint8_t > &buffer)

15.9.1 Detailed Description

Definition at line 46 of file `depth_traits.h`.

15.9.2 Member Function Documentation

15.9.2.1 fromMeters()

```
static uint16_t depthimagescanner::DepthTraits< uint16_t >::fromMeters (
    float depth )    [inline], [static]
```

Definition at line 50 of file `depth_traits.h`.

15.9.2.2 initializeBuffer()

```
static void depthimagescanner::DepthTraits< uint16_t >::initializeBuffer (
    std::vector< uint8_t > & buffer )    [inline], [static]
```

Definition at line 51 of file `depth_traits.h`.

15.9.2.3 toMeters()

```
static float depthimagescanner::DepthTraits< uint16_t >::toMeters (
    uint16_t depth )    [inline], [static]
```

Definition at line 49 of file `depth_traits.h`.

15.9.2.4 valid()

```
static bool depthimagescanner::DepthTraits< uint16_t >::valid (
    uint16_t depth )    [inline], [static]
```

Definition at line 48 of file `depth_traits.h`.

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

- [depthnav/depth_traits.h](#)

15.10 dispNav Class Reference

The `dispNav` class.

```
#include <dispnav.h>
```

Public Member Functions

- `dispNav ()`
dispNav
- void `setDisptime` (int delay)
setDisptime delay foe dispaly window
- void `displayimage` (bool flag)
displayimage show navigation images
- void `saveimage` (bool flag)
saveimage show navigation images
- void `showfeat` (bool flag)
showfeat draw features used for ibvs in displayed image if image is displayed
- void `closedisp` ()
closedisp
- void `setKeyImages` (std::string pkim, std::string nkim, std::string nnkim)
setKeyImages set current reference images
- void `dispNavigation` (cv::Mat &lc)
dispNavigation
- void `dispNavigation` (cv::Mat &lc, `ScaleLines` &cLines, `ScaleLines` &nLines, `ScaleLines` &nnLines, std::vector< std::vector< int > > &op)
dispNavigation
- void `displImages` (std::string title, cv::Mat &IP, cv::Mat &IC, cv::Mat &IN, cv::Mat &INN)
displImages display images in 2x2 grid window
- void `displImages` (std::string title, cv::Mat &IC, cv::Mat &IK)
displImages dispaly two images side by side in window
- void `displImages` (std::string title, cv::Mat &IC, `ScaleLines` linesInLeft, cv::Mat &IK, `ScaleLines` linesInRight, std::vector< unsigned int > matchResult)
displImages dispaly two images side by side in window with matched line segemnts
- void `setpseudocolour` (int maxlines)
setpseudocolour gebnerate color for lines segents dispaly
- void `displImages` (std::string title, cv::Mat &IP, cv::Mat &IC, cv::Mat &IN, cv::Mat &INN, `ScaleLines` &cLines, `ScaleLines` &nLines, `ScaleLines` &nnLines, std::vector< std::vector< int > > &op)
displImages display images in 2x2 grid window

Public Attributes

- int `size`
- int `sz`

15.10.1 Detailed Description

The `dispNav` class.

Definition at line 34 of file `dispnav.h`.

15.10.2 Constructor & Destructor Documentation

15.10.2.1 dispNav()

```
dispNav::dispNav ( )
```

[dispNav](#)

Definition at line 26 of file dispnav.cpp.

15.10.3 Member Function Documentation

15.10.3.1 closedisp()

```
void dispNav::closedisp ( )
```

[closedisp](#)

Definition at line 401 of file dispnav.cpp.

15.10.3.2 dispImages() [1/4]

```
void dispNav::dispImages (
    std::string title,
    cv::Mat & IC,
    cv::Mat & IK )
```

dispImages display two images side by side in window

Parameters

<i>title</i>	title of window
<i>IC</i>	cv::Mat image
<i>IK</i>	cv::Mat image

Definition at line 256 of file dispnav.cpp.

15.10.3.3 dispImages() [2/4]

```
void dispNav::dispImages (
    std::string title,
```

```

cv::Mat & IC,
ScaleLines linesInLeft,
cv::Mat & IK,
ScaleLines linesInRight,
std::vector< unsigned int > matchResult )

```

displImages display two images side by side in window with matched line segments

Parameters

<i>title</i>	title of window
<i>IC</i>	cv::Mat image
<i>linesInLeft</i>	lines in IC
<i>IK</i>	cv::Mat image
<i>linesInRight</i>	lines in IK
<i>matchResult</i>	matched line indices

Definition at line 296 of file dispnav.cpp.

15.10.3.4 displImages() [3/4]

```

void dispNav::dispImages (
    std::string title,
    cv::Mat & IP,
    cv::Mat & IC,
    cv::Mat & IN,
    cv::Mat & INN )

```

displImages display images in 2x2 grid window

Parameters

<i>title</i>	title of window
<i>IP</i>	cv::Mat image
<i>IC</i>	cv::Mat image
<i>IN</i>	cv::Mat image
<i>INN</i>	cv::Mat image

Definition at line 86 of file dispnav.cpp.

15.10.3.5 displImages() [4/4]

```

void dispNav::dispImages (
    std::string title,
    cv::Mat & IP,
    cv::Mat & IC,
    cv::Mat & IN,
    cv::Mat & INN,
    ScaleLines & cLines,

```

```
ScaleLines & nLines,  
ScaleLines & nnLines,  
std::vector< std::vector< int > > & op )
```

dispImages display images in 2x2 grid window

Parameters

<i>title</i>	title title of window
<i>IP</i>	cv::Mat image
<i>IC</i>	cv::Mat image
<i>IN</i>	cv::Mat image
<i>INN</i>	cv::Mat image
<i>cLines</i>	lines in IC
<i>nLines</i>	lines in IN
<i>nnLines</i>	lines in INN
<i>op</i>	matcjed line indices among IC, IN and INN

Definition at line 153 of file dispnav.cpp.

15.10.3.6 displayimage()

```
void dispNav::displayimage (  
    bool flag )
```

displayimage show navigation images

Parameters

<i>flag</i>	0 don't show => must be set o when run inside the Pepper robot
-------------	--

Definition at line 49 of file dispnav.cpp.

15.10.3.7 dispNavigation() [1/2]

```
void dispNav::dispNavigation (  
    cv::Mat & Ic )
```

dispNavigation

Parameters

<i>Ic</i>	Current Image
-----------	---------------

Definition at line 80 of file dispnav.cpp.

15.10.3.8 dispNavigation() [2/2]

```
void dispNav::dispNavigation (
    cv::Mat & Ic,
    ScaleLines & cLines,
    ScaleLines & nLines,
    ScaleLines & nnLines,
    std::vector< std::vector< int > > & op )
```

dispNavigation

Parameters

<i>Ic</i>	Current Image
<i>cLines</i>	lines in current image
<i>nLines</i>	lines in next reference image
<i>nnLines</i>	lines in second next reference image
<i>op</i>	matchedlines index

Definition at line 68 of file dispnav.cpp.

15.10.3.9 saveimage()

```
void dispNav::saveimage (
    bool flag )
```

saveimage show navigation images

Parameters

<i>flag</i>	0 don't save
-------------	--------------

Definition at line 54 of file dispnav.cpp.

15.10.3.10 setDisptime()

```
void dispNav::setDisptime (
    int delay )
```

setDisptime delay foe dispaly window

Parameters

<i>delay</i>	time in ms
--------------	------------

Definition at line 39 of file dispnav.cpp.

15.10.3.11 setKeyImages()

```
void dispNav::setKeyImages (
    std::string pkim,
    std::string nkim,
    std::string nnkim )
```

setKeyImages set current reference images

Parameters

<i>pkim</i>	previous reference image
<i>nkim</i>	next reference image
<i>nnkim</i>	second next reference image

Definition at line 58 of file dispnav.cpp.

15.10.3.12 setpseudocolour()

```
void dispNav::setpseudocolour (
    int maxlines )
```

setpseudocolour generate color for lines segments display

Parameters

<i>maxlines</i>	maximum number of lines
-----------------	-------------------------

Definition at line 362 of file dispnav.cpp.

15.10.3.13 showfeat()

```
void dispNav::showfeat (
    bool flag )
```

showfeat draw features used for ibvs in displayed image if image is displayed

Parameters

<i>flag</i>	0 don't draw
-------------	--------------

Definition at line 45 of file dispnav.cpp.

15.10.4 Member Data Documentation

15.10.4.1 size

```
int dispNav::size
```

Definition at line 64 of file dispnav.h.

15.10.4.2 sz

```
int dispNav::sz
```

Definition at line 64 of file dispnav.h.

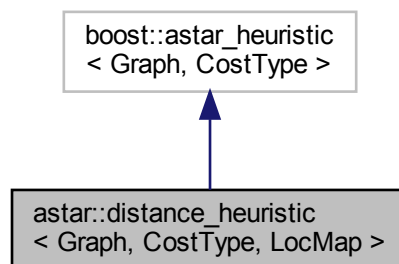
The documentation for this class was generated from the following files:

- [linenav/dispnav.h](#)
- [linenav/dispnav.cpp](#)

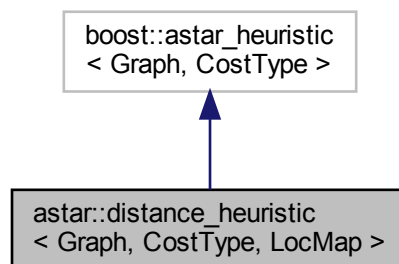
15.11 astar::distance_heuristic< Graph, CostType, LocMap > Class Template Reference

```
#include <topograph_astar.h>
```

Inheritance diagram for astar::distance_heuristic< Graph, CostType, LocMap >:



Collaboration diagram for astar::distance_heuristic< Graph, CostType, LocMap >:



Public Types

- typedef boost::graph_traits< Graph >::vertex_descriptor [Vertex](#)

Public Member Functions

- [distance_heuristic](#) (LocMap l, [Vertex](#) goal)
- CostType [operator\(\)](#) ([Vertex](#) u)

15.11.1 Detailed Description

```
template<class Graph, class CostType, class LocMap>
class astar::distance_heuristic< Graph, CostType, LocMap >
```

Definition at line 88 of file topograph_astar.h.

15.11.2 Member Typedef Documentation

15.11.2.1 Vertex

```
template<class Graph , class CostType , class LocMap >
typedef boost::graph_traits<Graph>::vertex_descriptor astar::distance\_heuristic< Graph, CostType,
LocMap >::Vertex
```

Definition at line 91 of file topograph_astar.h.

15.11.3 Constructor & Destructor Documentation

15.11.3.1 distance_heuristic()

```
template<class Graph , class CostType , class LocMap >
astar::distance\_heuristic< Graph, CostType, LocMap >::distance_heuristic (
    LocMap l,
    Vertex goal ) [inline]
```

Definition at line 92 of file topograph_astar.h.

15.11.4 Member Function Documentation

15.11.4.1 operator()

```
template<class Graph , class CostType , class LocMap >
CostType astar::distance_heuristic< Graph, CostType, LocMap >::operator() (
    Vertex u ) [inline]
```

Definition at line 94 of file topograph_astar.h.

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

- navmain/topograph_astar.h

15.12 tgraph::edge Struct Reference

```
#include <topograph_processor.h>
```

Public Attributes

- int [sid](#)
- int [eid](#)
- std::string [fold](#)

15.12.1 Detailed Description

Edges of topological graph sid: start id eid: end id fold : folder where the reference images of the edge lies edges and branches both represent the edge branches are used to simplify imgs retrival for ibvs

Definition at line 51 of file topograph_processor.h.

15.12.2 Member Data Documentation

15.12.2.1 eid

```
int tgraph::edge::eid
```

Definition at line 53 of file topograph_processor.h.

15.12.2.2 fold

```
std::string tgraph::edge::fold
```

Definition at line 54 of file topograph_processor.h.

15.12.2.3 sid

```
int tgraph::edge::sid
```

Definition at line 52 of file topograph_processor.h.

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

- navmain/[topograph_processor.h](#)

15.13 EdgeChains Struct Reference

```
#include <EDLineDetector.hh>
```

Public Attributes

- std::vector< unsigned int > [xCors](#)
- std::vector< unsigned int > [yCors](#)
- std::vector< unsigned int > [sId](#)
- unsigned int [numOfEdges](#)

15.13.1 Detailed Description

Definition at line 58 of file EDLineDetector.hh.

15.13.2 Member Data Documentation

15.13.2.1 numOfEdges

```
unsigned int EdgeChains::numOfEdges
```

Definition at line 62 of file EDLineDetector.hh.

15.13.2.2 sId

```
std::vector<unsigned int> EdgeChains::sId
```

Definition at line 61 of file EDLineDetector.hh.

15.13.2.3 xCors

```
std::vector<unsigned int> EdgeChains::xCors
```

Definition at line 59 of file EDLineDetector.hh.

15.13.2.4 yCors

```
std::vector<unsigned int> EdgeChains::yCors
```

Definition at line 60 of file EDLineDetector.hh.

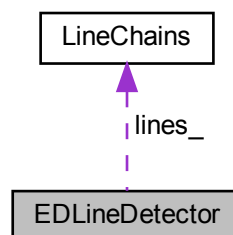
The documentation for this struct was generated from the following file:

- [linenav/EDLineDetector.hh](#)

15.14 EDLineDetector Class Reference

```
#include <EDLineDetector.hh>
```

Collaboration diagram for EDLineDetector:



Public Member Functions

- [EDLineDetector](#) ()
- [EDLineDetector](#) ([EDLineParam](#) param)
- [~EDLineDetector](#) ()
- int [EdgeDrawing](#) (cv::Mat &image, [EdgeChains](#) &edgeChains, bool smoothed=false)
- int [EDline](#) (cv::Mat &image, [LineChains](#) &lines, bool smoothed=false)
- int [EDline](#) (cv::Mat &image, bool smoothed=false)

Public Attributes

- cv::Mat [dxImg_](#)
- cv::Mat [dyImg_](#)
- cv::Mat [gImgWO_](#)
- [LineChains](#) [lines_](#)
- std::vector< std::array< double, 3 > > [lineEquations_](#)
- std::vector< std::array< float, 4 > > [lineEndpoints_](#)
- std::vector< float > [lineDirection_](#)
- std::vector< float > [lineSalience_](#)
- unsigned int [imageWidth](#)
- unsigned int [imageHeight](#)

15.14.1 Detailed Description

Definition at line 96 of file EDLineDetector.hh.

15.14.2 Constructor & Destructor Documentation

15.14.2.1 EDLineDetector() [1/2]

```
EDLineDetector::EDLineDetector ( )
```

Definition at line 60 of file EDLineDetector.cpp.

15.14.2.2 EDLineDetector() [2/2]

```
EDLineDetector::EDLineDetector (
    EDLineParam param )
```

Definition at line 73 of file EDLineDetector.cpp.

15.14.2.3 ~EDLineDetector()

```
EDLineDetector::~EDLineDetector ( )
```

Definition at line 109 of file EDLineDetector.cpp.

15.14.3 Member Function Documentation

15.14.3.1 EdgeDrawing()

```
int EDLineDetector::EdgeDrawing (
    cv::Mat & image,
    EdgeChains & edgeChains,
    bool smoothed = false )
```

Definition at line 137 of file EDLineDetector.cpp.

15.14.3.2 EDline() [1/2]

```
int EDLineDetector::EDline (
    cv::Mat & image,
    bool smoothed = false )
```

Definition at line 1273 of file EDLineDetector.cpp.

15.14.3.3 EDline() [2/2]

```
int EDLineDetector::EDline (
    cv::Mat & image,
    LineChains & lines,
    bool smoothed = false )
```

Definition at line 811 of file EDLineDetector.cpp.

15.14.4 Member Data Documentation

15.14.4.1 dxImg_

```
cv::Mat EDLineDetector::dxImg_
```

Definition at line 120 of file EDLineDetector.hh.

15.14.4.2 dyImg_

```
cv::Mat EDLineDetector::dyImg_
```

Definition at line 121 of file EDLineDetector.hh.

15.14.4.3 glmgWO_

```
cv::Mat EDLineDetector::glmgWO_
```

Definition at line 122 of file EDLineDetector.hh.

15.14.4.4 imageHeight

```
unsigned int EDLineDetector::imageHeight
```

Definition at line 133 of file EDLineDetector.hh.

15.14.4.5 imageWidth

```
unsigned int EDLineDetector::imageWidth
```

Definition at line 132 of file EDLineDetector.hh.

15.14.4.6 lineDirection_

```
std::vector<float> EDLineDetector::lineDirection_
```

Definition at line 129 of file EDLineDetector.hh.

15.14.4.7 lineEndpoints_

```
std::vector<std::array<float, 4> > EDLineDetector::lineEndpoints_
```

Definition at line 127 of file EDLineDetector.hh.

15.14.4.8 lineEquations_

```
std::vector<std::array<double, 3> > EDLineDetector::lineEquations_
```

Definition at line 125 of file EDLineDetector.hh.

15.14.4.9 lines_

[LineChains](#) EDLineDetector::lines_

Definition at line 123 of file EDLineDetector.hh.

15.14.4.10 lineSaliency_

std::vector<float> EDLineDetector::lineSaliency_

Definition at line 131 of file EDLineDetector.hh.

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

- [linenav/EDLineDetector.hh](#)
- [linenav/EDLineDetector.cpp](#)

15.15 EDLineParam Struct Reference

```
#include <EDLineDetector.hh>
```

Public Attributes

- int [ksize](#)
- float [sigma](#)
- float [gradientThreshold](#)
- float [anchorThreshold](#)
- int [scanIntervals](#)
- int [minLineLen](#)
- double [lineFitErrThreshold](#)

15.15.1 Detailed Description

Definition at line 74 of file EDLineDetector.hh.

15.15.2 Member Data Documentation

15.15.2.1 anchorThreshold

float EDLineParam::anchorThreshold

Definition at line 78 of file EDLineDetector.hh.

15.15.2.2 gradientThreshold

```
float EDLineParam::gradientThreshold
```

Definition at line 77 of file EDLineDetector.hh.

15.15.2.3 ksize

```
int EDLineParam::ksize
```

Definition at line 75 of file EDLineDetector.hh.

15.15.2.4 lineFitErrThreshold

```
double EDLineParam::lineFitErrThreshold
```

Definition at line 81 of file EDLineDetector.hh.

15.15.2.5 minLineLen

```
int EDLineParam::minLineLen
```

Definition at line 80 of file EDLineDetector.hh.

15.15.2.6 scanIntervals

```
int EDLineParam::scanIntervals
```

Definition at line 79 of file EDLineDetector.hh.

15.15.2.7 sigma

```
float EDLineParam::sigma
```

Definition at line 76 of file EDLineDetector.hh.

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

- linenav/[EDLineDetector.hh](#)

15.16 astar::found_goal Struct Reference

```
#include <topograph_astar.h>
```

15.16.1 Detailed Description

Definition at line 127 of file topograph_astar.h.

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

- [navmain/topograph_astar.h](#)

15.17 freespacenavigation Class Reference

```
#include <freespacenavigation.h>
```

Public Member Functions

- [freespacenavigation](#) (cv::Mat &kd)
freespacenavigation constructor
- [freespacenavigation](#) (cv::Mat &kd, int scan_height)
freespacenavigation constructor
- void [setinitialpose](#) (const std::vector< float > &pose)
setinitialpose initial odometry reading
- void [setcurrentpose](#) (const std::vector< float > &pose)
setcurrentpose set current odometry reading
- double [getvelfreespace](#) (cv::Mat &Id)
getvelfreespace returns velocity for freespace navigation
- double [getH](#) ()
getH returns weights required for fusion of control
- void [setbasevel](#) (double vt)
setbasevel set base forward velocity. Required for ibvs

15.17.1 Detailed Description

Definition at line 32 of file freespacenavigation.h.

15.17.2 Constructor & Destructor Documentation

15.17.2.1 freespacenavigation() [1/2]

```
freespacenavigation::freespacenavigation (
    cv::Mat & kd )
```

freespacenavigation constructor

Parameters

<i>kd</i>	: intrinsic parameters of the depth camera
-----------	--

Definition at line 28 of file freespacenavigation.cpp.

15.17.2.2 freespacenavigation() [2/2]

```
freespacenavigation::freespacenavigation (
    cv::Mat & kd,
    int scan_height )
```

freespacenavigation constructor

Parameters

<i>kd</i>	: intrinsic parameters of the depth camera
<i>scan_height</i>	: no of rows from center considered from the depth image for 2d grid map

Definition at line 48 of file freespacenavigation.cpp.

15.17.3 Member Function Documentation

15.17.3.1 getH()

```
double freespacenavigation::getH ( )
```

getH returns weights required for fusion of control

Returns

H

Definition at line 136 of file freespacenavigation.cpp.

15.17.3.2 getvelfreespace()

```
double freespacenavigation::getvelfreespace (
    cv::Mat & Id )
```

getvelfreespace returns velocity for freespace navigation

Parameters

<i>ld</i>	depth image
-----------	-------------

Returns

rotational velocity

Definition at line 101 of file freespacenavigation.cpp.

15.17.3.3 setbasevel()

```
void freespacenavigation::setbasevel (
    double vt = 0.18 )
```

setbasevel set base forward velocity. Required for ibvs

Parameters

<i>vt</i>	forward velocity
-----------	------------------

Definition at line 70 of file freespacenavigation.cpp.

15.17.3.4 setcurrentpose()

```
void freespacenavigation::setcurrentpose (
    const std::vector< float > & pose )
```

setcurrentpose set current odometry reading

Parameters

<i>pose</i>	vector consists of odometry value x,y, and theta
-------------	--

Definition at line 81 of file freespacenavigation.cpp.

15.17.3.5 setinitialpose()

```
void freespacenavigation::setinitialpose (
    const std::vector< float > & pose )
```

setinitialpose initial odometry reading

Parameters

<i>pose</i>	vector consists of odometry value x,y, and theta
-------------	--

Definition at line 74 of file freespacenavigation.cpp.

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

- [depthnav/freespacenavigation.h](#)
- [depthnav/freespacenavigation.cpp](#)

15.18 astar::graph_writer< Name, LocMap > Class Template Reference

```
#include <topograph_astar.h>
```

Public Member Functions

- [graph_writer](#) (Name n, LocMap l, float _minx, float _maxx, float _miny, float _maxy, unsigned int _ptx, unsigned int _pty)
- `template<class Vertex >`
void [operator\(\)](#) (std::ostream &out, const Vertex &v) const

15.18.1 Detailed Description

```
template<class Name, class LocMap>
class astar::graph_writer< Name, LocMap >
```

Definition at line 53 of file topograph_astar.h.

15.18.2 Constructor & Destructor Documentation

15.18.2.1 graph_writer()

```
template<class Name , class LocMap >
astar::graph_writer< Name, LocMap >::graph_writer (
    Name n,
    LocMap l,
    float _minx,
    float _maxx,
    float _miny,
    float _maxy,
    unsigned int _ptx,
    unsigned int _pty ) [inline]
```

Definition at line 55 of file topograph_astar.h.

15.18.3 Member Function Documentation

15.18.3.1 operator()

```
template<class Name , class LocMap >
template<class Vertex >
void astar::graph_writer< Name, LocMap >::operator() (
    std::ostream & out,
    const Vertex & v ) const [inline]
```

Definition at line 59 of file topograph_astar.h.

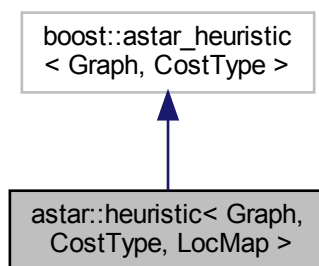
The documentation for this class was generated from the following file:

- navmain/[topograph_astar.h](#)

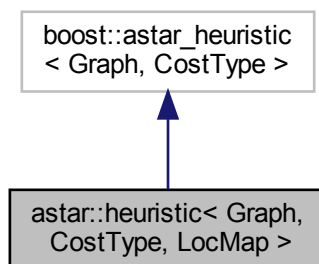
15.19 astar::heuristic< Graph, CostType, LocMap > Class Template Reference

```
#include <topograph_astar.h>
```

Inheritance diagram for astar::heuristic< Graph, CostType, LocMap >:



Collaboration diagram for astar::heuristic< Graph, CostType, LocMap >:



Public Types

- typedef boost::graph_traits< Graph >::vertex_descriptor [Vertex](#)

Public Member Functions

- [heuristic](#) (LocMap l, [Vertex](#) goal)
- CostType [operator\(\)](#) ([Vertex](#) u)

15.19.1 Detailed Description

```
template<class Graph, class CostType, class LocMap>
class astar::heuristic< Graph, CostType, LocMap >
```

Definition at line 107 of file topograph_astar.h.

15.19.2 Member Typedef Documentation

15.19.2.1 Vertex

```
template<class Graph , class CostType , class LocMap >
typedef boost::graph_traits<Graph>::vertex_descriptor astar::heuristic< Graph, CostType, LocMap >↔
::Vertex
```

Definition at line 110 of file topograph_astar.h.

15.19.3 Constructor & Destructor Documentation

15.19.3.1 heuristic()

```
template<class Graph , class CostType , class LocMap >
astar::heuristic< Graph, CostType, LocMap >::heuristic (
    LocMap l,
    Vertex goal ) [inline]
```

Definition at line 111 of file topograph_astar.h.

15.19.4 Member Function Documentation

15.19.4.1 operator()

```
template<class Graph , class CostType , class LocMap >
CostType astar::heuristic< Graph, CostType, LocMap >::operator() (
    Vertex u ) [inline]
```

Definition at line 113 of file `topograph_astar.h`.

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

- [navmain/topograph_astar.h](#)

15.20 kimRead Class Reference

```
#include <kimread.h>
```

Public Member Functions

- [kimRead](#) ()
kimRead
- [kimRead](#) (char *kimf)
kimRead
- [kimRead](#) (std::string &kimf)
kimRead Read reference images
- [kimRead](#) (std::string &kimf, int loc)
kimRead Read reference images
- [kimRead](#) (std::string &kimf, int sloc, int eloc)
kimRead Read reference images
- void [setnodes](#) (int sn, int en)
setnodes
- void [setKimbasefold](#) (std::string &fold)
setKimbasefold
- void [resetIndex](#) ()
resetIndex
- void [setKeyImageIndex](#) (int n)
setKeyImageIndex
- int [getStartIndex](#) ()
getStartIndex
- void [setCurrIndex](#) (int indx)
setCurrIndex
- void [getNextKeyImage](#) (std::string &kim)
getNextKeyImage
- void [getNextKeyImage](#) (std::string &kim, std::string &kil)
getNextKeyImage
- void [resetCurrentIndex](#) ()
resetCurrentIndex
- int [currindex](#) ()
currindex
- int [kimsiz](#)e ()
*kimsiz*e
- int [isLast](#) ()

- isLast*
- void [addKeyImages](#) ([kimRead](#) *kfptr, int stindx, int endindx)
 - addKeyImages add reference images to the current list*
- void [removeBKeyImages](#) (int idx)
 - removeBKeyImages remove reference image from the back of the list upto*
- void [insetKeyImage](#) (std::string &kim, std::string &kil)
 - insetKeyImage*
- void [removeKeyImages](#) (int idx)
 - removeKeyImages*
- void [insertKeyImages](#) ([kimRead](#) *kfptr, int desloc, int indx)
 - insertKeyImages*
- void [insertKeyImages](#) ([kimRead](#) *kfptr, int indx)
 - insertKeyImages*
- void [insertKeyImages](#) ([kimRead](#) *kfptr)
 - insertKeyImages instert reference image list*
- void [getkimfold](#) (std::string &kif)
 - getkimfold get referene image folder*
- void [showkim](#) ()
 - showkim display key images*
- [~kimRead](#) ()

15.20.1 Detailed Description

Definition at line 30 of file kimread.h.

15.20.2 Constructor & Destructor Documentation

15.20.2.1 kimRead() [1/5]

```
kimRead::kimRead ( )
```

[kimRead](#)

Definition at line 23 of file kimread.cpp.

15.20.2.2 kimRead() [2/5]

```
kimRead::kimRead (
    char * kimf )
```

[kimRead](#)

Parameters

<i>kimf</i>	folder containing reference images
-------------	------------------------------------

Definition at line 40 of file kimread.cpp.

15.20.2.3 kimRead() [3/5]

```
kimRead::kimRead (
    std::string & kimf )
```

[kimRead](#) Read reference images

Parameters

<i>kimf</i>	folder containing reference images
-------------	------------------------------------

Definition at line 86 of file kimread.cpp.

15.20.2.4 kimRead() [4/5]

```
kimRead::kimRead (
    std::string & kimf,
    int loc )
```

[kimRead](#) Read reference images

Parameters

<i>kimf</i>	folder containing reference images
<i>loc</i>	start index to insert

Definition at line 132 of file kimread.cpp.

15.20.2.5 kimRead() [5/5]

```
kimRead::kimRead (
    std::string & kimf,
    int sloc,
    int eloc )
```

[kimRead](#) Read reference images

Parameters

<i>kimf</i>	folder containing reference images
<i>sloc</i>	start index
<i>eloc</i>	end index

Definition at line 156 of file kimread.cpp.

15.20.2.6 ~kimRead()

```
kimRead::~kimRead ( )
```

Definition at line 375 of file kimread.cpp.

15.20.3 Member Function Documentation

15.20.3.1 addKeyImages()

```
void kimRead::addKeyImages (
    kimRead * kfptr,
    int stindx,
    int endindx )
```

addKeyImages add reference images to the current list

Parameters

<i>kfptr</i>	pointer of the reference image list
<i>stindx</i>	start index
<i>endindx</i>	end index

Definition at line 304 of file kimread.cpp.

15.20.3.2 currindex()

```
int kimRead::currindex ( )
```

currindex

Returns

current index

Definition at line 230 of file kimread.cpp.

15.20.3.3 getkimfold()

```
void kimRead::getkimfold (
    std::string & kif )
```

getkimfold get referene image folder

Parameters

<i>kif</i>	path
------------	------

Definition at line 260 of file kimread.cpp.

15.20.3.4 getNextKeyImage() [1/2]

```
void kimRead::getNextKeyImage (
    std::string & kim )
```

getNextKeyImage

Parameters

<i>kim</i>	read next reference image name
------------	--------------------------------

Definition at line 200 of file kimread.cpp.

15.20.3.5 getNextKeyImage() [2/2]

```
void kimRead::getNextKeyImage (
    std::string & kim,
    std::string & kil )
```

getNextKeyImage

Parameters

<i>kim</i>	next reference image name
<i>kil</i>	file conatining line segment of the corresponding reference image

Definition at line 207 of file kimread.cpp.

15.20.3.6 getStartIndex()

```
int kimRead::getStartIndex ( )
```

getStartIndex

Returns

Definition at line 219 of file kimread.cpp.

15.20.3.7 insertKeyImages() [1/3]

```
void kimRead::insertKeyImages (
    kimRead * kfptr )
```

insertKeyImages instert reference image list

Parameters

<i>kfptr</i>	pointer to the reference image list that is to be inserted
--------------	--

Definition at line 350 of file kimread.cpp.

15.20.3.8 insertKeyImages() [2/3]

```
void kimRead::insertKeyImages (
    kimRead * kfptr,
    int desloc,
    int indx )
```

insertKeyImages

Parameters

<i>kfptr</i>	pointer to reference image list
<i>desloc</i>	
<i>indx</i>	

Definition at line 287 of file kimread.cpp.

15.20.3.9 insertKeyImages() [3/3]

```
void kimRead::insertKeyImages (
    kimRead * kfptr,
    int indx )
```

insertKeyImages

Parameters

<i>kfptr</i>	
<i>indx</i>	

Definition at line 325 of file kimread.cpp.

15.20.3.10 insetKeyImage()

```
void kimRead::insetKeyImage (
    std::string & kim,
    std::string & kil )
```

insetKeyImage

Parameters

<i>kim</i>	<i>r</i>
<i>kil</i>	

Definition at line 254 of file kimread.cpp.

15.20.3.11 isLast()

```
int kimRead::isLast ( )
```

isLast

Returns

true if the index is at the end of the list

Definition at line 246 of file kimread.cpp.

15.20.3.12 kimsized()

```
int kimRead::kimsized ( )
```

kimsized

Returns

Definition at line 240 of file kimread.cpp.

15.20.3.13 removeBKeyImages()

```
void kimRead::removeBKeyImages (
    int idx )
```

removeBKeyImages remove reference image from the back of the list upto

Parameters

<i>idx</i>	index
------------	-------

Definition at line 275 of file kimread.cpp.

15.20.3.14 removeKeyImages()

```
void kimRead::removeKeyImages (
    int idx )
```

removeKeyImages

Parameters

<i>idx</i>	
------------	--

Definition at line 265 of file kimread.cpp.

15.20.3.15 resetCurrentIndex()

```
void kimRead::resetCurrentIndex ( )
```

resetCurrentIndex

Definition at line 235 of file kimread.cpp.

15.20.3.16 resetIndex()

```
void kimRead::resetIndex ( )
```

resetIndex

Definition at line 185 of file kimread.cpp.

15.20.3.17 setCurrIndex()

```
void kimRead::setCurrIndex (
    int indx )
```

setCurrIndex

Parameters

<i>indx</i>	current index
-------------	---------------

Definition at line 224 of file kimread.cpp.

15.20.3.18 setKeyImageIndex()

```
void kimRead::setKeyImageIndex (
    int n )
```

setKeyImageIndex

Parameters

<i>n</i>	index
----------	-------

Definition at line 191 of file kimread.cpp.

15.20.3.19 setKimbasefold()

```
void kimRead::setKimbasefold (
    std::string & fold )
```

setKimbasefold

Parameters

<i>fold</i>	folder location
-------------	-----------------

Definition at line 35 of file kimread.cpp.

15.20.3.20 setnodes()

```
void kimRead::setnodes (
    int sn,
    int en )
```

setnodes

Parameters

<i>sn</i>	start node
<i>en</i>	end node

Definition at line 81 of file kimread.cpp.

15.20.3.21 showkim()

```
void kimRead::showkim ( )
```

showkim display key images

Definition at line 366 of file kimread.cpp.

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

- [linenav/kimread.h](#)
- [linenav/kimread.cpp](#)

15.21 LineChains Struct Reference

```
#include <EDLineDetector.hh>
```

Public Attributes

- `std::vector< unsigned int >` [xCors](#)
- `std::vector< unsigned int >` [yCors](#)
- `std::vector< unsigned int >` [sld](#)
- `unsigned int` [numOfLines](#)

15.21.1 Detailed Description

Definition at line 64 of file EDLineDetector.hh.

15.21.2 Member Data Documentation

15.21.2.1 numOfLines

```
unsigned int LineChains::numOfLines
```

Definition at line 68 of file EDLineDetector.hh.

15.21.2.2 sId

```
std::vector<unsigned int> LineChains::sId
```

Definition at line 67 of file EDLineDetector.hh.

15.21.2.3 xCors

```
std::vector<unsigned int> LineChains::xCors
```

Definition at line 65 of file EDLineDetector.hh.

15.21.2.4 yCors

```
std::vector<unsigned int> LineChains::yCors
```

Definition at line 66 of file EDLineDetector.hh.

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

- [linenav/EDLineDetector.hh](#)

15.22 LineDescriptor Class Reference

```
#include <LineDescriptor.hh>
```

Public Types

- enum { [NearestNeighbor](#) =0, [NNDR](#) =1 }

Public Member Functions

- [LineDescriptor](#) ()
- [LineDescriptor](#) (unsigned int numOfBand, unsigned int widthOfBand)
- [~LineDescriptor](#) ()
- int [GetLineDescriptor](#) (cv::Mat &image, [ScaleLines](#) &keyLines)
- int [OctaveKeyLines](#) (cv::Mat &image, [ScaleLines](#) &keyLines)
- void [findLineDesc](#) ([ScaleLines](#) &keyLines)
- void [computeBinaryLineDesc](#) ([ScaleLines](#) &keyLines)
- int [MatchLineByDescriptor](#) ([ScaleLines](#) &keyLinesLeft, [ScaleLines](#) &keyLinesRight, std::vector< short > &matchLeft, std::vector< short > &matchRight, int criteria=[NNDR](#))
- int [LineMatchingBinary](#) ([ScaleLines](#) &keyLinesLeft, [ScaleLines](#) &keyLinesRight, std::vector< unsigned int > &match← Result)

Public Attributes

- float [LowestThreshold](#)
- float [NNDThreshold](#)
- int [bDistThreshold](#)

15.22.1 Detailed Description

Definition at line 28 of file LineDescriptor.hh.

15.22.2 Member Enumeration Documentation

15.22.2.1 anonymous enum

anonymous enum

Enumerator

NearestNeighbor	
NNDR	

Definition at line 34 of file LineDescriptor.hh.

15.22.3 Constructor & Destructor Documentation

15.22.3.1 LineDescriptor() [1/2]

```
LineDescriptor::LineDescriptor ( )
```

Definition at line 23 of file LineDescriptor.cpp.

15.22.3.2 LineDescriptor() [2/2]

```
LineDescriptor::LineDescriptor (
    unsigned int numOfBand,
    unsigned int widthOfBand )
```

Definition at line 61 of file LineDescriptor.cpp.

15.22.3.3 ~LineDescriptor()

```
LineDescriptor::~~LineDescriptor ( )
```

Definition at line 98 of file LineDescriptor.cpp.

15.22.4 Member Function Documentation

15.22.4.1 computeBinaryLineDesc()

```
void LineDescriptor::computeBinaryLineDesc (
    ScaleLines & keyLines )
```

Definition at line 1287 of file LineDescriptor.cpp.

15.22.4.2 findLineDesc()

```
void LineDescriptor::findLineDesc (
    ScaleLines & keyLines )
```

Definition at line 1140 of file LineDescriptor.cpp.

15.22.4.3 GetLineDescriptor()

```
int LineDescriptor::GetLineDescriptor (
    cv::Mat & image,
    ScaleLines & keyLines )
```

Definition at line 1170 of file LineDescriptor.cpp.

15.22.4.4 LineMatchingBinary()

```
int LineDescriptor::LineMatchingBinary (
    ScaleLines & keyLinesLeft,
    ScaleLines & keyLinesRight,
    std::vector< unsigned int > & matchResult )
```

Definition at line 1331 of file LineDescriptor.cpp.

15.22.4.5 MatchLineByDescriptor()

```
int LineDescriptor::MatchLineByDescriptor (
    ScaleLines & keyLinesLeft,
    ScaleLines & keyLinesRight,
    std::vector< short > & matchLeft,
    std::vector< short > & matchRight,
    int criteria = NNDR )
```

Definition at line 1200 of file LineDescriptor.cpp.

15.22.4.6 OctaveKeyLines()

```
int LineDescriptor::OctaveKeyLines (
    cv::Mat & image,
    ScaleLines & keyLines )
```

Definition at line 484 of file LineDescriptor.cpp.

15.22.5 Member Data Documentation

15.22.5.1 bDistThreshold

```
int LineDescriptor::bDistThreshold
```

Definition at line 53 of file LineDescriptor.hh.

15.22.5.2 LowestThreshold

```
float LineDescriptor::LowestThreshold
```

Definition at line 51 of file LineDescriptor.hh.

15.22.5.3 NNDRThreshold

```
float LineDescriptor::NNDRThreshold
```

Definition at line 52 of file LineDescriptor.hh.

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

- [linenav/LineDescriptor.hh](#)
- [linenav/LineDescriptor.cpp](#)

15.23 linematch Class Reference

```
#include <linematch.h>
```

Public Member Functions

- [linematch](#) ()
linematch
- void [readlinedesc](#) (std::string keyIn, [ScaleLines](#) &linesInRight)
readlinedesc read line descriptors from file
- void [matchlines](#) ([ScaleLines](#) &linesInLeft, [ScaleLines](#) &linesInRight, std::vector< unsigned int > &matchResult)
matchlines matches lines based on LBD descriptors
- void [matchlinesbinary](#) ([ScaleLines](#) &linesInLeft, [ScaleLines](#) &linesInRight, std::vector< unsigned int > &matchResult)
matchlinesbinary matches lines based on binary descriptors
- void [findCommonIndex](#) (std::vector< unsigned int > v1, std::vector< unsigned int > v2, std::vector< std::vector< int > > &op)
findCommonIndex find common index => for 3 view line matching between im1, im2 and im3
- void [findCommonIndex2](#) (std::vector< unsigned int > v1, std::vector< unsigned int > v2, std::vector< std::vector< int > > &op)
- int [getlinedesc](#) (cv::Mat &leftImage, [ScaleLines](#) &linesInLeft)
getlinedesc calulae line descriptors

15.23.1 Detailed Description

Definition at line 28 of file linematch.h.

15.23.2 Constructor & Destructor Documentation

15.23.2.1 linematch()

```
linematch::linematch ( )
```

linematch

Definition at line 23 of file linematch.cpp.

15.23.3 Member Function Documentation

15.23.3.1 findCommonIndex()

```
void linematch::findCommonIndex (
    std::vector< unsigned int > v1,
    std::vector< unsigned int > v2,
    std::vector< std::vector< int > > & op )
```

findCommonIndex find common index => for 3 view line matching between im1, im2 and im3

Parameters

<i>v1</i>	match pair between im1 im2
<i>v2</i>	match pair between im1 and im3
<i>op</i>	matched index

Definition at line 153 of file linematch.cpp.

15.23.3.2 findCommonIndex2()

```
void linematch::findCommonIndex2 (
    std::vector< unsigned int > v1,
    std::vector< unsigned int > v2,
    std::vector< std::vector< int > > & op )
```

Definition at line 183 of file linematch.cpp.

15.23.3.3 getlinedesc()

```
int linematch::getlinedesc (
    cv::Mat & leftImage,
    ScaleLines & linesInLeft )
```

getlinedesc calulae line descriptos

Parameters

<i>leftImage</i>	image
<i>linesInLeft</i>	line descriptors

Returns

Definition at line 42 of file linematch.cpp.

15.23.3.4 matchlines()

```
void linematch::matchlines (
    ScaleLines & linesInLeft,
    ScaleLines & linesInRight,
    std::vector< unsigned int > & matchResult )
```

matchlines matches lines based on LBD descriptors

Parameters

<i>linesInLeft</i>	
<i>linesInRight</i>	
<i>matchResult</i>	index of matched pairs

Definition at line 142 of file linematch.cpp.

15.23.3.5 matchlinesbinary()

```
void linematch::matchlinesbinary (
    ScaleLines & linesInLeft,
    ScaleLines & linesInRight,
    std::vector< unsigned int > & matchResult )
```

matchlinesbinary matches lines based on binary descriptors

Parameters

<i>linesInLeft</i>	
<i>linesInRight</i>	
<i>matchResult</i>	

Definition at line 147 of file linematch.cpp.

15.23.3.6 readlinedesc()

```
void linematch::readlinedesc (
    std::string keyIn,
    ScaleLines & linesInRight )
```

readlinedesc read line descriptors from file

Parameters

<i>keyIn</i>	file containing line segemts and descriptors
<i>linesInRight</i>	line segemnts and descriptors in ScaleLine format

Definition at line 84 of file linematch.cpp.

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

- [linenav/linematch.h](#)
- [linenav/linematch.cpp](#)

15.24 linenavigation Class Reference

```
#include <linenavigation.h>
```

Public Member Functions

- [linenavigation](#) ([dispNav](#) *d, cv::Mat &Kc)
linenavigation
- [linenavigation](#) ()
- void [setDisplay](#) ([dispNav](#) *d)
setDisplay
- void [setK](#) (cv::Mat &Kc)
setK
- int [initlocalisation](#) (cv::Mat &lc, [kimRead](#) &kf)
initlocalisation perform global localiztion in the reference image list
- void [initiallocalisationLines](#) (cv::Mat &lc, [kimRead](#) &kf, int &idx, int &nml)
initlocalisation perform global localiztion in the reference image list
- void [SetKeyImages](#) (std::string pim, std::string nim, std::string nnim)
SetKeyImages set current reference images.
- void [SwitchKeyImages](#) (std::string nnim)
SwitchKeyImages Switch reference images.
- int [SwitchtoNewKeyImages](#) ([kimRead](#) &kf)
SwitchtoNewKeyImages switch reference images.
- int [step](#) (cv::Mat &lc)
step perform succesive localiztion
- int [initiallocalisationLines](#) (cv::Mat &lc, [kimRead](#) &kf)
initlocalisation perform global localiztion in the reference image list
- int [setInitialKeyImages](#) (cv::Mat &lc, [kimRead](#) &kf, int index)
setInitialKeyImages set inital set of reference images
- double [getRotVel](#) ()
getRotVel calucluate rotational velocity 3 view version
- double [getRotVel2](#) ()
getRotVel2 calucluate rotational velocity 2 view version
- double [getRotVel](#) (std::vector< double > &err, std::vector< double > &jac)
getRotVel calucluate rotational velocity 3 view version
- double [getRotVel2](#) (std::vector< double > &err, std::vector< double > &jac)
getRotVel2 calucluate rotational velocity 2 view version
- double [getdiffN2NN](#) ()
getdiffN2NN

Public Attributes

- int [linesno](#)
- int [linesnoN](#)
- int [linesnoNN](#)

Protected Attributes

- std::string [previmg](#)
- std::string [nextimg](#)
- std::string [nextnextimg](#)

15.24.1 Detailed Description

Definition at line 31 of file `linenavigation.h`.

15.24.2 Constructor & Destructor Documentation

15.24.2.1 linenavigation() [1/2]

```
linenavigation::linenavigation (
    dispNav * d,
    cv::Mat & Kc )
```

linenavigation

Parameters

<i>d</i>	pointer to display
<i>Kc</i>	intrinsic matrix of RGB camera

Definition at line 49 of file linenavigation.cpp.

15.24.2.2 linenavigation() [2/2]

```
linenavigation::linenavigation ( )
```

Definition at line 24 of file linenavigation.cpp.

15.24.3 Member Function Documentation

15.24.3.1 getdiffN2NN()

```
double linenavigation::getdiffN2NN ( )
```

getdiffN2NN

Returns

lateral displacement between reference images

Definition at line 117 of file linenavigation.cpp.

15.24.3.2 getRotVel() [1/2]

```
double linenavigation::getRotVel ( )
```

getRotVel calucluate rotational velocity 3 view version

Returns

rotational velocity

Definition at line 595 of file linenavigation.cpp.

15.24.3.3 getRotVel() [2/2]

```
double linenavigation::getRotVel (
    std::vector< double > & err,
    std::vector< double > & jac )
```

getRotVel calucluate rotational velocity 3 view version

Parameters

<i>err</i>	error
<i>jac</i>	jacobian

Returns

rotational velocity

Definition at line 787 of file linenavigation.cpp.

15.24.3.4 getRotVel2() [1/2]

```
double linenavigation::getRotVel2 ( )
```

getRotVel2 calucluate rotational velocity 2 view version

Returns

rotational velocity

Definition at line 625 of file linenavigation.cpp.

15.24.3.5 getRotVel2() [2/2]

```
double linenavigation::getRotVel2 (
    std::vector< double > & err,
    std::vector< double > & jac )
```

getRotVel2 calucluate rotational velocity 2 view version

Parameters

<i>err</i>	error
<i>jac</i>	jacobian

Returns

rotational velocity

Definition at line 828 of file linenavigation.cpp.

15.24.3.6 initiallocalisationLines() [1/2]

```
int linenavigation::initiallocalisationLines (
    cv::Mat & Ic,
    kimRead & kf )
```

initlocalisation perform global localiztion in the reference image list

Parameters

<i>Ic</i>	current image
<i>kf</i>	reference image list

Definition at line 286 of file linenavigation.cpp.

15.24.3.7 initiallocalisationLines() [2/2]

```
void linenavigation::initiallocalisationLines (
    cv::Mat & Ic,
    kimRead & kf,
    int & idx,
    int & nml )
```

initlocalisation perform global localiztion in the reference image list

Parameters

<i>Ic</i>	current image
<i>kf</i>	reference image list
<i>idx</i>	index of the refernce image in the list that matches best with current image
<i>nml</i>	maximum number of matched lines

Definition at line 356 of file linenavigation.cpp.

15.24.3.8 initlocalisation()

```
int linenavigation::initlocalisation (
    cv::Mat & Ic,
    kimRead & kf )
```

initlocalisation perform global localiztion in the reference image list

Parameters

<i>Ic</i>	current image
<i>kf</i>	reference image list

Definition at line 176 of file linenavigation.cpp.

15.24.3.9 setDisplay()

```
void linenavigation::setDisplay (
    dispNav * d )
```

setDisplay

Parameters

<i>d</i>	pointer to display
----------	--------------------

Definition at line 37 of file linenavigation.cpp.

15.24.3.10 setInitialKeyImages()

```
int linenavigation::setInitialKeyImages (
    cv::Mat & Ic,
    kimRead & kf,
    int index )
```

setInitialKeyImages set inital set of reference images

Parameters

<i>Ic</i>	curr image
<i>kf</i>	reference image list
<i>index</i>	index in the list

Returns

Definition at line 437 of file linenavigation.cpp.

15.24.3.11 setK()

```
void linenavigation::setK (
    cv::Mat & Kc )
```

setK

Parameters

<i>Kc</i>	intrinsic matrix of RGB camera
-----------	--------------------------------

Definition at line 41 of file linenavigation.cpp.

15.24.3.12 SetKeyImages()

```
void linenavigation::SetKeyImages (
    std::string pim,
    std::string nim,
    std::string nnim )
```

SetKeyImages set current reference images.

Parameters

<i>pim</i>	path of previous reference image
<i>nim</i>	path of next reference image
<i>nnim</i>	path of second-next reference image

Definition at line 64 of file linenavigation.cpp.

15.24.3.13 step()

```
int linenavigation::step (
    cv::Mat & Ic )
```

step perform successive localization

Parameters

<i>Ic</i>	current image
-----------	---------------

Returns

1 if reference images need to be switched 0 continue <0 = expectations in linedetection/ matching

Definition at line 121 of file linenavigation.cpp.

15.24.3.14 SwitchKeyImages()

```
void linenavigation::SwitchKeyImages (
    std::string nnim )
```

SwitchKeyImages Switch reference images.

Parameters

<i>nnim</i>	path of second-next reference image
-------------	-------------------------------------

Definition at line 75 of file linenavigation.cpp.

15.24.3.15 SwitchtoNewKeyImages()

```
int linenavigation::SwitchtoNewKeyImages (
    kimRead & kf )
```

SwitchtoNewKeyImages switch reference images.

Parameters

<i>kf</i>	
-----------	--

Returns

1 for end of topological navigation

Definition at line 84 of file linenavigation.cpp.

15.24.4 Member Data Documentation

15.24.4.1 linesno

```
int linenavigation::linesno
```

Definition at line 191 of file linenavigation.h.

15.24.4.2 linesnoN

```
int linenavigation::linesnoN
```

Definition at line 192 of file linenavigation.h.

15.24.4.3 linesnoNN

```
int linenavigation::linesnoNN
```

Definition at line 193 of file linenavigation.h.

15.24.4.4 nextimg

```
std::string linenavigation::nextimg [protected]
```

Definition at line 76 of file linenavigation.h.

15.24.4.5 nextnextimg

```
std::string linenavigation::nextnextimg [protected]
```

Definition at line 76 of file linenavigation.h.

15.24.4.6 previmg

```
std::string linenavigation::previmg [protected]
```

Definition at line 76 of file linenavigation.h.

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

- [linenav/linenavigation.h](#)
- [linenav/linenavigation.cpp](#)

15.25 astar::location Struct Reference

```
#include <topograph_astar.h>
```

Public Attributes

- float [y](#)
- float [x](#)

15.25.1 Detailed Description

Definition at line 46 of file `topograph_astar.h`.

15.25.2 Member Data Documentation

15.25.2.1 [x](#)

```
float astar::location::x
```

Definition at line 48 of file `topograph_astar.h`.

15.25.2.2 [y](#)

```
float astar::location::y
```

Definition at line 48 of file `topograph_astar.h`.

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

- `navmain/topograph_astar.h`

15.26 Matrix< T > Class Template Reference

```
#include <PairwiseLineMatching.hh>
```

Public Member Functions

- [Matrix](#) (unsigned int r, unsigned int c)
- T *& [operator\[\]](#) (const int &index) const
- void [SetZero](#) ()
- [~Matrix](#) ()

15.26.1 Detailed Description

```
template<class T>
class Matrix< T >
```

Definition at line 38 of file `PairwiseLineMatching.hh`.

15.26.2 Constructor & Destructor Documentation

15.26.2.1 Matrix()

```
template<class T >
Matrix< T >::Matrix (
    unsigned int r,
    unsigned int c ) [inline]
```

Definition at line 44 of file PairwiseLineMatching.hh.

15.26.2.2 ~Matrix()

```
template<class T >
Matrix< T >::~~Matrix ( ) [inline]
```

Definition at line 62 of file PairwiseLineMatching.hh.

15.26.3 Member Function Documentation

15.26.3.1 operator[]()

```
template<class T >
T* & Matrix< T >::operator[] (
    const int & index ) const [inline]
```

Definition at line 52 of file PairwiseLineMatching.hh.

15.26.3.2 SetZero()

```
template<class T >
void Matrix< T >::SetZero ( ) [inline]
```

Definition at line 56 of file PairwiseLineMatching.hh.

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

- [linenav/PairwiseLineMatching.hh](#)

15.27 MyService Class Reference

Public Member Functions

- [MyService](#) (qi::AnyObject &almemory)
- void [myCallback](#) (const std::string &key, const qi::AnyValue &value, const qi::AnyValue &message)
- void [myCallback2](#) (const std::string &key, const qi::AnyValue &value, const qi::AnyValue &message)
- void [myCallback1](#) (const std::string &key, const qi::AnyValue &value, const qi::AnyValue &message)

15.27.1 Detailed Description

Definition at line 27 of file depthnav_Pepper.cpp.

15.27.2 Constructor & Destructor Documentation

15.27.2.1 MyService()

```
MyService::MyService (  
    qi::AnyObject & almemory ) [inline]
```

Definition at line 32 of file depthnav_Pepper.cpp.

15.27.3 Member Function Documentation

15.27.3.1 myCallback()

```
void MyService::myCallback (  
    const std::string & key,  
    const qi::AnyValue & value,  
    const qi::AnyValue & message ) [inline]
```

Definition at line 49 of file depthnav_Pepper.cpp.

15.27.3.2 myCallback1()

```
void MyService::myCallback1 (  
    const std::string & key,  
    const qi::AnyValue & value,  
    const qi::AnyValue & message ) [inline]
```

Definition at line 95 of file depthnav_Pepper.cpp.

15.27.3.3 myCallback2()

```
void MyService::myCallback2 (
    const std::string & key,
    const qi::AnyValue & value,
    const qi::AnyValue & message ) [inline]
```

Definition at line 65 of file depthnav_Pepper.cpp.

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

- depthnav/[depthnav_Pepper.cpp](#)

15.28 navigation Class Reference

The navigation class.

```
#include <navigation.h>
```

Public Member Functions

- [navigation](#) ([dispNav](#) *d, cv::Mat &Kc)
navigation
- void [initlocalisation](#) (cv::Mat &lc, [kimRead](#) &kf)
initlocalisation perform global localiztion in the reference image list
- void [initlocalisation](#) (cv::Mat &lc, [kimRead](#) &kf, int &idx, int &nIm)
initlocalisation perform global localiztion in the reference image list
- int [step](#) (cv::Mat &lc)
step perform succesive localiztion
- int [SwitchtoNewKeyImages](#) ([kimRead](#) &kf)
SwitchtoNewKeyImages switch reference images.
- double [getRotVel](#) ()
getRotVel calulate rotational velocity based on ibvs
- double [getturninginkim](#) ()
getturninginkim
- double [getinitdisp](#) ()
getinitdisp getinitdisplacemt get lateral displacement with refernce images
- void [usecollisionavoidance](#) (bool flag)
usecollisionavoidance set collisionavoidance flag
- bool [usecollisionavoidance](#) ()
usecollisionavoidance

Public Attributes

- ofstream [velfile](#)
- ofstream [featfile](#)
- ofstream [jacfile](#)
- ofstream [errfile](#)

15.28.1 Detailed Description

The navigation class.

Definition at line 38 of file navigation.h.

15.28.2 Constructor & Destructor Documentation

15.28.2.1 navigation()

```
navigation::navigation (
    dispNav * d,
    cv::Mat & Kc )
```

navigation

Parameters

<i>d</i>	pointer to display
<i>Kc</i>	intrinsic matrix of RGB camera

Definition at line 25 of file navigation.cpp.

15.28.3 Member Function Documentation

15.28.3.1 getinitdisp()

```
double navigation::getinitdisp ( )
```

getinitdisp getinitdisplacemt get lateral displacement with refernce images

Returns

displacement

Definition at line 116 of file navigation.cpp.

15.28.3.2 getRotVel()

```
double navigation::getRotVel ( )
```

getRotVel calulate rotational velocity based on ibvs

Returns

rotational velocity

Definition at line 87 of file navigation.cpp.

15.28.3.3 getturninginkim()

```
double navigation::getturninginkim ( )
```

getturninginkim

Returns

turnval

Definition at line 123 of file navigation.cpp.

15.28.3.4 initlocalisation() [1/2]

```
void navigation::initlocalisation (
    cv::Mat & Ic,
    kimRead & kf )
```

initlocalisation perform global localiztion in the reference image list

Parameters

<i>Ic</i>	current image
<i>kf</i>	reference image list

Definition at line 63 of file navigation.cpp.

15.28.3.5 initlocalisation() [2/2]

```
void navigation::initlocalisation (
    cv::Mat & Ic,
    kimRead & kf,
    int & idx,
    int & nIm )
```

initlocalisation perform global localiztion in the reference image list

Parameters

<i>Ic</i>	current image
<i>kf</i>	reference image list
<i>idx</i>	index of the reference image in the list that matches best with current image
<i>nIm</i>	maximum number of matched lines

Definition at line 70 of file navigation.cpp.

15.28.3.6 step()

```
int navigation::step (
    cv::Mat & Ic )
```

step perform succesive localizion

Parameters

<i>Ic</i>	current image
-----------	---------------

Returns

1 if reference images need to be switched 0 continue <0 = expections in linedetection/ matching

Definition at line 76 of file navigation.cpp.

15.28.3.7 SwitchtoNewKeyImages()

```
int navigation::SwitchtoNewKeyImages (
    kimRead & kf )
```

SwitchtoNewKeyImages switch reference images.

Parameters

<i>kf</i>	
-----------	--

Returns

1 for end of topological navigation

Definition at line 82 of file navigation.cpp.

15.28.3.8 usecollisionavoidance() [1/2]

```
bool navigation::usecollisionavoidance ( )
usecollisionavoidance
```

Returns

0 if collision avoidance is not used 1 if used

Definition at line 59 of file navigation.cpp.

15.28.3.9 usecollisionavoidance() [2/2]

```
void navigation::usecollisionavoidance (
    bool flag )
```

usecollisionavoidance set collisionavoidance flag

Parameters

<i>flag</i>	0=> don't use , 1 =>use
-------------	-------------------------

Definition at line 55 of file navigation.cpp.

15.28.4 Member Data Documentation

15.28.4.1 errfile

```
ofstream navigation::errfile
```

Definition at line 67 of file navigation.h.

15.28.4.2 featfile

```
ofstream navigation::featfile
```

Definition at line 67 of file navigation.h.

15.28.4.3 jacfile

```
ofstream navigation::jacfile
```

Definition at line 67 of file navigation.h.

15.28.4.4 velfile

```
ofstream navigation::velfile
```

Definition at line 67 of file navigation.h.

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

- navmain/[navigation.h](#)
- navmain/[navigation.cpp](#)

15.29 NavInside Class Reference

```
#include <navinside.h>
```

Public Member Functions

- [NavInside](#) ()
[NavInside::NavInside](#).
- [~NavInside](#) ()
- void [setPointers](#) ([astar::astar](#) &as, [tgraph::topmapprocessor](#) &TP)
setPointers set pointers rekaed to graph processing and searching
- void [setPepperRobotPointer](#) ([pepperInterface](#) *pr, [dispNav](#) *d)
setPepperRobotPointer
- void [setKimfoldoffline](#) (std::string kif)
setKimfoldoffline
- void [startRobotNavigation](#) ()
- void [setGraph](#) (int N, [astar::node](#) *node_array, int num_edges, [astar::edge](#) *edge_array, [astar::cost](#) *wts)

15.29.1 Detailed Description

Definition at line 30 of file navinside.h.

15.29.2 Constructor & Destructor Documentation

15.29.2.1 NavInside()

```
NavInside::NavInside ( )
```

[NavInside::NavInside](#).

Parameters

<i>parent</i>	
---------------	--

Definition at line 29 of file navinside.cpp.

15.29.2.2 ~NavInside()

```
NavInside::~NavInside ( )
```

Definition at line 52 of file navinside.cpp.

15.29.3 Member Function Documentation

15.29.3.1 setGraph()

```
void NavInside::setGraph (
    int N,
    astar::node * node_array,
    int num_edges,
    astar::edge * edge_array,
    astar::cost * wt )
```

Definition at line 57 of file navinside.cpp.

15.29.3.2 setKimfoldoffline()

```
void NavInside::setKimfoldoffline (
    std::string kif )
```

setKimfoldoffline

Parameters

<i>kif</i>	
------------	--

Definition at line 564 of file navinside.cpp.

15.29.3.3 setPepperRobotPointer()

```
void NavInside::setPepperRobotPointer (
    pepperInterface * pr,
    dispNav * d )
```

setPepperRobotPointer

Parameters

<i>pr</i>	
-----------	--

Definition at line 190 of file navinside.cpp.

15.29.3.4 setPointers()

```
void NavInside::setPointers (
    astar::astar & as,
    tgraph::topmapprocessor & TP )
```

setPointers set pointers rekaed to graph processing and searching

Parameters

<i>as</i>	Pointer to astar search
<i>TP</i>	Pointer to topological garph processor

Definition at line 179 of file navinside.cpp.

15.29.3.5 startRobotNavigation()

```
void NavInside::startRobotNavigation ( )
```

Definition at line 232 of file navinside.cpp.

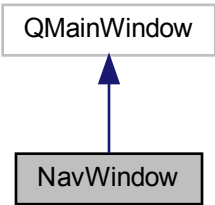
The documentation for this class was generated from the following files:

- navmain/maingui/[navinside.h](#)
- navmain/maingui/[navinside.cpp](#)

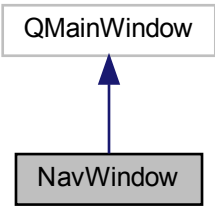
15.30 NavWindow Class Reference

```
#include <navwindow.h>
```

Inheritance diagram for NavWindow:



Collaboration diagram for NavWindow:



Public Member Functions

- [NavWindow](#) (QWidget *parent=nullptr)
NavWindow.
- [~NavWindow](#) ()
- void [drawGraph](#) (int N, [astar::node](#) *node_array, int num_edges, [astar::edge](#) *edge_array, [astar::cost](#) *wts)
drawGraph create ad draw topological graph
- void [setPointers](#) ([astar::astar](#) &as, [tgraph::topmapprocessor](#) &TP)
setPointers set pointers rekared to graph processing and searching
- void [setPepperRobotPointer](#) ([pepperInterface](#) *pr, [dispNav](#) *d)
setPepperRobotPointer
- void [setKimfoldoffline](#) (std::string kif)
setKimfoldoffline

15.30.1 Detailed Description

Definition at line 38 of file navwindow.h.

15.30.2 Constructor & Destructor Documentation

15.30.2.1 NavWindow()

```
NavWindow::NavWindow (
    QWidget * parent = nullptr ) [explicit]
```

[NavWindow.](#)

[NavWindow::NavWindow.](#)

Parameters

<i>parent</i>	
---------------	--

Definition at line 34 of file navwindow.cpp.

15.30.2.2 ~NavWindow()

```
NavWindow::~NavWindow ( )
```

Definition at line 179 of file navwindow.cpp.

15.30.3 Member Function Documentation

15.30.3.1 drawGraph()

```
void NavWindow::drawGraph (
    int N,
    astar::node * node_array,
    int num_edges,
    astar::edge * edge_array,
    astar::cost * wt )
```

drawGraph create ad draw topological graph

Parameters

<i>N</i>	numebr of nodes
<i>node_array</i>	nodes list
<i>num_edges</i>	number of edges
<i>edge_array</i>	edge list
<i>wt</i>	edge weights

Definition at line 100 of file navwindow.cpp.

15.30.3.2 setKimfoldoffline()

```
void NavWindow::setKimfoldoffline (
    std::string kif )
```

setKimfoldoffline

Parameters

<i>kif</i>	
------------	--

Definition at line 772 of file navwindow.cpp.

15.30.3.3 setPepperRobotPointer()

```
void NavWindow::setPepperRobotPointer (
    pepperInterface * pr,
    dispNav * d )
```

setPepperRobotPointer

Parameters

<i>pr</i>	
-----------	--

Definition at line 314 of file navwindow.cpp.

15.30.3.4 setPointers()

```
void NavWindow::setPointers (
    astar::astar & as,
    tgraph::topmapprocessor & TP )
```

setPointers set pointers rekaed to graph processing and searching

Parameters

<i>as</i>	Pointer to astar search
<i>TP</i>	Pointer to topological garph processor

Definition at line 303 of file navwindow.cpp.

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

- navmain/maingui/[navwindow.h](#)
- navmain/maingui/[navwindow.cpp](#)

15.31 tgraph::node Struct Reference

The node struct each node has nodeid starting from 0. nodename : higher level id for node.

```
#include <topograph_processor.h>
```

Public Attributes

- int [nodeid](#)
- std::string [nodename](#)

15.31.1 Detailed Description

The node struct each node has nodeid starting from 0. nodename : higher level id for node.

Definition at line 37 of file topograph_processor.h.

15.31.2 Member Data Documentation

15.31.2.1 nodeid

```
int tgraph::node::nodeid
```

Definition at line 38 of file topograph_processor.h.

15.31.2.2 nodename

```
std::string tgraph::node::nodename
```

Definition at line 39 of file `topograph_processor.h`.

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

- [navmain/topograph_processor.h](#)

15.32 Node Struct Reference

```
#include <PairwiseLineMatching.hh>
```

Public Attributes

- unsigned int [leftLineID](#)
- unsigned int [rightLineID](#)

15.32.1 Detailed Description

Definition at line 18 of file `PairwiseLineMatching.hh`.

15.32.2 Member Data Documentation

15.32.2.1 leftLineID

```
unsigned int Node::leftLineID
```

Definition at line 19 of file `PairwiseLineMatching.hh`.

15.32.2.2 rightLineID

```
unsigned int Node::rightLineID
```

Definition at line 20 of file `PairwiseLineMatching.hh`.

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

- [linenav/PairwiseLineMatching.hh](#)

15.33 OctaveLine Struct Reference

```
#include <LineDescriptor.hh>
```

Public Attributes

- unsigned int [octaveCount](#)
- unsigned int [lineIDInOctave](#)
- unsigned int [lineIDInScaleLineVec](#)
- float [lineLength](#)

15.33.1 Detailed Description

Definition at line 19 of file LineDescriptor.hh.

15.33.2 Member Data Documentation

15.33.2.1 lineIDInOctave

```
unsigned int OctaveLine::lineIDInOctave
```

Definition at line 21 of file LineDescriptor.hh.

15.33.2.2 lineIDInScaleLineVec

```
unsigned int OctaveLine::lineIDInScaleLineVec
```

Definition at line 22 of file LineDescriptor.hh.

15.33.2.3 lineLength

```
float OctaveLine::lineLength
```

Definition at line 23 of file LineDescriptor.hh.

15.33.2.4 octaveCount

```
unsigned int OctaveLine::octaveCount
```

Definition at line 20 of file LineDescriptor.hh.

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

- [linenav/LineDescriptor.hh](#)

15.34 OctaveSingleLine Struct Reference

```
#include <LineStructure.hh>
```

Public Attributes

- float [startPointX](#)
- float [startPointY](#)
- float [endPointX](#)
- float [endPointY](#)
- float [sPointInOctaveX](#)
- float [sPointInOctaveY](#)
- float [ePointInOctaveX](#)
- float [ePointInOctaveY](#)
- float [direction](#)
- float [salience](#)
- float [lineLength](#)
- unsigned int [numOfPixels](#)
- unsigned int [octaveCount](#)
- std::vector< float > [descriptor](#)
- std::vector< unsigned char > [bdescriptor](#)

15.34.1 Detailed Description

Definition at line 48 of file LineStructure.hh.

15.34.2 Member Data Documentation

15.34.2.1 bdescriptor

```
std::vector<unsigned char> OctaveSingleLine::bdescriptor
```

Definition at line 74 of file LineStructure.hh.

15.34.2.2 descriptor

```
std::vector<float> OctaveSingleLine::descriptor
```

Definition at line 72 of file LineStructure.hh.

15.34.2.3 direction

```
float OctaveSingleLine::direction
```

Definition at line 62 of file LineStructure.hh.

15.34.2.4 endPointX

```
float OctaveSingleLine::endPointX
```

Definition at line 54 of file LineStructure.hh.

15.34.2.5 endPointY

```
float OctaveSingleLine::endPointY
```

Definition at line 55 of file LineStructure.hh.

15.34.2.6 ePointInOctaveX

```
float OctaveSingleLine::ePointInOctaveX
```

Definition at line 59 of file LineStructure.hh.

15.34.2.7 ePointInOctaveY

```
float OctaveSingleLine::ePointInOctaveY
```

Definition at line 60 of file LineStructure.hh.

15.34.2.8 lineLength

```
float OctaveSingleLine::lineLength
```

Definition at line 66 of file LineStructure.hh.

15.34.2.9 numOfPixels

```
unsigned int OctaveSingleLine::numOfPixels
```

Definition at line 68 of file LineStructure.hh.

15.34.2.10 octaveCount

```
unsigned int OctaveSingleLine::octaveCount
```

Definition at line 70 of file LineStructure.hh.

15.34.2.11 salience

```
float OctaveSingleLine::salience
```

Definition at line 64 of file LineStructure.hh.

15.34.2.12 sPointInOctaveX

```
float OctaveSingleLine::sPointInOctaveX
```

Definition at line 57 of file LineStructure.hh.

15.34.2.13 sPointInOctaveY

```
float OctaveSingleLine::sPointInOctaveY
```

Definition at line 58 of file LineStructure.hh.

15.34.2.14 startPointX

```
float OctaveSingleLine::startPointX
```

Definition at line 52 of file LineStructure.hh.

15.34.2.15 startPointY

```
float OctaveSingleLine::startPointY
```

Definition at line 53 of file LineStructure.hh.

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

- [linenav/LineStructure.hh](#)

15.35 PairwiseLineMatching Class Reference

```
#include <PairwiseLineMatching.hh>
```

Public Member Functions

- [PairwiseLineMatching](#) ()
- void [LineMatching](#) ([ScaleLines](#) &linesInLeft, [ScaleLines](#) &linesInRight, std::vector< unsigned int > &matchResult)

15.35.1 Detailed Description

Definition at line 67 of file PairwiseLineMatching.hh.

15.35.2 Constructor & Destructor Documentation

15.35.2.1 PairwiseLineMatching()

```
PairwiseLineMatching::PairwiseLineMatching ( ) [inline]
```

Definition at line 70 of file PairwiseLineMatching.hh.

15.35.3 Member Function Documentation

15.35.3.1 LineMatching()

```
void PairwiseLineMatching::LineMatching (
    ScaleLines & linesInLeft,
    ScaleLines & linesInRight,
    std::vector< unsigned int > & matchResult )
```

Definition at line 63 of file PairwiseLineMatching.cpp.

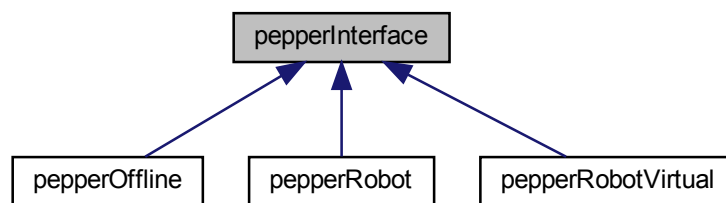
The documentation for this class was generated from the following files:

- [linenav/PairwiseLineMatching.hh](#)
- [linenav/PairwiseLineMatching.cpp](#)

15.36 pepperInterface Class Reference

```
#include <pepperInterface.h>
```

Inheritance diagram for pepperInterface:



Public Member Functions

- [pepperInterface](#) (const std::string &opt_ip)
- virtual void [openCamera](#) (int id)=0
- virtual void [startBaseMotionController](#) ()=0
- virtual void [initPosture](#) ()=0
- virtual void [startPepper](#) (int id)=0
- virtual void [getCurrImage](#) (cv::Mat &l)=0
- virtual void [openDepthCamera](#) ()=0
- virtual void [getCurrDepthImage](#) (cv::Mat &l)=0
- virtual void [setBaseVelocities](#) (float vr, float wr)=0
- virtual void [setBaseVelocities](#) (float vr, float vs, float wr)=0
- virtual std::vector< float > [getOdometryReading](#) ()=0
- virtual void [rotate180](#) ()=0
- virtual void [wait](#) (long t)=0
- virtual void [adjusthead](#) ()=0
- virtual cv::Mat [getK](#) ()=0
- virtual cv::Mat [getKd](#) ()=0
- virtual int [getid](#) ()=0
- virtual int [getmode](#) ()=0
- virtual [~pepperInterface](#) ()=0

15.36.1 Detailed Description

Definition at line 28 of file pepperInterface.h.

15.36.2 Constructor & Destructor Documentation

15.36.2.1 pepperInterface()

```
pepperInterface::pepperInterface (
    const std::string & opt_ip )
```

Definition at line 25 of file pepperInterface.cpp.

15.36.2.2 ~pepperInterface()

```
pepperInterface::~~pepperInterface ( ) [pure virtual]
```

Definition at line 32 of file pepperInterface.cpp.

15.36.3 Member Function Documentation

15.36.3.1 adjusthead()

```
virtual void pepperInterface::adjusthead ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.2 getCurrDepthImage()

```
virtual void pepperInterface::getCurrDepthImage (
    cv::Mat & I ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.3 getCurImage()

```
virtual void pepperInterface::getCurImage (
    cv::Mat & I ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.4 getId()

```
virtual int pepperInterface::getId ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.5 getK()

```
virtual cv::Mat pepperInterface::getK ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.6 getKd()

```
virtual cv::Mat pepperInterface::getKd ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.7 getmode()

```
virtual int pepperInterface::getmode ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.8 getOdometryReading()

```
virtual std::vector<float> pepperInterface::getOdometryReading ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.9 initPosture()

```
virtual void pepperInterface::initPosture ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.10 openCamera()

```
virtual void pepperInterface::openCamera (
    int id ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.11 openDepthCamera()

```
virtual void pepperInterface::openDepthCamera ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.12 rotate180()

```
virtual void pepperInterface::rotate180 ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.13 setBaseVelocities() [1/2]

```
virtual void pepperInterface::setBaseVelocities (
    float vr,
    float vs,
    float wr ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.14 setBaseVelocities() [2/2]

```
virtual void pepperInterface::setBaseVelocities (
    float vr,
    float wr ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.15 startBaseMotionController()

```
virtual void pepperInterface::startBaseMotionController ( ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.16 startPepper()

```
virtual void pepperInterface::startPepper (
    int id ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

15.36.3.17 wait()

```
virtual void pepperInterface::wait (
    long t ) [pure virtual]
```

Implemented in [pepperRobotVirtual](#), [pepperRobot](#), and [pepperOffline](#).

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

- navmain/[pepperInterface.h](#)
- navmain/[pepperInterface.cpp](#)

15.37 pepperNavigation Class Reference

The [pepperNavigation](#) class.

```
#include <peppernavigation.h>
```

Public Member Functions

- [pepperNavigation](#) (cv::Mat &K, [dispNav](#) *dn)
pepperNavigation
- void [setfreespacenav](#) (cv::Mat &Kd)
setfreespacenav enable free space navigation
- void [setPath](#) ([kimRead](#) *KF)
setPath
- void [localise](#) (cv::Mat &I)
localise perform global localization in the path defined by reference image list
- float [navigate](#) (cv::Mat &cim)
navigate Perform Succesive Image-Based Navigation
- int [continueNav](#) (float &turnval)
continueNav check if reference images need to be switched
- void [closedisp](#) ()
closedisp

- float [getturninginkim](#) ()
getturninginkim
- float [getinitdisplacemt](#) (cv::Mat &cim)
getinitdisplacemt get lateral displacement with refernce images
- void [setofileid](#) (int id)
setofileid filenmaes for debugging
- void [setinitialpose](#) (std::vector< float > &pose)
setinitialpose set odometry pose at the start of navigation
- void [setcurrentpose](#) (std::vector< float > &pose)
setcurrentpose
- float [getvelfreespace](#) (cv::Mat &ld)
getvelfreespa current depth imagece free-space navigation
- float [getH](#) ()
getH fusion of control ref. Paper
- void [setbasevel](#) (double vt)

15.37.1 Detailed Description

The [pepperNavigation](#) class.

Definition at line 37 of file peppernavigation.h.

15.37.2 Constructor & Destructor Documentation

15.37.2.1 pepperNavigation()

```
pepperNavigation::pepperNavigation (
    cv::Mat & K,
    dispNav * dn ) [inline]
```

[pepperNavigation](#)

Parameters

K	instrinsic matrix of RGB camera
-----	---------------------------------

Definition at line 59 of file peppernavigation.h.

15.37.3 Member Function Documentation

15.37.3.1 closedisp()

```
void pepperNavigation::closedisp ( ) [inline]
closedisp
```

Definition at line 152 of file peppernavigation.h.

15.37.3.2 continueNav()

```
int pepperNavigation::continueNav (
    float & turnval ) [inline]
```

continueNav check if reference images need to be switched

Parameters

<i>turnval</i>	turning value between refernce images
----------------	---------------------------------------

Returns

0 => end of navigation 2=> switch reference images 1=> just continue

Definition at line 130 of file peppernavigation.h.

15.37.3.3 getH()

```
float pepperNavigation::getH ( ) [inline]
```

getH fusion of control ref. Paper

Returns

H

Definition at line 253 of file peppernavigation.h.

15.37.3.4 getinitdisplacemt()

```
float pepperNavigation::getinitdisplacemt (
    cv::Mat & cim ) [inline]
```

getinitdisplacemt get lateral displacement with refernce images

Parameters

<i>cim</i>	curr images
------------	-------------

Returns

displacemnt

Definition at line 173 of file peppernavigation.h.

15.37.3.5 getturninginkim()

```
float pepperNavigation::getturninginkim ( ) [inline]
```

getturninginkim

Returns

turnval

Definition at line 163 of file peppernavigation.h.

15.37.3.6 getvelfreespace()

```
float pepperNavigation::getvelfreespace (
    cv::Mat & Id ) [inline]
```

getvelfreespa current depth imagece free-space navigation

Parameters

<i>Id</i>	
-----------	--

Returns

rotational velocity to drive into free space

Definition at line 242 of file peppernavigation.h.

15.37.3.7 localise()

```
void pepperNavigation::localise (
    cv::Mat & I ) [inline]
```

localise perform global localization in the path defined by reference image list

Parameters

<i>I</i>	current RGB image
----------	-------------------

Definition at line 97 of file peppernavigation.h.

15.37.3.8 navigate()

```
float pepperNavigation::navigate (
    cv::Mat & cim ) [inline]
```

navigate Perform Succesive Image-Based Navigation

Parameters

<i>cim</i>	Current Image
------------	---------------

Returns

Rottaional Velocity

Definition at line 109 of file peppernavigation.h.

15.37.3.9 setbasevel()

```
void pepperNavigation::setbasevel (
    double vt ) [inline]
```

set base forward velocity

Definition at line 263 of file peppernavigation.h.

15.37.3.10 setcurrentpose()

```
void pepperNavigation::setcurrentpose (
    std::vector< float > & pose ) [inline]
```

setcurrentpose

Parameters

<i>pose</i>	current ododmetry reading
-------------	---------------------------

Definition at line 230 of file peppernavigation.h.

15.37.3.11 setfreespacenav()

```
void pepperNavigation::setfreespacenav (
    cv::Mat & Kd ) [inline]
```

setfreespacenav enable free space navigation

Parameters

<i>Kd</i>	instrinsic matrix of depth camera
-----------	-----------------------------------

Definition at line 75 of file peppernavigation.h.

15.37.3.12 setinitialpose()

```
void pepperNavigation::setinitialpose (
    std::vector< float > & pose ) [inline]
```

setinitialpose set odometry pose at the start of navigation

Parameters

<i>pose</i>	ododmetry reading
-------------	-------------------

Definition at line 219 of file peppernavigation.h.

15.37.3.13 setofileid()

```
void pepperNavigation::setofileid (
    int id ) [inline]
```

setofileid filenmaes for debugging

Parameters

<i>id</i>	
-----------	--

Definition at line 184 of file peppernavigation.h.

15.37.3.14 setPath()

```
void pepperNavigation::setPath (
    kimRead * KF ) [inline]
```

setPath

Parameters

<i>KF</i>	pointer of reference image list
-----------	---------------------------------

Definition at line 86 of file peppernavigation.h.

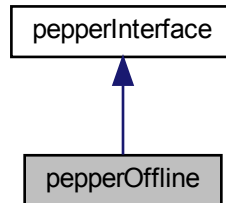
The documentation for this class was generated from the following file:

- navmain/[peppernavigation.h](#)

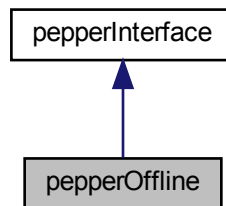
15.38 pepperOffline Class Reference

```
#include <pepperOffline.h>
```

Inheritance diagram for pepperOffline:



Collaboration diagram for pepperOffline:



Public Member Functions

- [pepperOffline](#) (const std::string &opt_ip)
 pepperOffline
- void [openCamera](#) (int id=0)
 openCamera reads image file names from the folder
- void [openDepthCamera](#) ()
- void [startBaseMotionController](#) ()
- void [initPosture](#) ()
- void [startPepper](#) (int id)
 startPepper initialies image path for image-based localizations
- void [getCurrImage](#) (cv::Mat &I)
 getCurrImage reads image from folder
- void [getCurrDepthImage](#) (cv::Mat &I)
- void [setBaseVelocities](#) (float vr, float wr)
- void [setBaseVelocities](#) (float vr, float vs, float wr)
- std::vector< float > [getOdometryReading](#) ()
- void [rotate180](#) ()

- void [wait](#) (long t)
wait delay in ms
- cv::Mat [getK](#) ()
- cv::Mat [getKd](#) ()
- int [getmode](#) ()
getmode tells it is offline mode
- void [adjusthead](#) ()
- int [getid](#) ()

15.38.1 Detailed Description

Definition at line 32 of file pepperOffline.h.

15.38.2 Constructor & Destructor Documentation

15.38.2.1 pepperOffline()

```
pepperOffline::pepperOffline (
    const std::string & opt_ip )
```

[pepperOffline](#)

Parameters

<i>opt↵ _ip</i>	= path of the folder where navigation image lies
---------------------	--

Definition at line 31 of file pepperOffline.cpp.

15.38.3 Member Function Documentation

15.38.3.1 adjusthead()

```
void pepperOffline::adjusthead ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 87 of file pepperOffline.cpp.

15.38.3.2 getCurrDepthImage()

```
void pepperOffline::getCurrDepthImage (
    cv::Mat & I ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 60 of file pepperOffline.cpp.

15.38.3.3 getCurrImage()

```
void pepperOffline::getCurrImage (
    cv::Mat & I ) [virtual]
```

getCurrImage reads image from folder

Parameters

/	
---	--

Implements [pepperInterface](#).

Definition at line 91 of file pepperOffline.cpp.

15.38.3.4 getid()

```
int pepperOffline::getid ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 125 of file pepperOffline.cpp.

15.38.3.5 getK()

```
cv::Mat pepperOffline::getK ( ) [virtual]
```

return intrinsic parameters

Implements [pepperInterface](#).

Definition at line 64 of file pepperOffline.cpp.

15.38.3.6 getKd()

```
cv::Mat pepperOffline::getKd ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 67 of file pepperOffline.cpp.

15.38.3.7 getmode()

```
int pepperOffline::getmode ( ) [virtual]
```

getmode tells it is offline mode

Returns

1

Implements [pepperInterface](#).

Definition at line 121 of file pepperOffline.cpp.

15.38.3.8 getOdometryReading()

```
std::vector< float > pepperOffline::getOdometryReading ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 107 of file pepperOffline.cpp.

15.38.3.9 initPosture()

```
void pepperOffline::initPosture ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 75 of file pepperOffline.cpp.

15.38.3.10 openCamera()

```
void pepperOffline::openCamera (
    int id = 0 ) [virtual]
```

openCamera reads image file names from the folder

Parameters

<i>id</i>	
-----------	--

Implements [pepperInterface](#).

Definition at line 49 of file pepperOffline.cpp.

15.38.3.11 openDepthCamera()

```
void pepperOffline::openDepthCamera ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 56 of file pepperOffline.cpp.

15.38.3.12 rotate180()

```
void pepperOffline::rotate180 ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 112 of file pepperOffline.cpp.

15.38.3.13 setBaseVelocities() [1/2]

```
void pepperOffline::setBaseVelocities (
    float vr,
    float vs,
    float wr ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 103 of file pepperOffline.cpp.

15.38.3.14 setBaseVelocities() [2/2]

```
void pepperOffline::setBaseVelocities (
    float vr,
    float wr ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 98 of file pepperOffline.cpp.

15.38.3.15 startBaseMotionController()

```
void pepperOffline::startBaseMotionController ( ) [virtual]
```

Implements [pepperInterface](#).

Definition at line 71 of file pepperOffline.cpp.

15.38.3.16 startPepper()

```
void pepperOffline::startPepper (
    int id = 0 ) [virtual]
```

startPepper initialies image path for image-based localizations

Parameters

<i>id</i>	
-----------	--

Implements [pepperInterface](#).

Definition at line 79 of file pepperOffline.cpp.

15.38.3.17 wait()

```
void pepperOffline::wait (
    long t ) [virtual]
```

wait delay in ms

Implements [pepperInterface](#).

Definition at line 116 of file pepperOffline.cpp.

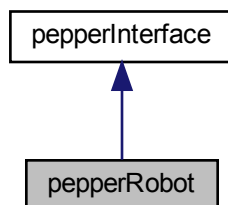
The documentation for this class was generated from the following files:

- navmain/[pepperOffline.h](#)
- navmain/[pepperOffline.cpp](#)

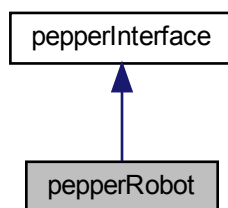
15.39 pepperRobot Class Reference

```
#include <pepperRobot.h>
```

Inheritance diagram for pepperRobot:



Collaboration diagram for pepperRobot:



Public Member Functions

- [pepperRobot](#) (const std::string &opt_ip)
pepperRobot
- void [openCamera](#) (int id=0)
openCamera Start RGB Camera
- void [openDepthCamera](#) ()
openDepthCamera start Depth Camera
- void [startBaseMotionController](#) ()
startBaseMotionController
- void [initPosture](#) ()
initPosture set initial predefined posture for navigation
- void [startPepper](#) (int id)
startPepper
- void [getCurrImage](#) (cv::Mat &I)
getCurrImage acquire image from RGB camera
- void [getCurrDepthImage](#) (cv::Mat &I)

- getCurrDepthImage* acquire image from depth camera
- void [setBaseVelocities](#) (float vr, float wr)
 - setBaseVelocities*
- void [setBaseVelocities](#) (float vr, float vs, float wr)
 - setBaseVelocities*
- std::vector< float > [getOdometryReading](#) ()
 - getOdometryReading* read current odometry reading
- void [rotate180](#) ()
 - rotate180* rotate robot by 180
- void [wait](#) (long t)
 - wait* delay
- void [adjusthead](#) ()
 - adjusthead* correct the head postion so that the robot always look forward
- cv::Mat [getK](#) ()
 - getK* intrinsic parameter of top RGB camera
- cv::Mat [getKd](#) ()
 - getKd* intrinsic parameter of depth camera
- int [getmode](#) ()
 - getmode* tells it is the online navigation mode for pepper
- int [getid](#) ()
 - getid* returns camera id

15.39.1 Detailed Description

Definition at line 30 of file pepperRobot.h.

15.39.2 Constructor & Destructor Documentation

15.39.2.1 pepperRobot()

```
pepperRobot::pepperRobot (
    const std::string & opt_ip )
```

[pepperRobot](#)

Parameters

<i>opt↔ _ip</i>	url of Pepper
---------------------	---------------

Definition at line 32 of file pepperRobot.cpp.

15.39.3 Member Function Documentation

15.39.3.1 adjusthead()

```
void pepperRobot::adjusthead ( ) [virtual]
```

adjusthead correct the head postion so that the robot always look forward

Implements [pepperInterface](#).

Definition at line 178 of file pepperRobot.cpp.

15.39.3.2 getCurrDepthImage()

```
void pepperRobot::getCurrDepthImage (
    cv::Mat & I ) [virtual]
```

getCurrDepthImage acquire image from depth camera

Parameters

/	image in cv::Mat
---	------------------

Implements [pepperInterface](#).

Definition at line 201 of file pepperRobot.cpp.

15.39.3.3 getCurrImage()

```
void pepperRobot::getCurrImage (
    cv::Mat & I ) [virtual]
```

getCurrImage acquire image from RGB camera

Parameters

/	image in cv::Mat
---	------------------

Implements [pepperInterface](#).

Definition at line 190 of file pepperRobot.cpp.

15.39.3.4 getid()

```
int pepperRobot::getid ( ) [virtual]
```

getid returns camera id

Returns

0=> top cam 1=>btom cam

Implements [pepperInterface](#).

Definition at line 286 of file pepperRobot.cpp.

15.39.3.5 getK()

```
cv::Mat pepperRobot::getK ( ) [virtual]
```

getK instrinsic parameter of top RGB camera

Returns

Intrinsic [Matrix](#)

Implements [pepperInterface](#).

Definition at line 101 of file pepperRobot.cpp.

15.39.3.6 getKd()

```
cv::Mat pepperRobot::getKd ( ) [virtual]
```

getKd instrinsic parameter of depth camera

Returns

Intrinsic [Matrix](#)

Implements [pepperInterface](#).

Definition at line 112 of file pepperRobot.cpp.

15.39.3.7 getmode()

```
int pepperRobot::getmode ( ) [virtual]
```

getmode tells it is the online navigation mode for pepper

Returns

0

Implements [pepperInterface](#).

Definition at line 276 of file pepperRobot.cpp.

15.39.3.8 getOdometryReading()

```
std::vector< float > pepperRobot::getOdometryReading ( ) [virtual]
```

getOdometryReading read current odometry reading

Returns

odometry value (x, y, theta)

Implements [pepperInterface](#).

Definition at line 235 of file pepperRobot.cpp.

15.39.3.9 initPosture()

```
void pepperRobot::initPosture ( ) [virtual]
```

initPosture set initial predefined posture for navigation

Implements [pepperInterface](#).

Definition at line 145 of file pepperRobot.cpp.

15.39.3.10 openCamera()

```
void pepperRobot::openCamera (
    int id = 0 ) [virtual]
```

openCamera Start RGB Camera

Parameters

<i>id</i>	top= 0 buttom =1
-----------	------------------

Implements [pepperInterface](#).

Definition at line 53 of file pepperRobot.cpp.

15.39.3.11 openDepthCamera()

```
void pepperRobot::openDepthCamera ( ) [virtual]
```

openDepthCamera start Depth Camera

Implements [pepperInterface](#).

Definition at line 81 of file pepperRobot.cpp.

15.39.3.12 rotate180()

```
void pepperRobot::rotate180 ( ) [virtual]
```

rotate180 rotate robot by 180

Implements [pepperInterface](#).

Definition at line 245 of file pepperRobot.cpp.

15.39.3.13 setBaseVelocities() [1/2]

```
void pepperRobot::setBaseVelocities (
    float vr,
    float vs,
    float wr ) [virtual]
```

setBaseVelocities

Parameters

<i>vr</i>	forward velocity
<i>vs</i>	lateral transaltional velocity
<i>wr</i>	rotational velocity

Implements [pepperInterface](#).

Definition at line 224 of file pepperRobot.cpp.

15.39.3.14 setBaseVelocities() [2/2]

```
void pepperRobot::setBaseVelocities (
    float vr,
    float wr ) [virtual]
```

setBaseVelocities

Parameters

<i>vr</i>	forward velocity
<i>wr</i>	rotational velocity

Implements [pepperInterface](#).

Definition at line 212 of file pepperRobot.cpp.

15.39.3.15 startBaseMotionController()

```
void pepperRobot::startBaseMotionController ( ) [virtual]
```

startBaseMotionController

Implements [pepperInterface](#).

Definition at line 122 of file pepperRobot.cpp.

15.39.3.16 startPepper()

```
void pepperRobot::startPepper (
    int id = 0 ) [virtual]
```

startPepper

Parameters

<i>id</i>	RGB camera id
-----------	---------------

Implements [pepperInterface](#).

Definition at line 166 of file pepperRobot.cpp.

15.39.3.17 wait()

```
void pepperRobot::wait (
    long t ) [virtual]
```

wait delay

Parameters

<i>t</i>	time in ms
----------	------------

Implements [pepperInterface](#).

Definition at line 296 of file pepperRobot.cpp.

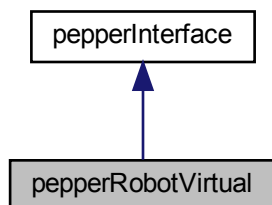
The documentation for this class was generated from the following files:

- [navmain/pepperRobot.h](#)
- [navmain/pepperRobot.cpp](#)
- [navmain/pepperRobotVirtual.cpp](#)

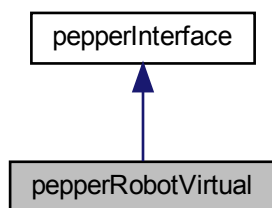
15.40 pepperRobotVirtual Class Reference

```
#include <pepperRobotVirtual.h>
```

Inheritance diagram for pepperRobotVirtual:



Collaboration diagram for pepperRobotVirtual:



Public Member Functions

- [pepperRobotVirtual](#) (const std::string &opt_ip)
 [pepperRobotVirtual](#)
- void [openCamera](#) (int id=0)
 [openCamera](#) Start RGB Camera
- void [openDepthCamera](#) ()
 [openDepthCamera](#) start Depth Camera
- void [startBaseMotionController](#) ()
 [startBaseMotionController](#)
- void [initPosture](#) ()
 [initPosture](#) set initial predefined posture for navigation
- void [startPepper](#) (int id)
 [startPepper](#)
- void [getCurrImage](#) (cv::Mat &I)
 [getCurrImage](#) acquire image from RGB camera
- void [getCurrDepthImage](#) (cv::Mat &I)

- getCurrDepthImage* acquire image from depth camera
- void [setBaseVelocities](#) (float vr, float wr)
 - setBaseVelocities*
- void [setBaseVelocities](#) (float vr, float vs, float wr)
 - setBaseVelocities*
- std::vector< float > [getOdometryReading](#) ()
 - getOdometryReading* read current odometry reading
- void [rotate180](#) ()
 - rotate180* rotate robot by 180
- void [wait](#) (long t)
 - wait* delay
- void [adjusthead](#) ()
 - adjusthead* correct the head postion so that the robot always look forward
- cv::Mat [getK](#) ()
 - getK* intrinsic parameter of top RGB camera
- cv::Mat [getKd](#) ()
 - getKd* intrinsic parameter of depth camera
- int [getmode](#) ()
 - getmode* tells it is the online navigation mode for pepper
- int [getid](#) ()
 - getid* returns camera id

15.40.1 Detailed Description

Definition at line 32 of file pepperRobotVirtual.h.

15.40.2 Constructor & Destructor Documentation

15.40.2.1 pepperRobotVirtual()

```
pepperRobotVirtual::pepperRobotVirtual (
    const std::string & opt_ip )
```

[pepperRobotVirtual](#)

[pepperRobot](#)

Parameters

<i>opt_↵</i> <i>_ip</i>	iamge folder
<i>opt_↵</i> <i>_ip</i>	url of Pepper

Definition at line 34 of file pepperRobotVirtual.cpp.

15.40.3 Member Function Documentation

15.40.3.1 adjusthead()

```
void pepperRobotVirtual::adjusthead ( ) [virtual]
```

adjusthead correct the head postion so that the robot always look forward

Implements [pepperInterface](#).

Definition at line 189 of file pepperRobotVirtual.cpp.

15.40.3.2 getCurrDepthImage()

```
void pepperRobotVirtual::getCurrDepthImage (
    cv::Mat & I ) [virtual]
```

getCurrDepthImage acquire image from depth camera

Parameters

/	image in cv::Mat
---	------------------

Implements [pepperInterface](#).

Definition at line 213 of file pepperRobotVirtual.cpp.

15.40.3.3 getCurrImage()

```
void pepperRobotVirtual::getCurrImage (
    cv::Mat & I ) [virtual]
```

getCurrImage acquire image from RGB camera

Parameters

/	image in cv::Mat
---	------------------

Implements [pepperInterface](#).

Definition at line 201 of file pepperRobotVirtual.cpp.

15.40.3.4 `getId()`

```
int pepperRobotVirtual::getId ( ) [virtual]
```

`getId` returns camera id

Returns

0=> top cam 1=>btom cam

Implements [pepperInterface](#).

Definition at line 303 of file `pepperRobotVirtual.cpp`.

15.40.3.5 `getK()`

```
cv::Mat pepperRobotVirtual::getK ( ) [virtual]
```

`getK` intrinsic parameter of top RGB camera

Returns

Intrinsic [Matrix](#)

Implements [pepperInterface](#).

Definition at line 112 of file `pepperRobotVirtual.cpp`.

15.40.3.6 `getKd()`

```
cv::Mat pepperRobotVirtual::getKd ( ) [virtual]
```

`getKd` intrinsic parameter of depth camera

Returns

Intrinsic [Matrix](#)

Implements [pepperInterface](#).

Definition at line 123 of file `pepperRobotVirtual.cpp`.

15.40.3.7 getmode()

```
int pepperRobotVirtual::getmode ( ) [virtual]
```

getmode tells it is the online navigation mode for pepper

Returns

0

1

Implements [pepperInterface](#).

Definition at line 293 of file pepperRobotVirtual.cpp.

15.40.3.8 getOdometryReading()

```
std::vector<float> pepperRobotVirtual::getOdometryReading ( ) [virtual]
```

getOdometryReading read current odometry reading

Returns

odometry value (x, y, theta)

Implements [pepperInterface](#).

15.40.3.9 initPosture()

```
void pepperRobotVirtual::initPosture ( ) [virtual]
```

initPosture set initial predefined posture for navigation

Implements [pepperInterface](#).

Definition at line 156 of file pepperRobotVirtual.cpp.

15.40.3.10 openCamera()

```
void pepperRobotVirtual::openCamera (
    int id = 0 ) [virtual]
```

openCamera Start RGB Camera

Parameters

<i>id</i>	top= 0 buttom =1
-----------	------------------

Implements [pepperInterface](#).

Definition at line 60 of file pepperRobotVirtual.cpp.

15.40.3.11 openDepthCamera()

```
void pepperRobotVirtual::openDepthCamera ( ) [virtual]
```

openDepthCamera start Depth Camera

Implements [pepperInterface](#).

Definition at line 92 of file pepperRobotVirtual.cpp.

15.40.3.12 rotate180()

```
void pepperRobotVirtual::rotate180 ( ) [virtual]
```

rotate180 rotate robot by 180

Implements [pepperInterface](#).

Definition at line 262 of file pepperRobotVirtual.cpp.

15.40.3.13 setBaseVelocities() [1/2]

```
void pepperRobotVirtual::setBaseVelocities (
    float vr,
    float vs,
    float wr ) [virtual]
```

setBaseVelocities

Parameters

<i>vr</i>	forward velocity
<i>vs</i>	lateral transaltional velocity
<i>wr</i>	rotational velocity

Implements [pepperInterface](#).

Definition at line 236 of file pepperRobotVirtual.cpp.

15.40.3.14 setBaseVelocities() [2/2]

```
void pepperRobotVirtual::setBaseVelocities (
    float vr,
    float wr ) [virtual]
```

setBaseVelocities

Parameters

<i>vr</i>	forward velocity
<i>wr</i>	rotational velocity

Implements [pepperInterface](#).

Definition at line 224 of file pepperRobotVirtual.cpp.

15.40.3.15 startBaseMotionController()

```
void pepperRobotVirtual::startBaseMotionController ( ) [virtual]
```

startBaseMotionController

Implements [pepperInterface](#).

Definition at line 133 of file pepperRobotVirtual.cpp.

15.40.3.16 startPepper()

```
void pepperRobotVirtual::startPepper (
    int id = 0 ) [virtual]
```

startPepper

Parameters

<i>id</i>	RGB camera id
-----------	---------------

Implements [pepperInterface](#).

Definition at line 177 of file pepperRobotVirtual.cpp.

15.40.3.17 `wait()`

```
void pepperRobotVirtual::wait (
    long t ) [virtual]
```

wait delay

Parameters

<i>t</i>	time in ms
----------	------------

Implements [pepperInterface](#).

Definition at line 313 of file `pepperRobotVirtual.cpp`.

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

- `navmain/pepperRobotVirtual.h`
- `navmain/pepperRobotVirtual.cpp`

15.41 `pepperServices` Class Reference

The [pepperServices](#) class.

```
#include <pepperevents.h>
```

Public Member Functions

- [pepperServices](#) (`qi::AnyObject &almemory`)
pepperServices
- void [moveCallback](#) (`const std::string &key`, `const qi::AnyValue &value`, `const qi::AnyValue &message`)
moveCallback
- void [armsCallback](#) (`const std::string &key`, `const qi::AnyValue &value`, `const qi::AnyValue &message`)
armsCallback

Public Attributes

- bool [eventraised](#)

15.41.1 Detailed Description

The [pepperServices](#) class.

Definition at line 29 of file `pepperevents.h`.

15.41.2 Constructor & Destructor Documentation

15.41.2.1 `pepperServices()`

```
pepperServices::pepperServices (
    qi::AnyObject & almemory ) [inline]
```

[pepperServices](#)

Parameters

<i>almemory</i>	AL::Memory pointer
-----------------	--------------------

Definition at line 37 of file pepperevents.h.

15.41.3 Member Function Documentation

15.41.3.1 armsCallback()

```
void pepperServices::armsCallback (
    const std::string & key,
    const qi::AnyValue & value,
    const qi::AnyValue & message ) [inline]
```

armsCallback

Definition at line 68 of file pepperevents.h.

15.41.3.2 moveCallback()

```
void pepperServices::moveCallback (
    const std::string & key,
    const qi::AnyValue & value,
    const qi::AnyValue & message ) [inline]
```

moveCallback

Definition at line 49 of file pepperevents.h.

15.41.4 Member Data Documentation

15.41.4.1 eventraised

```
bool pepperServices::eventraised
```

Definition at line 31 of file pepperevents.h.

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

- [navmain/pepperevents.h](#)

15.42 Pixel Struct Reference

```
#include <EDLineDetector.hh>
```

Public Attributes

- unsigned int [x](#)
- unsigned int [y](#)

15.42.1 Detailed Description

Definition at line 54 of file EDLineDetector.hh.

15.42.2 Member Data Documentation

15.42.2.1 [x](#)

```
unsigned int Pixel::x
```

Definition at line 55 of file EDLineDetector.hh.

15.42.2.2 [y](#)

```
unsigned int Pixel::y
```

Definition at line 56 of file EDLineDetector.hh.

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

- [linenav/EDLineDetector.hh](#)

15.43 AL::Math::Pose2D Struct Reference

A pose in a 2-dimentional space.

```
#include <alpose2d.h>
```

Public Member Functions

- [Pose2D](#) ()
Create a [Pose2D](#) initialized with 0.0f.
- [Pose2D](#) (float pInit)
Create a [Pose2D](#) initialize with the same float.
- [Pose2D](#) (float pX, float pY, float pTheta)
Create a [Pose2D](#) initialized with explicit value.
- [Pose2D](#) (const std::vector< float > &pFloats)
Create a [Pose2D](#) with an std::vector.
- [Pose2D operator+](#) (const [Pose2D](#) &pPos2) const
Overloading of operator + for [Pose2D](#).
- [Pose2D operator-](#) (const [Pose2D](#) &pPos2) const
Overloading of operator - for [Pose2D](#).
- [Pose2D operator+](#) (void) const
Overloading of operator + for [Pose2D](#).
- [Pose2D operator-](#) () const
Overloading of operator - for [Pose2D](#).
- [Pose2D & operator+=](#) (const [Pose2D](#) &pPos2)
Overloading of operator += for [Pose2D](#).
- [Pose2D & operator-=](#) (const [Pose2D](#) &pPos2)
Overloading of operator -= for [Pose2D](#).
- [Pose2D & operator*=](#) (const [Pose2D](#) &pPos2)
Overloading of operator *= for [Pose2D](#).
- [Pose2D operator*](#) (const [Pose2D](#) &pPos2) const
Overloading of operator * for [Pose2D](#).
- bool [operator==](#) (const [Pose2D](#) &pPos2) const
Overloading of operator == for [Pose2D](#).
- bool [operator!=](#) (const [Pose2D](#) &pPos2) const
Overloading of operator != for [Pose2D](#).
- [Pose2D operator*](#) (float pVal) const
Overloading of operator * for [Pose2D](#).
- [Pose2D operator/](#) (float pVal) const
Overloading of operator / for [Pose2D](#).
- [Pose2D & operator*=](#) (float pVal)
Overloading of operator *= for [Pose2D](#).
- [Pose2D & operator/=](#) (float pVal)
Overloading of operator /= for [Pose2D](#).
- float [distanceSquared](#) (const [Pose2D](#) &pPos2) const
Compute the squared distance between the actual [Pose2D](#) and the one give in argument.
- float [distance](#) (const [Pose2D](#) &pPos2) const
Compute the distance between the actual [Pose2D](#) and the one give in argument.
- [Pose2D inverse](#) () const
Return the inverse of the [Pose2D](#)
- [Pose2D diff](#) (const [Pose2D](#) &pPos2) const
Compute the [Pose2D](#) between the actual [Pose2D](#) and the one given in argument:
- bool [isNear](#) (const [Pose2D](#) &pPos2, const float &pEpsilon=0.0001f) const
Check if the actual [Pose2D](#) is near the one given in argument.
- void [toVector](#) (std::vector< float > &pReturnVector) const
Return the [Pose2D](#) as a vector of float [x, y, theta].
- std::vector< float > [toVector](#) (void) const
- void [writeToVector](#) (std::vector< float >::iterator &plt) const
Write [x, y, theta] in the vector and update the iterator. It is assumed the vector has enough space.

- float [norm](#) () const
Compute the norm of the current [Pose2D](#).
- [Pose2D normalize](#) () const
Normalize the current [Pose2D](#) position.
- float [getAngle](#) (void) const
Returns the angle of the current [Pose2D](#).

Static Public Member Functions

- static [Pose2D fromPolarCoordinates](#) (const float pRadius, const float pAngle)
Create a [Pose2D](#) from polar coordinates.

Public Attributes

- float [x](#)
- float [y](#)
- float [theta](#)

15.43.1 Detailed Description

A pose in a 2-dimentional space.

On a plane a position is totally defined by the postions x,y and the rotation theta.

Definition at line 25 of file `alpose2d.h`.

15.43.2 Constructor & Destructor Documentation

15.43.2.1 [Pose2D\(\)](#) [1/4]

```
AL::Math::Pose2D::Pose2D ( )
```

Create a [Pose2D](#) initialized with 0.0f.

$$\begin{bmatrix} x \\ y \\ theta \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

Definition at line 15 of file `alpose2d.cpp`.

15.43.2.2 [Pose2D\(\)](#) [2/4]

```
AL::Math::Pose2D::Pose2D (
    float pInit ) [explicit]
```

Create a [Pose2D](#) initialize with the same float.

$$\begin{bmatrix} x \\ y \\ theta \end{bmatrix} = \begin{bmatrix} pInit \\ pInit \\ pInit \end{bmatrix}$$

Parameters

<i>pInit</i>	the float value for each member
--------------	---------------------------------

Definition at line 17 of file `alpose2d.cpp`.

15.43.2.3 Pose2D() [3/4]

```
AL::Math::Pose2D::Pose2D (
    float pX,
    float pY,
    float pTheta ) [explicit]
```

Create a [Pose2D](#) initialized with explicit value.

$$\begin{bmatrix} x \\ y \\ theta \end{bmatrix} = \begin{bmatrix} pX \\ pY \\ pTheta \end{bmatrix}$$

Parameters

<i>pX</i>	the float value for x
<i>pY</i>	the float value for y
<i>pTheta</i>	the float value for theta

Definition at line 19 of file `alpose2d.cpp`.

15.43.2.4 Pose2D() [4/4]

```
AL::Math::Pose2D::Pose2D (
    const std::vector< float > & pFloats )
```

Create a [Pose2D](#) with an `std::vector`.

$$\begin{bmatrix} x \\ y \\ theta \end{bmatrix} = \begin{bmatrix} pFloats[0] \\ pFloats[1] \\ pFloats[2] \end{bmatrix}$$

Parameters

<i>pFloats</i>	An <code>std::vector<float></code> of size 3 for respectively:
----------------	--

x, y and theta

Definition at line 27 of file `alpose2d.cpp`.

15.43.3 Member Function Documentation

15.43.3.1 diff()

```
Pose2D AL::Math::Pose2D::diff (
    const Pose2D & pPos2 ) const
```

Compute the [Pose2D](#) between the actual [Pose2D](#) and the one given in argument:

result: $\text{inverse}(\text{pPos1}) * \text{pPos2}$

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 129 of file `alpose2d.cpp`.

15.43.3.2 distance()

```
float AL::Math::Pose2D::distance (
    const Pose2D & pPos2 ) const
```

Compute the distance between the actual [Pose2D](#) and the one give in argument.

$$\sqrt{(pPos1.x - pPos2.x)^2 + (pPos1.y - pPos2.y)^2}$$

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Returns

the float distance between the two [Pose2D](#)

Definition at line 95 of file `alpose2d.cpp`.

15.43.3.3 distanceSquared()

```
float AL::Math::Pose2D::distanceSquared (
    const Pose2D & pPos2 ) const
```

Compute the squared distance between the actual [Pose2D](#) and the one give in argument.

This avoids doing the sqrt needed for a true distance.

$$(pPos1.x - pPos2.x)^2 + (pPos1.y - pPos2.y)^2$$

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Returns

the float squared distance between the two [Pose2D](#)

Definition at line 90 of file `alpose2d.cpp`.

15.43.3.4 fromPolarCoordinates()

```
Pose2D AL::Math::Pose2D::fromPolarCoordinates (
    const float pRadius,
    const float pAngle ) [static]
```

Create a [Pose2D](#) from polar coordinates.

Parameters

<i>pRadius</i>	the polar radius
<i>pAngle</i>	the polar angle in radians

Returns

the [Pose2D](#) extracted from the polar coordinates

Definition at line 240 of file `alpose2d.cpp`.

15.43.3.5 getAngle()

```
float AL::Math::Pose2D::getAngle (
    void ) const [inline]
```

Returns the angle of the current [Pose2D](#).

result: $\text{atan2}(\text{pose.y}, \text{pose.x})$

Definition at line 322 of file `alpose2d.h`.

15.43.3.6 inverse()

```
Pose2D AL::Math::Pose2D::inverse ( ) const
```

Return the inverse of the [Pose2D](#)

Definition at line 191 of file `alpose2d.cpp`.

15.43.3.7 isNear()

```
bool AL::Math::Pose2D::isNear (
    const Pose2D & pPos2,
    const float & pEpsilon = 0.0001f ) const
```

Check if the actual [Pose2D](#) is near the one given in argument.

Parameters

<i>pPos2</i>	the second Pose2D
<i>pEpsilon</i>	an optionnal epsilon distance - default: 0.0001

Returns

true if the distance between the two [Pose2D](#) is less than pEpsilon

Definition at line 182 of file `alpose2d.cpp`.

15.43.3.8 norm()

```
float AL::Math::Pose2D::norm ( ) const [inline]
```

Compute the norm of the current [Pose2D](#).

result: $\sqrt{\text{pose.x}^2 + \text{pose.y}^2}$

Definition at line 301 of file `alpose2d.h`.

15.43.3.9 normalize()

```
Pose2D AL::Math::Pose2D::normalize ( ) const
```

Normalize the current [Pose2D](#) position.

$$pRes = \frac{pPos}{\text{norm}(pPos)}$$

Returns

the [Pose2D](#) normalized

Definition at line 156 of file `alpose2d.cpp`.

15.43.3.10 operator!=(=)

```
bool AL::Math::Pose2D::operator!= (
    const Pose2D & pPos2 ) const
```

Overloading of operator != for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 85 of file `alpose2d.cpp`.

15.43.3.11 `operator*()` [1/2]

```
Pose2D AL::Math::Pose2D::operator* (
    const Pose2D & pPos2 ) const [inline]
```

Overloading of operator `*` for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 182 of file `alpose2d.h`.

15.43.3.12 `operator*()` [2/2]

```
Pose2D AL::Math::Pose2D::operator* (
    float pVal ) const [inline]
```

Overloading of operator `*` for [Pose2D](#).

Parameters

<i>pVal</i>	the float factor
-------------	------------------

Definition at line 206 of file `alpose2d.h`.

15.43.3.13 `operator*=()` [1/2]

```
Pose2D & AL::Math::Pose2D::operator*= (
    const Pose2D & pPos2 )
```

Overloading of operator `*=` for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 48 of file `alpose2d.cpp`.

15.43.3.14 operator*=() [2/2]

```
Pose2D & AL::Math::Pose2D::operator*= (
    float pVal )
```

Overloading of operator *= for Pose2D.

Parameters

<i>pVal</i>	the float factor
-------------	------------------

Definition at line 110 of file alpose2d.cpp.

15.43.3.15 operator+() [1/2]

```
Pose2D AL::Math::Pose2D::operator+ (
    const Pose2D & pPos2 ) const [inline]
```

Overloading of operator + for Pose2D.

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-------------------

Definition at line 130 of file alpose2d.h.

15.43.3.16 operator+() [2/2]

```
Pose2D AL::Math::Pose2D::operator+ (
    void ) const [inline]
```

Overloading of operator + for Pose2D.

Definition at line 147 of file alpose2d.h.

15.43.3.17 operator+=()

```
Pose2D & AL::Math::Pose2D::operator+= (
    const Pose2D & pPos2 )
```

Overloading of operator += for Pose2D.

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 62 of file `alpose2d.cpp`.

15.43.3.18 `operator-()` [1/2]

```
Pose2D AL::Math::Pose2D::operator- ( ) const [inline]
```

Overloading of operator - for [Pose2D](#).

Definition at line 155 of file `alpose2d.h`.

15.43.3.19 `operator-()` [2/2]

```
Pose2D AL::Math::Pose2D::operator- (
    const Pose2D & pPos2 ) const [inline]
```

Overloading of operator - for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 139 of file `alpose2d.h`.

15.43.3.20 `operator-=()`

```
Pose2D & AL::Math::Pose2D::operator-= (
    const Pose2D & pPos2 )
```

Overloading of operator -= for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 70 of file `alpose2d.cpp`.

15.43.3.21 `operator/()`

```
Pose2D AL::Math::Pose2D::operator/ (
```



```
float pVal ) const
```

Overloading of operator / for [Pose2D](#).

Parameters

<i>pVal</i>	the float factor
-------------	------------------

Definition at line 100 of file `alpose2d.cpp`.

15.43.3.22 operator/=()

```
Pose2D & AL::Math::Pose2D::operator/= (
    float pVal )
```

Overloading of operator /= for [Pose2D](#).

Parameters

<i>pVal</i>	the float factor
-------------	------------------

Definition at line 118 of file `alpose2d.cpp`.

15.43.3.23 operator==()

```
bool AL::Math::Pose2D::operator== (
    const Pose2D & pPos2 ) const
```

Overloading of operator == for [Pose2D](#).

Parameters

<i>pPos2</i>	the second Pose2D
--------------	-----------------------------------

Definition at line 78 of file `alpose2d.cpp`.

15.43.3.24 toVector() [1/2]

```
void AL::Math::Pose2D::toVector (
    std::vector< float > & pReturnVector ) const
```

Return the [Pose2D](#) as a vector of float [x, y, theta].

Definition at line 134 of file `alpose2d.cpp`.

15.43.3.25 toVector() [2/2]

```
std::vector< float > AL::Math::Pose2D::toVector (
    void ) const
```

Definition at line 142 of file `alpose2d.cpp`.

15.43.3.26 writeToVector()

```
void AL::Math::Pose2D::writeToVector (
    std::vector< float >::iterator & pIt ) const
```

Write [x, y, theta] in the vector and update the iterator. It is assumed the vector has enough space.

Definition at line 149 of file `alpose2d.cpp`.

15.43.4 Member Data Documentation

15.43.4.1 theta

```
float AL::Math::Pose2D::theta
```

Definition at line 31 of file `alpose2d.h`.

15.43.4.2 x

```
float AL::Math::Pose2D::x
```

Definition at line 27 of file `alpose2d.h`.

15.43.4.3 y

```
float AL::Math::Pose2D::y
```

Definition at line 29 of file `alpose2d.h`.

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

- [depthnav/alpose2d.h](#)
- [depthnav/alpose2d.cpp](#)

15.44 SingleLine Struct Reference

```
#include <LineStructure.hh>
```

Public Attributes

- double [rho](#)
- double [theta](#)
- double [linePointX](#)
- double [linePointY](#)
- double [startPointX](#)
- double [startPointY](#)
- double [endPointX](#)
- double [endPointY](#)
- double [direction](#)
- double [gradientMagnitude](#)
- double [darkSideGrayValue](#)
- double [lightSideGrayValue](#)
- double [lineLength](#)
- double [width](#)
- int [numOfPixels](#)
- std::vector< double > [descriptor](#)
- std::vector< unsigned char > [bdescriptor](#)

15.44.1 Detailed Description

Definition at line 13 of file LineStructure.hh.

15.44.2 Member Data Documentation

15.44.2.1 bdescriptor

```
std::vector<unsigned char> SingleLine::bdescriptor
```

Definition at line 42 of file LineStructure.hh.

15.44.2.2 darkSideGrayValue

```
double SingleLine::darkSideGrayValue
```

Definition at line 30 of file LineStructure.hh.

15.44.2.3 descriptor

```
std::vector<double> SingleLine::descriptor
```

Definition at line 40 of file LineStructure.hh.

15.44.2.4 direction

```
double SingleLine::direction
```

Definition at line 26 of file LineStructure.hh.

15.44.2.5 endPointX

```
double SingleLine::endPointX
```

Definition at line 23 of file LineStructure.hh.

15.44.2.6 endPointY

```
double SingleLine::endPointY
```

Definition at line 24 of file LineStructure.hh.

15.44.2.7 gradientMagnitude

```
double SingleLine::gradientMagnitude
```

Definition at line 28 of file LineStructure.hh.

15.44.2.8 lightSideGrayValue

```
double SingleLine::lightSideGrayValue
```

Definition at line 32 of file LineStructure.hh.

15.44.2.9 lineLength

```
double SingleLine::lineLength
```

Definition at line 34 of file LineStructure.hh.

15.44.2.10 linePointX

```
double SingleLine::linePointX
```

Definition at line 18 of file LineStructure.hh.

15.44.2.11 linePointY

```
double SingleLine::linePointY
```

Definition at line 19 of file LineStructure.hh.

15.44.2.12 numOfPixels

```
int SingleLine::numOfPixels
```

Definition at line 38 of file LineStructure.hh.

15.44.2.13 rho

```
double SingleLine::rho
```

Definition at line 16 of file LineStructure.hh.

15.44.2.14 startPointX

```
double SingleLine::startPointX
```

Definition at line 21 of file LineStructure.hh.

15.44.2.15 startPointY

```
double SingleLine::startPointY
```

Definition at line 22 of file LineStructure.hh.

15.44.2.16 theta

```
double SingleLine::theta
```

Definition at line 17 of file LineStructure.hh.

15.44.2.17 width

```
double SingleLine::width
```

Definition at line 36 of file LineStructure.hh.

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

- [linenav/LineStructure.hh](#)

15.45 tgraph::topmapprocessor Class Reference

The topmapprocessor class.

```
#include <topograph_processor.h>
```

Public Member Functions

- [topmapprocessor](#) ()
topmapprocessor
- void [setNavigationPtr](#) ([navigation](#) *NAV)
setNavigationPtr Set pointers for navigation
- int [getweightofcurrenedge](#) ()
getweightofcurrenedge
- bool [getobsavoidflag](#) ()
getobsavoidflag
- int [getindexofcurrenedge](#) ()
getindexofcurrenedge location of the reference image in the edge that matches best with the current view
- int [localizeintopomap](#) (cv::Mat &cim, std::pair< int, int > &nod)
localizeintopomap find the edge in the topological map correponding to the current view of the robot
- void [kimshow](#) ([kimRead](#) *kf)
kimshow show reference images in the list
- void [process](#) (char *imfold)
process process topological garp
- int [getbranchindx](#) (int s, int e, char d)
getbranchindx read edge
- void [setpathkim](#) (std::vector< int > optpath, [kimRead](#) *kf)
setpathkim get reference image list of optimal path
- [kimRead](#) * [setpathkim](#) (std::vector< int > optpath)
setpathkim get reference image list of optimal path
- int [getdirflag](#) ()
getdirflag

15.45.1 Detailed Description

The topmaprocessor class.

Definition at line 82 of file topograph_processor.h.

15.45.2 Constructor & Destructor Documentation

15.45.2.1 topmaprocessor()

```
tgraph::topmaprocessor::topmaprocessor ( ) [inline]
```

topmaprocessor

Definition at line 222 of file topograph_processor.h.

15.45.3 Member Function Documentation

15.45.3.1 getbranchindx()

```
int tgraph::topmaprocessor::getbranchindx (
    int s,
    int e,
    char d ) [inline]
```

getbranchindx read edge

Parameters

<i>s</i>	start node
<i>e</i>	end node
<i>d</i>	dir

Returns

Definition at line 365 of file topograph_processor.h.

15.45.3.2 getdirflag()

```
int tgraph::topmaprocessor::getdirflag ( ) [inline]
```

getdirflag

Returns

Definition at line 591 of file topograph_processor.h.

15.45.3.3 getindexofcurrenedge()

```
int tgraph::topmapprocessor::getindexofcurrenedge ( ) [inline]
```

getindexofcurrenedge location of the reference image in the edge that matches best with the current view

Returns

scaled position (to maintain same scale as edge weight(

Definition at line 271 of file topograph_processor.h.

15.45.3.4 getobsavoidflag()

```
bool tgraph::topmapprocessor::getobsavoidflag ( ) [inline]
```

getobsavoidflag

Returns

true if obs avoid is used

Definition at line 262 of file topograph_processor.h.

15.45.3.5 getweightofcurrenedge()

```
int tgraph::topmapprocessor::getweightofcurrenedge ( ) [inline]
```

getweightofcurrenedge

Returns

scaled weight of current edge

Definition at line 253 of file topograph_processor.h.

15.45.3.6 kimshow()

```
void tgraph::topmapprocessor::kimshow (
    kimRead * kf ) [inline]
```

kimshow show reference images in the list

Parameters

<i>pointer</i>	to the reference image list
----------------	-----------------------------

Definition at line 329 of file topograph_processor.h.

15.45.3.7 localizeintopomap()

```
int tgraph::topmapprocessor::localizeintopomap (
    cv::Mat & cim,
    std::pair< int, int > & nod ) [inline]
```

localizeintopomap find the edge in the topological map correponding to the current view of the robot

Parameters

<i>cim</i>	current image
<i>nod</i>	current edge <startnode, endnode>

Returns

Definition at line 282 of file topograph_processor.h.

15.45.3.8 process()

```
void tgraph::topmapprocessor::process (
    char * imfold ) [inline]
```

process process topological garp

Parameters

<i>imfold</i>	folder where topological garph lies
---------------	-------------------------------------

Definition at line 350 of file topograph_processor.h.

15.45.3.9 setNavigationPtr()

```
void tgraph::topmapprocessor::setNavigationPtr (
    navigation * NAV ) [inline]
```

setNavigationPtr Set pointers for navigaton

Parameters

<i>NAV</i>	
------------	--

Definition at line 242 of file topograph_processor.h.

15.45.3.10 setpathkim() [1/2]

```
kimRead* tgraph::topmapprocessor::setpathkim (
    std::vector< int > optpath ) [inline]
```

setpathkim get reference image list of optimal path

Parameters

<i>optpath</i>	list of nodes in optimal path
<i>kf</i>	list of reference images

Definition at line 479 of file topograph_processor.h.

15.45.3.11 setpathkim() [2/2]

```
void tgraph::topmapprocessor::setpathkim (
    std::vector< int > optpath,
    kimRead * kf ) [inline]
```

setpathkim get reference image list of optimal path

Parameters

<i>optpath</i>	list of nodes in optimal path
<i>kf</i>	list of reference images

Definition at line 383 of file topograph_processor.h.

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

- [navmain/topograph_processor.h](#)

15.46 tgraph::topograph Struct Reference

The topograph struct.

```
#include <topograph_processor.h>
```

Public Attributes

- unsigned long [nbranch](#)
- std::string [directionname](#)
- unsigned int [tag](#)
- std::vector< [branch](#) > [node](#)

15.46.1 Detailed Description

The topograph struct.

Definition at line 70 of file topograph_processor.h.

15.46.2 Member Data Documentation

15.46.2.1 directionname

```
std::string tgraph::topograph::directionname
```

Definition at line 73 of file topograph_processor.h.

15.46.2.2 nbranch

```
unsigned long tgraph::topograph::nbranch
```

Definition at line 72 of file topograph_processor.h.

15.46.2.3 node

```
std::vector<branch> tgraph::topograph::node
```

Definition at line 75 of file topograph_processor.h.

15.46.2.4 tag

```
unsigned int tgraph::topograph::tag
```

Definition at line 74 of file topograph_processor.h.

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

- [navmain/topograph_processor.h](#)

15.47 vpControl Class Reference

```
#include <vpControl.hpp>
```

Public Member Functions

- [vpControl](#) (qi::SessionPtr session)
[vpControl](#) startt [vpControl](#) Service
- virtual [~vpControl](#) ()
- std::vector< float > [getJointValues](#) (std::vector< std::string > jointNames) const
- void [setDesJointVelocity](#) (std::vector< std::string > jointNames, std::vector< float > vel)
- void [setOneDesJointVelocity](#) (std::string jointName, float vel)
- void [setTask](#) ()
- qi::PeriodicTask::Callback [printTime](#) ()
- void [applyJointVelocity](#) ()
- void [start](#) ()
- void [stopJoint](#) ()
- void [stop](#) ()

15.47.1 Detailed Description

Definition at line 47 of file vpControl.hpp.

15.47.2 Constructor & Destructor Documentation

15.47.2.1 vpControl()

```
vpControl::vpControl (
    qi::SessionPtr session )
```

[vpControl](#) startt [vpControl](#) Service

Parameters

<i>session</i>	pointer to the current session
----------------	--------------------------------

Definition at line 54 of file vpControl.cpp.

15.47.2.2 ~vpControl()

```
vpControl::~vpControl ( ) [virtual]
```

Definition at line 65 of file vpControl.cpp.

15.47.3 Member Function Documentation

15.47.3.1 applyJointVelocity()

```
void vpControl::applyJointVelocity ( )
```

Definition at line 204 of file vpControl.cpp.

15.47.3.2 getJointValues()

```
std::vector< float > vpControl::getJointValues (
    std::vector< std::string > jointNames ) const
```

Definition at line 187 of file vpControl.cpp.

15.47.3.3 printTime()

```
qi::PeriodicTask::Callback vpControl::printTime ( )
```

Definition at line 194 of file vpControl.cpp.

15.47.3.4 setDesJointVelocity()

```
void vpControl::setDesJointVelocity (
    std::vector< std::string > jointNames,
    std::vector< float > vel )
```

Definition at line 104 of file vpControl.cpp.

15.47.3.5 setOneDesJointVelocity()

```
void vpControl::setOneDesJointVelocity (
    std::string jointName,
    float vel )
```

Definition at line 95 of file vpControl.cpp.

15.47.3.6 setTask()

```
void vpControl::setTask ( )
```

Definition at line 418 of file vpControl.cpp.

15.47.3.7 start()

```
void vpControl::start ( )
```

Definition at line 71 of file vpControl.cpp.

15.47.3.8 stop()

```
void vpControl::stop ( )
```

Definition at line 86 of file vpControl.cpp.

15.47.3.9 stopJoint()

```
void vpControl::stopJoint ( )
```

Definition at line 77 of file vpControl.cpp.

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

- [pepper_qi/include/vpControl.hpp](#)
- [pepper_qi/src/vpControl.cpp](#)

15.48 vpNaoqiGrabber Class Reference

```
#include <vpNaoqiGrabber.h>
```

Public Member Functions

- [vpNaoqiGrabber](#) (const qi::SessionPtr &session)
vpNaoqiGrabber constructor for grabbing image
- virtual [~vpNaoqiGrabber](#) ()
~vpNaoqiGrabber
- void [acquire](#) (cv::Mat &I)
acquire acquire RGB image
- void [acquire](#) (cv::Mat &I, struct timeval ×tamp)
- void [acquiredepth](#) (cv::Mat &I)
acquiredepth acquire depth image
- void [acquiredepth](#) (cv::Mat &I, struct timeval ×tamp)
- void [acquirevoxel](#) (cv::Mat &I)
- void [cleanup](#) ()
- std::string [getCameraName](#) ()
- unsigned int [getWidth](#) () const
- unsigned int [getHeight](#) () const
- qi::AnyObject [getProxy](#) () const
- void [open](#) ()
open Start Camera
- void [setCamera](#) (const int &camera_id)
setCamera id 0 => top RGB, 1=> bottom RGB, 2=> depth camera 2 => RawDepthColorSpace, 3 => DepthColorSpace, 4 => kDistanceColorSpace, 5 => XYZColorSpace, 6 => InfraredColorSpace. as defined in http://doc.aldebaran.com/2-5/family/pepper_technical/video_3D_pep.html
- void [setFramerate](#) (int fps)
- void [setCameraResolution](#) (const int &resolution)
- bool [setCameraParameter](#) (const int ¶meterId, const int &value)

Protected Attributes

- qi::AnyObject [m_pVideo](#)
Video proxy.
- qi::AnyObject [m_pMemory](#)
Memory proxy.
- std::string [m_handle](#)
Handle Video proxy.
- int [m_fps](#)
Requested frame per second.
- bool [m_isOpen](#)
Proxy opened status.
- int [m_width](#)
Image width.
- int [m_height](#)
Image height.
- cv::Mat [m_img](#)
Image data.
- std::string [m_cameraName](#)
Camera name.
- int [m_camerald](#)
Camera identifier.
- int [m_resolution](#)
Resolution camera.
- int [m_colorspace](#)
Colorspace.

- int [m_CV_imtype](#)
- int [m_colrspaceconvop](#)
- std::string [m_robotType](#)
Nao, Pepper or Romeo.
- bool [m_pepper](#)
True if robot is Pepper.
- bool [m_cameraMulti](#)

15.48.1 Detailed Description

Definition at line 60 of file `vpNaoqiGrabber.h`.

15.48.2 Constructor & Destructor Documentation

15.48.2.1 `vpNaoqiGrabber()`

```
vpNaoqiGrabber::vpNaoqiGrabber (
    const qi::SessionPtr & session )
```

[vpNaoqiGrabber](#) constructor for grabbing image

Parameters

<i>session</i>	current session pointer
----------------	-------------------------

Default constructor tat set the default parameters as:

- camera framerate: 30 fps
- `m_cameraId`: 0

Definition at line 57 of file `vpNaoqiGrabber.cpp`.

15.48.2.2 `~vpNaoqiGrabber()`

```
vpNaoqiGrabber::~vpNaoqiGrabber ( ) [virtual]
```

`~vpNaoqiGrabber`

Destructor that call [cleanup\(\)](#).

Definition at line 80 of file `vpNaoqiGrabber.cpp`.

15.48.3 Member Function Documentation

15.48.3.1 acquire() [1/2]

```
void vpNaoqiGrabber::acquire (
    cv::Mat & I )
```

acquire acquire RGB image

Parameters

/	image in cv::Mat
---	------------------

Definition at line 206 of file vpNaoqiGrabber.cpp.

15.48.3.2 acquire() [2/2]

```
void vpNaoqiGrabber::acquire (
    cv::Mat & I,
    struct timeval & timestamp )
```

Definition at line 219 of file vpNaoqiGrabber.cpp.

15.48.3.3 acquiredepth() [1/2]

```
void vpNaoqiGrabber::acquiredepth (
    cv::Mat & I )
```

acquiredepth acquire depth image

Parameters

/	image in cv::Mat
---	------------------

Definition at line 212 of file vpNaoqiGrabber.cpp.

15.48.3.4 acquiredepth() [2/2]

```
void vpNaoqiGrabber::acquiredepth (
    cv::Mat & I,
    struct timeval & timestamp )
```

Definition at line 275 of file vpNaoqiGrabber.cpp.

15.48.3.5 acquirevoxel()

```
void vpNaoqiGrabber::acquirevoxel (
    cv::Mat & I )
```

Definition at line 249 of file vpNaoqiGrabber.cpp.

15.48.3.6 cleanup()

```
void vpNaoqiGrabber::cleanup ( )
```

Definition at line 193 of file vpNaoqiGrabber.cpp.

15.48.3.7 getCameraName()

```
std::string vpNaoqiGrabber::getCameraName ( ) [inline]
```

Definition at line 154 of file vpNaoqiGrabber.h.

15.48.3.8 getHeight()

```
unsigned int vpNaoqiGrabber::getHeight ( ) const [inline]
```

Returns

Image height.

Definition at line 167 of file vpNaoqiGrabber.h.

15.48.3.9 getProxy()

```
qi::AnyObject vpNaoqiGrabber::getProxy ( ) const [inline]
```

Return the video device proxy used to grab images.

Definition at line 175 of file vpNaoqiGrabber.h.

15.48.3.10 getWidth()

```
unsigned int vpNaoqiGrabber::getWidth ( ) const [inline]
```

Returns

Image width.

Definition at line 159 of file vpNaoqiGrabber.h.

15.48.3.11 open()

```
void vpNaoqiGrabber::open ( )
```

open Start Camera

Definition at line 145 of file vpNaoqiGrabber.cpp.

15.48.3.12 setCamera()

```
void vpNaoqiGrabber::setCamera (
    const int & camera_id )
```

setCamera id 0 => top RGB, 1=> buttom RGB, 2-6=> depth camera 2 => RawDepthColorSpace, 3 => DepthColorSpace, 4 => kDistanceColorSpace, 5 => XYZColorSpace, 6 => InfraredColorSpace. as defined in http://doc.aldebaran.com/2-5/family/pepper_technical/video_3D_pep.html

Parameters

<i>camera↵_id</i>	Select the camera to use.
<i>camera↵_id</i>	: Camera identifier for Pepper : CameraTop(0), CameraBottom(1), CameraDepth(2)

Definition at line 90 of file vpNaoqiGrabber.cpp.

15.48.3.13 setCameraParameter()

```
bool vpNaoqiGrabber::setCameraParameter (
    const int & parameterId,
    const int & value )
```

Set a camera parameter.

Parameters

<i>parameter↔ id</i>	: Camera parameter requested.
<i>value</i>	: Value requested.

Returns

True if succesfull.

Definition at line 315 of file vpNaoqiGrabber.cpp.

15.48.3.14 setCameraResolution()

```
void vpNaoqiGrabber::setCameraResolution (
    const int & resolution ) [inline]
```

Set the camera resolution.

Parameters

<i>resolution</i>	: Index camera resolution.
-------------------	----------------------------

See also

[open\(\)](#)

Definition at line 214 of file vpNaoqiGrabber.h.

15.48.3.15 setFrameRate()

```
void vpNaoqiGrabber::setFrameRate (
    int fps ) [inline]
```

Set the camera framerate. In the constructor, the default framerate is set to 30 Hz.

Parameters

<i>fps</i>	: New framerate in Hz.
------------	------------------------

See also

[open\(\)](#)

Definition at line 202 of file vpNaoqiGrabber.h.

15.48.4 Member Data Documentation

15.48.4.1 m_cameraId

```
int vpNaoqiGrabber::m_cameraId [protected]
```

Camera identifier.

Definition at line 110 of file vpNaoqiGrabber.h.

15.48.4.2 m_cameraMulti

```
bool vpNaoqiGrabber::m_cameraMulti [protected]
```

Definition at line 119 of file vpNaoqiGrabber.h.

15.48.4.3 m_cameraName

```
std::string vpNaoqiGrabber::m_cameraName [protected]
```

Camera name.

Definition at line 109 of file vpNaoqiGrabber.h.

15.48.4.4 m_colorspace

```
int vpNaoqiGrabber::m_colorspace [protected]
```

Colorspace.

Definition at line 112 of file vpNaoqiGrabber.h.

15.48.4.5 m_colrspaceconvop

```
int vpNaoqiGrabber::m_colrspaceconvop [protected]
```

Definition at line 114 of file vpNaoqiGrabber.h.

15.48.4.6 m_CV_imtype

```
int vpNaoqiGrabber::m_CV_imtype [protected]
```

Definition at line 113 of file vpNaoqiGrabber.h.

15.48.4.7 m_fps

```
int vpNaoqiGrabber::m_fps [protected]
```

Requested frame per second.

Definition at line 104 of file vpNaoqiGrabber.h.

15.48.4.8 m_handle

```
std::string vpNaoqiGrabber::m_handle [protected]
```

Handle Video proxy.

Definition at line 103 of file vpNaoqiGrabber.h.

15.48.4.9 m_height

```
int vpNaoqiGrabber::m_height [protected]
```

Image height.

Definition at line 107 of file vpNaoqiGrabber.h.

15.48.4.10 m_img

```
cv::Mat vpNaoqiGrabber::m_img [protected]
```

Image data.

Definition at line 108 of file vpNaoqiGrabber.h.

15.48.4.11 m_isOpen

```
bool vpNaoqiGrabber::m_isOpen [protected]
```

Proxy opened status.

Definition at line 105 of file vpNaoqiGrabber.h.

15.48.4.12 m_pepper

```
bool vpNaoqiGrabber::m_pepper [protected]
```

True if robot is Pepper.

Definition at line 116 of file vpNaoqiGrabber.h.

15.48.4.13 m_pMemory

```
qi::AnyObject vpNaoqiGrabber::m_pMemory [protected]
```

Memory proxy.

Definition at line 102 of file vpNaoqiGrabber.h.

15.48.4.14 m_pVideo

```
qi::AnyObject vpNaoqiGrabber::m_pVideo [protected]
```

Video proxy.

Definition at line 101 of file vpNaoqiGrabber.h.

15.48.4.15 m_resolution

```
int vpNaoqiGrabber::m_resolution [protected]
```

Resolution camera.

Definition at line 111 of file vpNaoqiGrabber.h.

15.48.4.16 m_robotType

```
std::string vpNaoqiGrabber::m_robotType [protected]
```

Nao, Pepper or Romeo.

Definition at line 115 of file vpNaoqiGrabber.h.

15.48.4.17 m_width

```
int vpNaoqiGrabber::m_width [protected]
```

Image width.

Definition at line 106 of file vpNaoqiGrabber.h.

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

- pepper_qi/include/vpNaoqiGrabber.h
- pepper_qi/src/vpNaoqiGrabber.cpp

15.49 vpNaoqiRobot Class Reference

```
#include <vpNaoqiRobot.h>
```

Public Types

- enum [RobotType](#) { [Pepper](#), [Unknown](#) }

Public Member Functions

- [vpNaoqiRobot](#) (const qi::SessionPtr &session)
- virtual [~vpNaoqiRobot](#) ()
- void [cleanup](#) ()
- std::vector< float > [getAngles](#) (const std::string &name, const bool &useSensors=true) const
- std::vector< float > [getAngles](#) (const std::vector< std::string > &name, const bool &useSensors=true) const
- std::vector< std::string > [getBodyNames](#) (const std::string &names) const
- std::vector< float > [getPosition](#) (const std::string &names, const bool &useSensors=true) const
- void [getPosition](#) (const std::vector< std::string > &names, std::vector< float > &q, const bool &useSensors=true) const
- qi::AnyObject * [getMotionProxy](#) ()
- std::vector< std::vector< float > > [getLimits](#) (const std::string &name) const
- std::string [getRobotName](#) () const
- [RobotType](#) [getRobotType](#) () const
- void [open](#) ()
open robot
- void [moveTo](#) (const float &x, const float &y, const float &theta) const
moveTo move robot to the specified (x,y,theta) from current position
- bool [rotate180](#) () const
rotate180 rotate robot by 180

- void [setCollisionProtection](#) (bool protection) const
- void [setExternalCollisionProtectionEnabled](#) (const std::string &name, const bool &enable) const
- void [setPosition](#) (const std::string &name, const float &angles, const float &fractionMaxSpeed) const
- void [setPosition](#) (const std::vector< std::string > &names, const std::vector< float > &jointPosition, const float &fractionMaxSpeed) const
- void [setStiffness](#) (const std::string &names, const float &stiffness) const
- void [setStiffness](#) (const std::vector< std::string > &names, const std::vector< float > &stiffness) const
- void [setStiffness](#) (const std::vector< std::string > &names, const float &stiffness) const
- void [setVelocity](#) (const std::vector< std::string > &names, const std::vector< float > &jointVel) const
- void [getJointVelocity](#) (const std::vector< std::string > &names, std::vector< float > &jointVel) const
- void [setBaseVelocity](#) (const std::vector< float > &jointVel) const
setBaseVelocity set velocity of the base
- void [setBaseVelocity](#) (const float &vx, const float &vy, const float &wz) const
- void [startPepperControl](#) () const
- void [stop](#) (const std::string &name) const
- void [stop](#) (const std::vector< std::string > &names) const
- void [stopPepperControl](#) () const
- void [stopBase](#) () const

Protected Attributes

- qi::AnyObject [m_pMemory](#)
Memory proxy.
- qi::AnyObject [m_pMotion](#)
Motion proxy.
- qi::AnyObject [m_pepper_control](#)
Proxy to Pepper_control.
- bool [m_isOpen](#)
Proxy opened status.
- bool [m_collisionProtection](#)
Collision protection enabling status.
- std::string [m_robotName](#)
Name of the robot.
- [RobotType](#) [m_robotType](#)
Indicate if the robot is Pepper.

15.49.1 Detailed Description

Definition at line 60 of file vpNaoqiRobot.h.

15.49.2 Member Enumeration Documentation

15.49.2.1 RobotType

```
enum vpNaoqiRobot::RobotType
```

Enumerator

Pepper	
Unknown	

Definition at line 64 of file vpNaoqiRobot.h.

15.49.3 Constructor & Destructor Documentation

15.49.3.1 vpNaoqiRobot()

```
vpNaoqiRobot::vpNaoqiRobot (
    const qi::SessionPtr & session )
```

Default constructor that set the default parameters as:

- robot ip address: "198.18.0.1"
- collision protection: enabled

Definition at line 58 of file vpNaoqiRobot.cpp.

15.49.3.2 ~vpNaoqiRobot()

```
vpNaoqiRobot::~vpNaoqiRobot ( ) [virtual]
```

Destructor that call [cleanup\(\)](#).

Definition at line 81 of file vpNaoqiRobot.cpp.

15.49.4 Member Function Documentation

15.49.4.1 cleanup()

```
void vpNaoqiRobot::cleanup ( )
```

Destroy the connexion to the motion proxy.

Definition at line 133 of file vpNaoqiRobot.cpp.

15.49.4.2 getAngles() [1/2]

```
std::vector< float > vpNaoqiRobot::getAngles (
    const std::string & name,
    const bool & useSensors = true ) const
```

get joint angles

Get the value of all the joints of the chain.

Parameters

<i>names</i>	: Names the joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
--------------	--

Returns

The value of the joints.

Definition at line 456 of file vpNaoqiRobot.cpp.

15.49.4.3 getAngles() [2/2]

```
std::vector< float > vpNaoqiRobot::getAngles (
    const std::vector< std::string > & name,
    const bool & useSensors = true ) const
```

Get the value of all the joints in the vector.

Parameters

<i>names</i>	: Vector containing the names of the joints.
--------------	--

Returns

The value of the joints.

Definition at line 467 of file vpNaoqiRobot.cpp.

15.49.4.4 getBodyNames()

```
std::vector< std::string > vpNaoqiRobot::getBodyNames (
    const std::string & names ) const
```

get body names

Get the name of all the joints of the chain.

Parameters

<i>names</i>	: Names the joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
--------------	--

Returns

The name of the joints.

Definition at line 482 of file vpNaoqiRobot.cpp.

15.49.4.5 getJointVelocity()

```
void vpNaoqiRobot::getJointVelocity (
    const std::vector< std::string > & names,
    std::vector< float > & jointVel ) const
```

Get the joints velocities.

Parameters

<i>names</i>	: Vector with the joint names.
<i>names</i>	: Vector to fill with the joint velocities.

Definition at line 586 of file vpNaoqiRobot.cpp.

15.49.4.6 getLimits()

```
std::vector< std::vector< float > > vpNaoqiRobot::getLimits (
    const std::string & name ) const
```

Get the minAngle (rad), maxAngle (rad), maxVelocity (rad.s-1) and maxTorque (N.m). for a given joint or actuator in the body.

Parameters

<i>name</i>	: Name of a joint, chain, "Body", "JointActuators", "Joints" or "Actuators".
-------------	--

Returns

Vector containing the minAngle, maxAngle, maxVelocity and maxTorque for all the joints specified.

Definition at line 525 of file vpNaoqiRobot.cpp.

15.49.4.7 getMotionProxy()

```
qi::AnyObject * vpNaoqiRobot::getMotionProxy ( )
```

getMotionProxy

Get the motion proxy

Returns

pointer to the motion proxy

Definition at line 498 of file vpNaoqiRobot.cpp.

15.49.4.8 getPosition() [1/2]

```
std::vector<float> vpNaoqiRobot::getPosition (
    const std::string & names,
    const bool & useSensors = true ) const
```

get Poistions of different joints/ odeometry

15.49.4.9 getPosition() [2/2]

```
void vpNaoqiRobot::getPosition (
    const std::vector< std::string > & names,
    std::vector< float > & q,
    const bool & useSensors = true ) const
```

Get the position of all the joints in the vector.

Parameters

<i>names</i>	: Names the joints.
<i>useSensors</i>	: If true, sensor positions will be returned. If false, it will be the command.
<i>q</i>	: Joint position in radians.

Definition at line 513 of file vpNaoqiRobot.cpp.

15.49.4.10 getRobotName()

```
std::string vpNaoqiRobot::getRobotName ( ) const [inline]
```

get robot name and type=> supported Pepper, Nao, Romeo2

Definition at line 122 of file vpNaoqiRobot.h.

15.49.4.11 getRobotType()

```
RobotType vpNaoqiRobot::getRobotType ( ) const [inline]
```

Definition at line 124 of file vpNaoqiRobot.h.

15.49.4.12 moveTo()

```
void vpNaoqiRobot::moveTo (
    const float & x,
    const float & y,
    const float & theta ) const
```

moveTo move robot to the specified (x,y,theta) from current position

Parameters

<i>x</i>	
<i>y</i>	
<i>theta</i>	Apply a velocity vector to a vector of joints. Use just one call to apply the velocities.
<i>names</i>	: Names the joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
<i>jointVel</i>	: Joint velocity vector with values expressed in rad/s (vpColVector).
<i>verbose</i>	: If true activates printings.

Apply a velocity vector to a vector of joints. Use just one call to apply the velocities.

Parameters

<i>names</i>	: Names the joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
<i>jointVel</i>	: Joint velocity vector with values expressed in rad/s.
<i>verbose</i>	: If true activates printings.

Makes the robot move to the given pose in the ground plane, relative to FRAME_ROBOT. This is a blocking call.

Parameters

<i>x</i>	: Distance along the X axis in meters.
<i>y</i>	: Distance along the Y axis in meters.
<i>theta</i>	: Rotation around the Z axis in radians [-3.1415 to 3.1415]

Definition at line 349 of file vpNaoqiRobot.cpp.

15.49.4.13 open()

```
void vpNaoqiRobot::open ( )
```

open robot

Open the connection with the robot. All the parameters should be set before calling this function.

```
int main(int argc, char** argv)
{
    std::string opt_ip = "192.168.0.24";
    for (unsigned int i=0; i<argc; i++) {
        if (std::string(argv[i]) == "--ip")
            opt_ip = argv[i+1];
        else if (std::string(argv[i]) == "--help") {
            std::cout << "Usage: " << argv[0] << "[--ip <robot address>] " << std::endl;
            return 0;
        }
    }
    // Create a session to connect with the Robot
    qi::SessionPtr session = qi::makeSession();
    std::string ip_port = "tcp://" + opt_ip + ":9559";
    session->connect(ip_port);
    if (!opt_ip.empty()) {
        std::cout << "Connect to robot with ip address: " << opt_ip << std::endl;
    }
    vpNaoqiRobot robot(session);
    robot.open();
    return 0;
}
```

Definition at line 118 of file vpNaoqiRobot.cpp.

15.49.4.14 rotate180()

```
bool vpNaoqiRobot::rotate180 ( ) const
```

rotate180 rotate robot by 180

Returns

sucees or fail

Definition at line 355 of file vpNaoqiRobot.cpp.

15.49.4.15 setBaseVelocity() [1/2]

```
void vpNaoqiRobot::setBaseVelocity (
    const float & vx,
    const float & vy,
    const float & wz ) const
```

Apply a velocity Vx, Vy, Wz to Pepper.

Parameters

<i>vel</i>	: Joint velocity vector with values expressed in rad/s.
------------	---

Definition at line 438 of file vpNaoqiRobot.cpp.

15.49.4.16 setBaseVelocity() [2/2]

```
void vpNaoqiRobot::setBaseVelocity (
    const std::vector< float > & jointVel ) const
```

setBaseVelocity set velocity of the base

Parameters

<i>jointVel</i>	vx, vy,wz => translation velocities and roational velocities. Note: vy and wz can't be set together at a same time
-----------------	--

Apply a velocity Vx, Vy, Wz to Pepper.

Parameters

<i>vel</i>	: Joint velocity vector with values expressed in rad/s.
------------	---

Definition at line 419 of file vpNaoqiRobot.cpp.

15.49.4.17 setCollisionProtection()

```
void vpNaoqiRobot::setCollisionProtection (
    bool protection ) [inline]
```

Enable/disable the collision protection. In the constructor, the collision protection is enabled by default.

Parameters

<i>protection</i>	: true to enable collision protection, false to disable.
-------------------	--

Definition at line 154 of file vpNaoqiRobot.h.

15.49.4.18 setExternalCollisionProtectionEnabled()

```
void vpNaoqiRobot::setExternalCollisionProtectionEnabled (
    const std::string & name,
    const bool & enable ) const
```

Set External collision

Parameters

<i>name</i>	: The name {"All", "Move", "Arms", "LArm" or "RArm"}.
<i>enable</i>	Activate or deactivate the external collision of the desired name.

Definition at line 541 of file vpNaoqiRobot.cpp.

15.49.4.19 setPosition() [1/2]

```
void vpNaoqiRobot::setPosition (
    const std::string & name,
    const float & angle,
    const float & fractionMaxSpeed ) const
```

Set the position of all the joints of the chain.

Parameters

<i>names</i>	: Names the chain.
<i>angles</i>	: Joint positions in radians.
<i>fractionMaxSpeed</i>	: The fraction of maximum speed to use. Value should be comprised between 0 and 1.

Definition at line 554 of file vpNaoqiRobot.cpp.

15.49.4.20 setPosition() [2/2]

```
void vpNaoqiRobot::setPosition (
    const std::vector< std::string > & names,
    const std::vector< float > & jointPosition,
    const float & fractionMaxSpeed ) const
```

Set joint positions.

Parameters

<i>names</i>	: std::vector with the names the joints.
<i>jointPosition</i>	: Joint positions in radians.
<i>fractionMaxSpeed</i>	: The fraction of maximum speed to use. Value should be comprised between 0 and 1.

Definition at line 569 of file vpNaoqiRobot.cpp.

15.49.4.21 setStiffness() [1/3]

```
void vpNaoqiRobot::setStiffness (
    const std::string & names,
    const float & stiffness ) const
```

setStiffness of joints

Set the stiffness to a chain name, or to a specific joint.

Parameters

<i>names</i>	: Names of the chains, "Body", "JointActuators", "Joints" or "Actuators".
<i>stiffness</i>	: Stiffness parameter that should be between 0 (no stiffness) and 1 (full stiffness).

Definition at line 152 of file vpNaoqiRobot.cpp.

15.49.4.22 setStiffness() [2/3]

```
void vpNaoqiRobot::setStiffness (
    const std::vector< std::string > & names,
    const float & stiffness ) const
```

Set the stiffness of a list of joints.

Parameters

<i>names</i>	: Vector with the joint names.
<i>stiffness</i>	: Stiffness parameter that should be between 0 (no stiffness) and 1 (full stiffness).

Definition at line 175 of file vpNaoqiRobot.cpp.

15.49.4.23 setStiffness() [3/3]

```
void vpNaoqiRobot::setStiffness (
    const std::vector< std::string > & names,
    const std::vector< float > & stiffness ) const
```

Set the stiffness of a list of joints.

Parameters

<i>names</i>	: Vector with the joint names.
<i>stiffness</i>	: Stiffness parameters that should be between 0 (no stiffness) and 1 (full stiffness).

Definition at line 164 of file vpNaoqiRobot.cpp.

15.49.4.24 setVelocity()

```
void vpNaoqiRobot::setVelocity (
    const std::vector< std::string > & names,
    const std::vector< float > & jointVel ) const
```

Apply a velocity vector to a vector of joints.

Parameters

<i>names</i>	: Names the joints, chains, "Body", "JointActuators", "Joints" or "Actuators".
<i>jointVel</i>	: Joint velocity vector with values expressed in rad/s.
<i>verbose</i>	: If true activates printings.

Definition at line 196 of file vpNaoqiRobot.cpp.

15.49.4.25 startPepperControl()

```
void vpNaoqiRobot::startPepperControl ( ) const
```

stop Pepper

Start the service pepper_control.

Definition at line 393 of file vpNaoqiRobot.cpp.

15.49.4.26 stop() [1/2]

```
void vpNaoqiRobot::stop (
    const std::string & name ) const
```

Stop joint in a chain.

Parameters

<i>names</i>	: Chain or joint name.
--------------	------------------------

Definition at line 376 of file vpNaoqiRobot.cpp.

15.49.4.27 stop() [2/2]

```
void vpNaoqiRobot::stop (
    const std::vector< std::string > & names ) const
```

Stop joints.

Parameters

<i>names</i>	: Vector with the joint names.
--------------	--------------------------------

Definition at line 365 of file vpNaoqiRobot.cpp.

15.49.4.28 stopBase()

```
void vpNaoqiRobot::stopBase ( ) const
```

Stop the velocity of the base.

Definition at line 402 of file vpNaoqiRobot.cpp.

15.49.4.29 stopPepperControl()

```
void vpNaoqiRobot::stopPepperControl ( ) const
```

Stop the service pepper_control.

Definition at line 385 of file vpNaoqiRobot.cpp.

15.49.5 Member Data Documentation**15.49.5.1 m_collisionProtection**

```
bool vpNaoqiRobot::m_collisionProtection [protected]
```

Collision protection enabling status.

Definition at line 76 of file vpNaoqiRobot.h.

15.49.5.2 m_isOpen

```
bool vpNaoqiRobot::m_isOpen [protected]
```

Proxy opened status.

Definition at line 75 of file vpNaoqiRobot.h.

15.49.5.3 m_pepper_control

```
qi::AnyObject vpNaoqiRobot::m_pepper_control [protected]
```

Proxy to Pepper_control.

Definition at line 73 of file vpNaoqiRobot.h.

15.49.5.4 m_pMemory

```
qi::AnyObject vpNaoqiRobot::m_pMemory [protected]
```

Memory proxy.

Definition at line 71 of file vpNaoqiRobot.h.

15.49.5.5 m_pMotion

```
qi::AnyObject vpNaoqiRobot::m_pMotion [protected]
```

Motion proxy.

Definition at line 72 of file vpNaoqiRobot.h.

15.49.5.6 m_robotName

```
std::string vpNaoqiRobot::m_robotName [protected]
```

Name of the robot.

Definition at line 77 of file vpNaoqiRobot.h.

15.49.5.7 m_robotType

`RobotType` `vpNaoqiRobot::m_robotType` [protected]

Indicate if the robot is Pepper.

Definition at line 78 of file `vpNaoqiRobot.h`.

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

- `pepper_qi/include/vpNaoqiRobot.h`
- `pepper_qi/src/vpNaoqiRobot.cpp`

15.50 astar::weight_writer< WeightMap > Class Template Reference

```
#include <topograph_astar.h>
```

Public Member Functions

- `weight_writer` (`WeightMap` w)
- `template<class Edge >`
void `operator()` (`std::ostream &out`, `const Edge &e`) `const`

15.50.1 Detailed Description

```
template<class WeightMap>
class astar::weight_writer< WeightMap >
```

Definition at line 75 of file `topograph_astar.h`.

15.50.2 Constructor & Destructor Documentation

15.50.2.1 weight_writer()

```
template<class WeightMap >
astar::weight_writer< WeightMap >::weight_writer (
    WeightMap w ) [inline]
```

Definition at line 77 of file `topograph_astar.h`.

15.50.3 Member Function Documentation

15.50.3.1 operator>()

```
template<class WeightMap >
template<class Edge >
void astar::weight_writer< WeightMap >::operator() (
    std::ostream & out,
    const Edge & e ) const [inline]
```

Definition at line 79 of file `topograph_astar.h`.

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

- `navmain/topograph_astar.h`

Chapter 16

File Documentation

16.1 cmake/BIAS_CMakeLists.txt File Reference

16.2 cmake/libqgv_CMakeLists.txt File Reference

Functions

- [find_package](#) (Qt5 REQUIRED COMPONENTS Core Widgets Test) set(CMAKE_INCLUDE_CURRENT_DIR ON) [SET](#)(qgvlib_CPP qgv/private/QGVCore.cpp qgv/private/QGVGraphPrivate.cpp qgv/private/QGVEdgePrivate.cpp qgv/private/QGVGvcPrivate.cpp qgv/private/QGVNodePrivate.cpp qgv/QGVEdge.cpp qgv/QGVNode.cpp qgv/QGVScene.cpp qgv/QGVSubGraph.cpp) INCLUDE_DIRECTORIES(\$
- qgv qgv private [SET](#) (QT_LIBRARIES Qt5::Core Qt5::Widgets Qt5::Test) ADD_LIBRARY(qgvcore SHARED \$
- [TARGET_LINK_LIBRARIES](#) (qgvcore \${QT_LIBRARIES} \${GRAPHVIZ_GVC_LIBRARY} \${GRAPHVIZ_CGRAPH_LIBRARY} \${GRAPHVIZ_CDT_LIBRARY}) [SET](#)(qgv_LIBS \$

16.2.1 Function Documentation

16.2.1.1 find_package()

```
find_package (
    Qt5 REQUIRED COMPONENTS Core Widgets Test )
```

Definition at line 6 of file libqgv_CMakeLists.txt.

16.2.1.2 SET()

```
qgv qgv private SET (
    QT_LIBRARIES Qt5::Core Qt5::Widgets Qt5::Test )
```

Definition at line 31 of file libqgv_CMakeLists.txt.

16.2.1.3 TARGET_LINK_LIBRARIES()

```
TARGET_LINK_LIBRARIES (
    qgvcore ${QT_LIBRARIES} ${GRAPHVIZ_GVC_LIBRARY} ${GRAPHVIZ_CGRAPH_LIBRARY} ${GRAPHVIZ_CDT_
_LIBRARY} )
```

Definition at line 42 of file libqgv_CMakeLists.txt.

16.3 CMakeLists.txt File Reference

Functions

- [cmake_minimum_required](#) (VERSION 3.0.0) project(PepperNavigation) set(CMAKE_PREFIX_PATH/home/suman/soft/third_↵_party/opencv/install/lib/cmake/opencv4/home/suman/soft/third_party/naoqi/install/share/naoqi_libqi/cmake/home/suman/soft/third_↵_party/naoqi/install/share/naoqi_libqicore/cmake) set(CMAKE_AUTOMOC ON) set(CMAKE_AUTOUIC ON) set(CM↵AKE_INCLUDE_CURRENT_DIR ON) set(CMAKE_BUILD_TYPE RelWithDebInfo) [option](#)(USE_BIAS_LIBRARY "Use BIAS library for line matching [if](#) not OpenCV version is used" ON) [option](#)(USE_SYSTEM_ARPAK_SUPERLU "Use [system](#) Arpac
- blas and superlu OFF [option](#) (RUN_INSIDE_PEPPER "Run on board" OFF) [if](#)(RUN_INSIDE_PEPPER) set(USE_SY↵STEM_ARPAK_SUPERLU 0) [endif](#)() [if](#)(USE_BIAS_LIBRARY) include(\$
- cmake BIAS_CMakeLists.txt [endif](#) () [if](#)(NOT RUN_INSIDE_PEPPER) include(\$

Variables

- [lapack](#)

16.3.1 Function Documentation

16.3.1.1 cmake_minimum_required()

```
cmake_minimum_required (
    VERSION 3.0.  0 )
```

16.3.1.2 endif()

```
cmake libqgv_CMakeLists.txt endif ( )
```

Definition at line 34 of file CMakeLists.txt.

16.3.1.3 option()

```
blas and superlu OFF option (
    RUN_INSIDE_PEPPER "Run on board" OFF )
```

Definition at line 25 of file CMakeLists.txt.

16.3.2 Variable Documentation

16.3.2.1 lapack

lapack

Definition at line 24 of file CMakeLists.txt.

16.4 depthnav/CMakeLists.txt File Reference

Functions

- [cmake_minimum_required](#) (VERSION 3.0) project(pepper_fsnav) set(CMAKE_PREFIX_PATH/home/suman/soft/thirdparty/OpenCV/install_lat/lib/cmake/opencv4/opt/ros/kinetic/share/naoqi_libqi/cmake/opt/ros/kinetic/share/naoqi_libqicore) set(CMAKE_BUILD_TYPE "Relwithdebinfo") find_package(OpenCV 4 REQUIRED) [find_package](#)(naoqi_libqi) [find_package](#)(naoqi_libqicore) [find_package](#)(Boost COMPONENTS filesystem [system](#) REQUIRED) include_directories(".") include_directories(SYSTEM \$

16.4.1 Function Documentation

16.4.1.1 cmake_minimum_required()

```
cmake_minimum_required (
    VERSION 3. 0 )
```

Definition at line 6 of file CMakeLists.txt.

16.5 pepper_qi/CMakeLists.txt File Reference

Functions

- [cmake_minimum_required](#) (VERSION 2.6.4 FATAL_ERROR) project(naoqi_ocv) set(CMAKE_BUILD_TYPE "Release" CACHE String "Choose the type of build

16.5.1 Function Documentation

16.5.1.1 cmake_minimum_required()

```
cmake_minimum_required (
    VERSION 2.6.4 FATAL_ERROR )
```


16.6 data/README.md File Reference

16.7 depthnav/README.md File Reference

16.8 linenav/README.md File Reference

16.9 mapping/README.md File Reference

16.10 navmain/maingui/README.md File Reference

16.11 navmain/README.md File Reference

16.12 pepper_qi/README.md File Reference

16.13 qgv/README.md File Reference

16.14 README.md File Reference

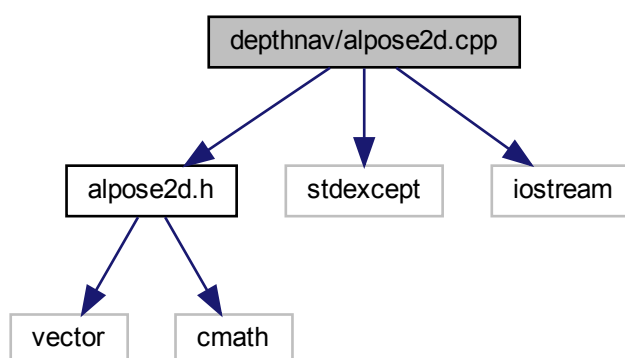
16.15 depthnav/alpose2d.cpp File Reference

```
#include "alpose2d.h"
```

```
#include <stdexcept>
```

```
#include <iostream>
```

Include dependency graph for alpose2d.cpp:



Namespaces

- [AL](#)
- [AL::Math](#)

Functions

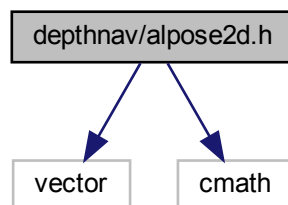
- float [AL::Math::distanceSquared](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the squared distance between two [Pose2D](#).
- float [AL::Math::distance](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the distance between two [Pose2D](#).
- void [AL::Math::pose2DInverse](#) (const Pose2D &pPos, Pose2D &pRes)
Compute the inverse of a [Pose2D](#).
- void [AL::Math::pose2dInvertInPlace](#) (Pose2D &pPos)
Inverse the given [Pose2D](#) in place:
- Pose2D [AL::Math::pose2dDiff](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the [Pose2D](#) between the actual [Pose2D](#) and the one give in argument result:
- Pose2D [AL::Math::pose2DInverse](#) (const Pose2D &pPos)
Compute the inverse of a [Pose2D](#).
- Pose2D [AL::Math::pinv](#) (const Pose2D &pPos)
Alternative name for inverse: return the pose2d inverse of the given [Pose2D](#).

16.16 depthnav/alpose2d.h File Reference

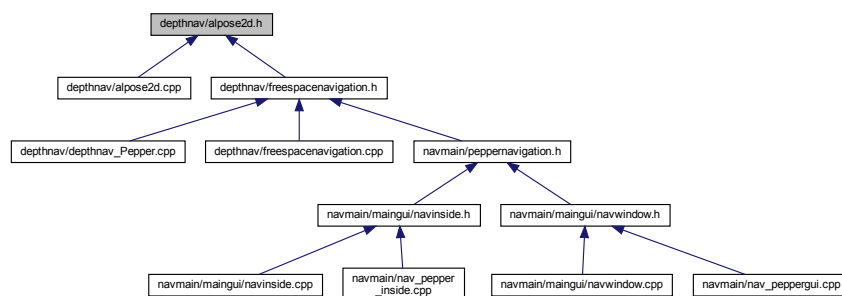
```
#include <vector>
```

```
#include <cmath>
```

Include dependency graph for alpose2d.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [AL::Math::Pose2D](#)
A pose in a 2-dimentional space.

Namespaces

- [AL](#)
- [AL::Math](#)

Macros

- [#define _LIBALMATH_ALMATH_TYPES_ALPOSE2D_H_](#)

Functions

- float [AL::Math::distanceSquared](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the squared distance between two [Pose2D](#).
- float [AL::Math::distance](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the distance between two [Pose2D](#).
- void [AL::Math::pose2dInvertInPlace](#) (Pose2D &pPos)
Inverse the given [Pose2D](#) in place:
- Pose2D [AL::Math::pinv](#) (const Pose2D &pPos)
Alternative name for inverse: return the pose2d inverse of the given [Pose2D](#).
- Pose2D [AL::Math::pose2dDiff](#) (const Pose2D &pPos1, const Pose2D &pPos2)
Compute the [Pose2D](#) between the actual [Pose2D](#) and the one give in argument result:
- Pose2D [AL::Math::pose2DInverse](#) (const Pose2D &pPos)
Compute the inverse of a [Pose2D](#).
- void [AL::Math::pose2DInverse](#) (const Pose2D &pPos, Pose2D &pRes)
Compute the inverse of a [Pose2D](#).

16.16.1 Macro Definition Documentation

16.16.1.1 [_LIBALMATH_ALMATH_TYPES_ALPOSE2D_H_](#)

```
#define _LIBALMATH_ALMATH_TYPES_ALPOSE2D_H_
```

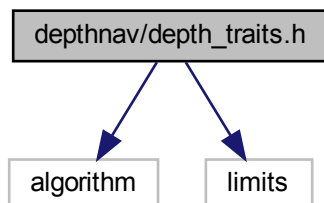
Definition at line 10 of file alpose2d.h.

16.17 depthnav/depth_traits.h File Reference

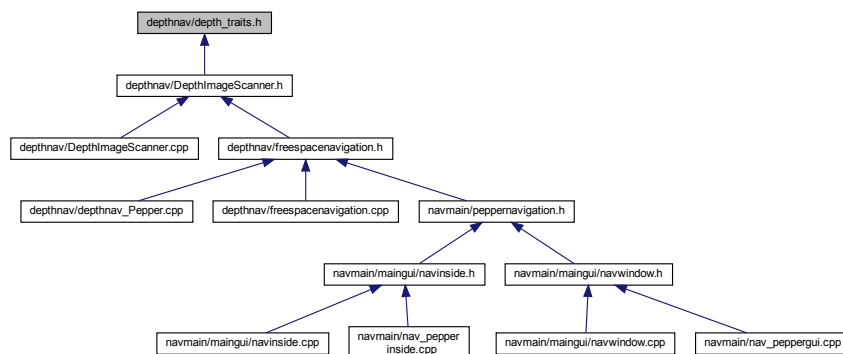
```
#include <algorithm>
```

```
#include <limits>
```

Include dependency graph for depth_traits.h:



This graph shows which files directly or indirectly include this file:



Classes

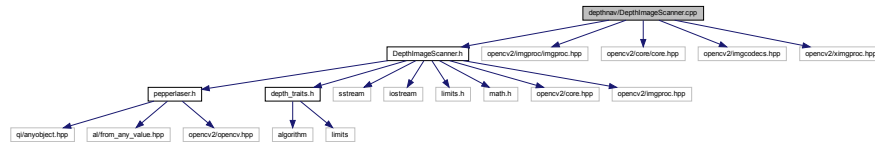
- struct [depthimagescanner::DepthTraits< T >](#)
- struct [depthimagescanner::DepthTraits< uint16_t >](#)
- struct [depthimagescanner::DepthTraits< float >](#)

Namespaces

- [depthimagescanner](#)

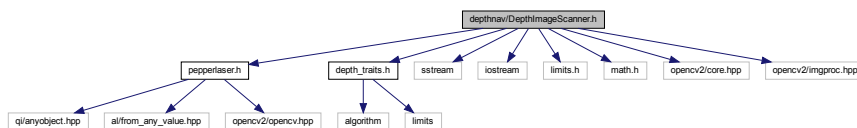
16.18 depthnav/DepthImageScanner.cpp File Reference

```
#include "DepthImageScanner.h"
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/ximgproc.hpp>
Include dependency graph for DepthImageScanner.cpp:
```

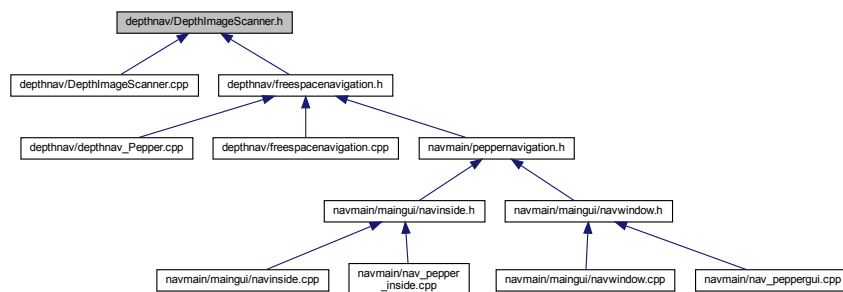


16.19 depthnav/DepthImageScanner.h File Reference

```
#include "pepperlaser.h"
#include "depth_traits.h"
#include <sstream>
#include <iostream>
#include <limits.h>
#include <math.h>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
Include dependency graph for DepthImageScanner.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `depthimagescanner::DepthImageScanner`

16.20.1.2 main()

```
int main (
    int argc,
    char ** argv )
```

Definition at line 175 of file depthnav_Pepper.cpp.

16.20.1.3 processdepth()

```
void processdepth (
    cv::Mat & Id,
    cv::Mat & K )
```

Definition at line 160 of file depthnav_Pepper.cpp.

16.20.1.4 QI_REGISTER_MT_OBJECT()

```
QI_REGISTER_MT_OBJECT (
    MyService ,
    myCallback ,
    myCallback2 ,
    myCallback1 )
```

16.20.2 Variable Documentation

16.20.2.1 eventraised

```
bool eventraised
```

Definition at line 13 of file depthnav_Pepper.cpp.

16.20.2.2 robo

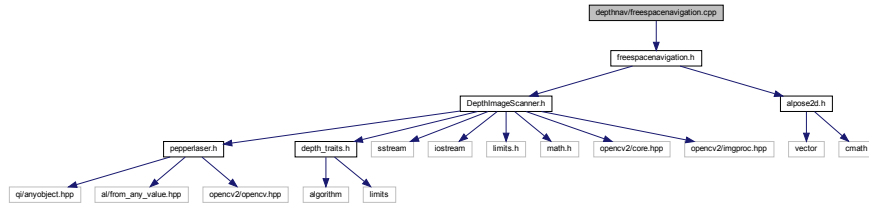
```
vpNaoqiRobot* robo
```

Definition at line 14 of file depthnav_Pepper.cpp.

16.21 depthnav/freespacenavigation.cpp File Reference

```
#include "freespacenavigation.h"
```

Include dependency graph for freespacenavigation.cpp:

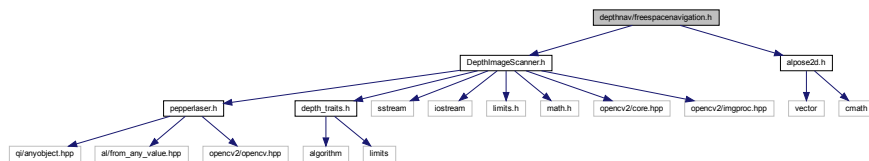


16.22 depthnav/freespacenavigation.h File Reference

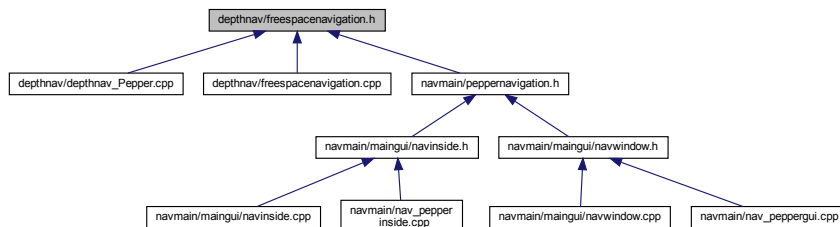
```
#include "DepthImageScanner.h"
```

```
#include "alpose2d.h"
```

Include dependency graph for freespacenavigation.h:



This graph shows which files directly or indirectly include this file:

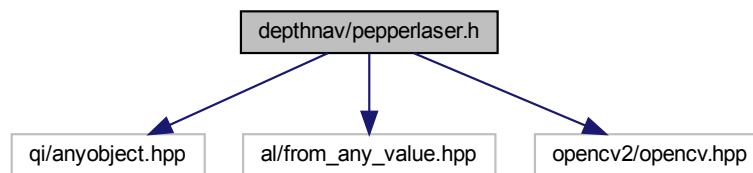


Classes

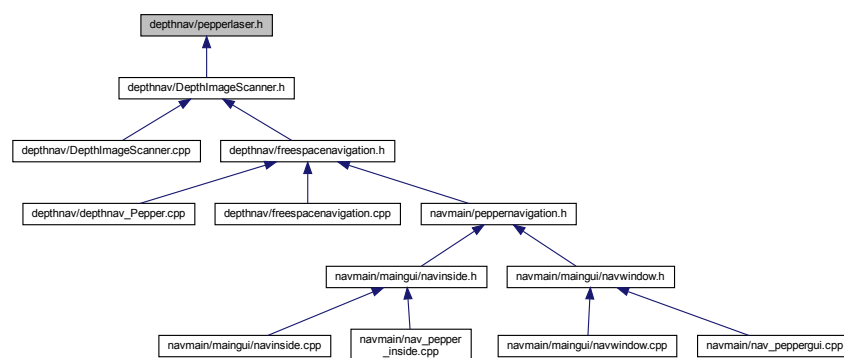
- class [freespacenavigation](#)

16.23 depthnav/pepperlaser.h File Reference

```
#include <qi/anyobject.hpp>
#include <al/from_any_value.hpp>
#include <opencv2/opencv.hpp>
Include dependency graph for pepperlaser.h:
```



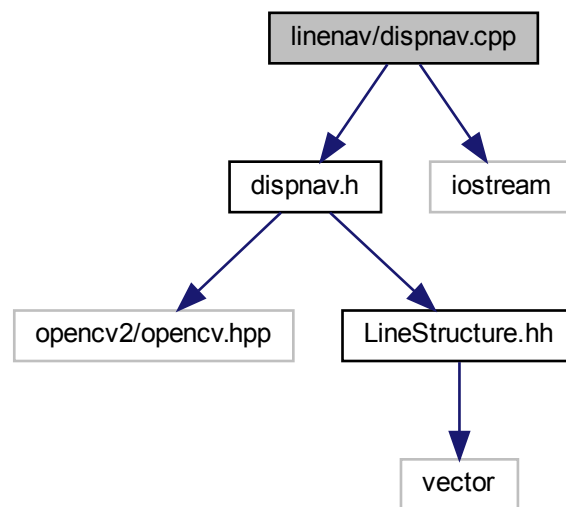
This graph shows which files directly or indirectly include this file:



16.24 linenav/dispnv.cpp File Reference

```
#include "dispnv.h"
#include <iostream>
```

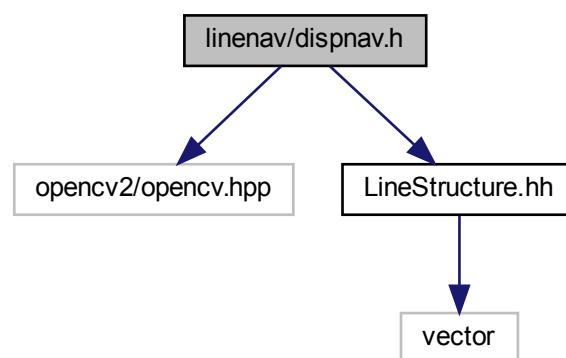
Include dependency graph for dispnav.cpp:



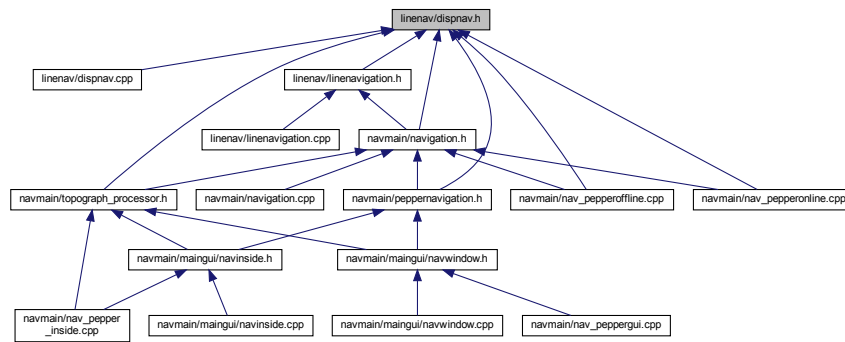
16.25 linenav/dispnav.h File Reference

```
#include <opencv2/opencv.hpp>
#include "LineStructure.hh"
```

Include dependency graph for dispnav.h:



This graph shows which files directly or indirectly include this file:



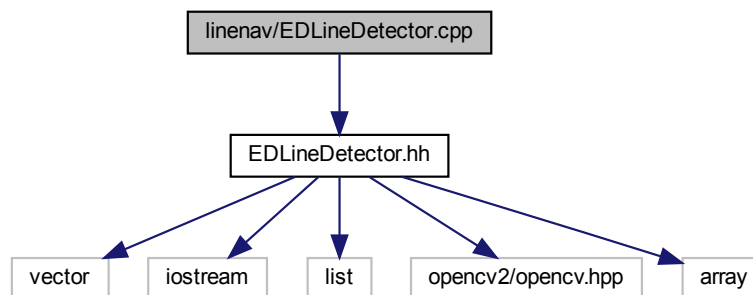
Classes

- class [dispNav](#)
The *dispNav* class.

16.26 linenav/EDLineDetector.cpp File Reference

```
#include "EDLineDetector.hh"
```

Include dependency graph for EDLineDetector.cpp:



Macros

- #define [Horizontal](#) 255
- #define [Vertical](#) 0
- #define [UpDir](#) 1
- #define [RightDir](#) 2
- #define [DownDir](#) 3
- #define [LeftDir](#) 4
- #define [TryTime](#) 6
- #define [SkipEdgePoint](#) 2

Functions

- void `writeMat` (cv::Mat `m`, std::string `name`, int `n`)

16.26.1 Macro Definition Documentation

16.26.1.1 DownDir

```
#define DownDir 3
```

Definition at line 51 of file EDLineDetector.cpp.

16.26.1.2 Horizontal

```
#define Horizontal 255
```

Definition at line 47 of file EDLineDetector.cpp.

16.26.1.3 LeftDir

```
#define LeftDir 4
```

Definition at line 52 of file EDLineDetector.cpp.

16.26.1.4 RightDir

```
#define RightDir 2
```

Definition at line 50 of file EDLineDetector.cpp.

16.26.1.5 SkipEdgePoint

```
#define SkipEdgePoint 2
```

Definition at line 54 of file EDLineDetector.cpp.

16.26.1.6 TryTime

```
#define TryTime 6
```

Definition at line 53 of file EDLineDetector.cpp.

16.26.1.7 UpDir

```
#define UpDir 1
```

Definition at line 49 of file EDLineDetector.cpp.

16.26.1.8 Vertical

```
#define Vertical 0
```

Definition at line 48 of file EDLineDetector.cpp.

16.26.2 Function Documentation

16.26.2.1 writeMat()

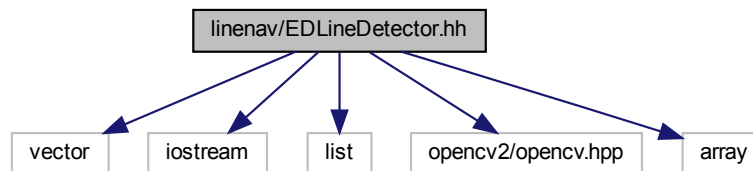
```
void writeMat (
    cv::Mat m,
    std::string name,
    int n )
```

Definition at line 124 of file EDLineDetector.cpp.

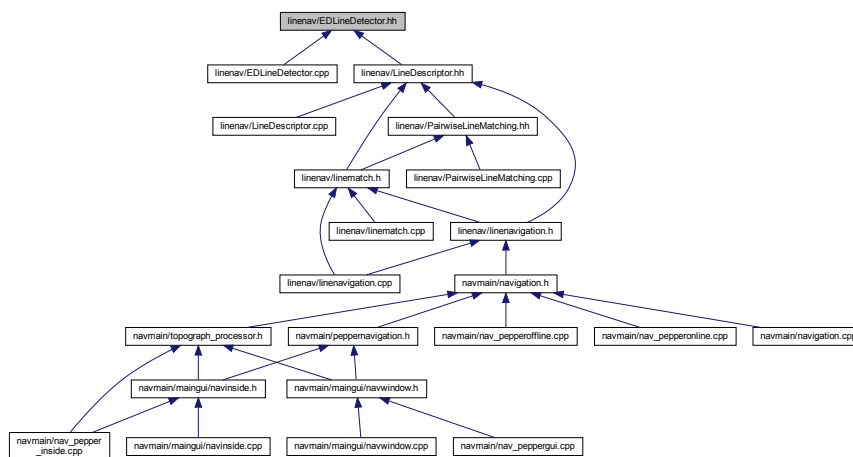
16.27 linenav/EDLineDetector.hh File Reference

```
#include <vector>
#include <iostream>
#include <list>
#include <opencv2/opencv.hpp>
#include <array>
```

Include dependency graph for EDLineDetector.hh:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Pixel](#)
- struct [EdgeChains](#)
- struct [LineChains](#)
- struct [EDLineParam](#)
- class [EDLineDetector](#)

Macros

- #define [RELATIVE_ERROR_FACTOR](#) 100.0
- #define [M_LN10](#) 2.30258509299404568402
- #define [log_gamma](#)(x) ((x)>15.0?log_gamma_windschitt(x):log_gamma_lanczos(x))

Typedefs

- typedef std::list< [Pixel](#) > [PixelChain](#)

16.27.1 Macro Definition Documentation

16.27.1.1 log_gamma

```
#define log_gamma(  
    x ) ((x)>15.0?log_gamma_windschitl(x):log_gamma_lanczos(x))
```

Definition at line 86 of file EDLineDetector.hh.

16.27.1.2 M_LN10

```
#define M_LN10 2.30258509299404568402
```

Definition at line 85 of file EDLineDetector.hh.

16.27.1.3 RELATIVE_ERROR_FACTOR

```
#define RELATIVE_ERROR_FACTOR 100.0
```

Definition at line 84 of file EDLineDetector.hh.

16.27.2 Typedef Documentation

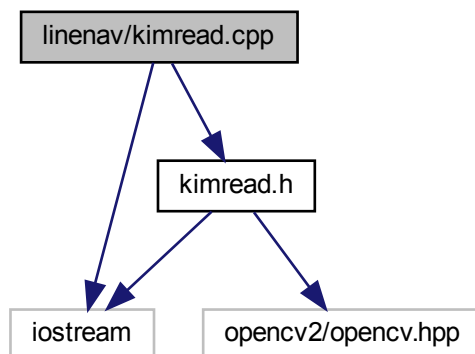
16.27.2.1 PixelChain

```
typedef std::list<Pixel> PixelChain
```

Definition at line 71 of file EDLineDetector.hh.

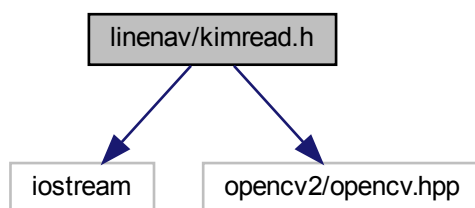
16.28 linenav/kimread.cpp File Reference

```
#include "kimread.h"  
#include <iostream>  
Include dependency graph for kimread.cpp:
```

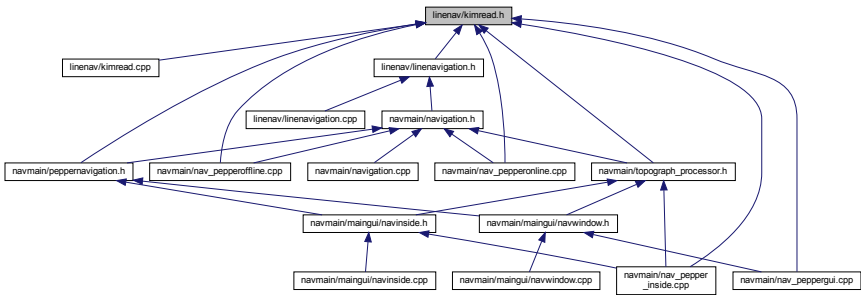


16.29 linenav/kimread.h File Reference

```
#include <iostream>  
#include <opencv2/opencv.hpp>  
Include dependency graph for kimread.h:
```



This graph shows which files directly or indirectly include this file:

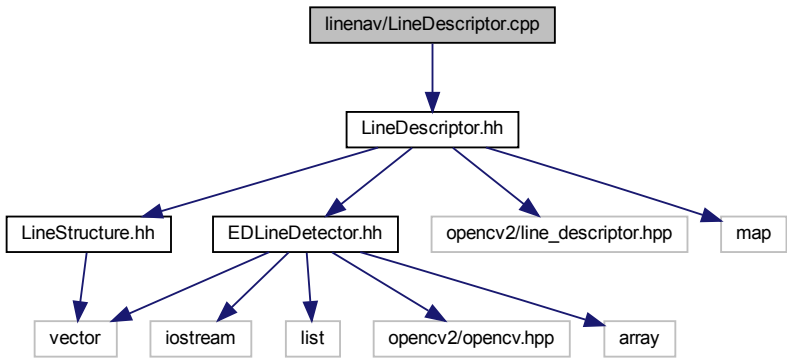


Classes

- class [kimRead](#)

16.30 linenav/LineDescriptor.cpp File Reference

```
#include "LineDescriptor.hh"
Include dependency graph for LineDescriptor.cpp:
```



Macros

- #define [SalienceScale](#) 0.9
- #define [NO_OF_OCTAVES](#) 1

16.30.1 Macro Definition Documentation

16.30.1.1 NO_OF_OCTAVES

```
#define NO_OF_OCTAVES 1
```

Definition at line 15 of file LineDescriptor.cpp.

16.30.1.2 SaliencyScale

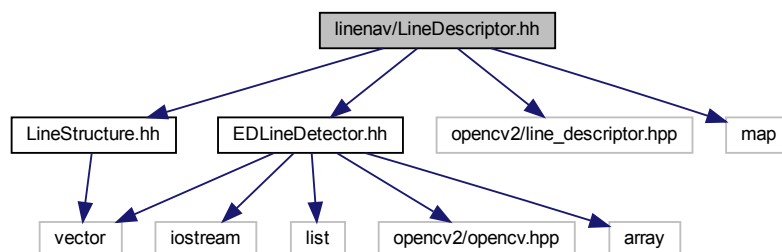
```
#define SaliencyScale 0.9
```

Definition at line 14 of file LineDescriptor.cpp.

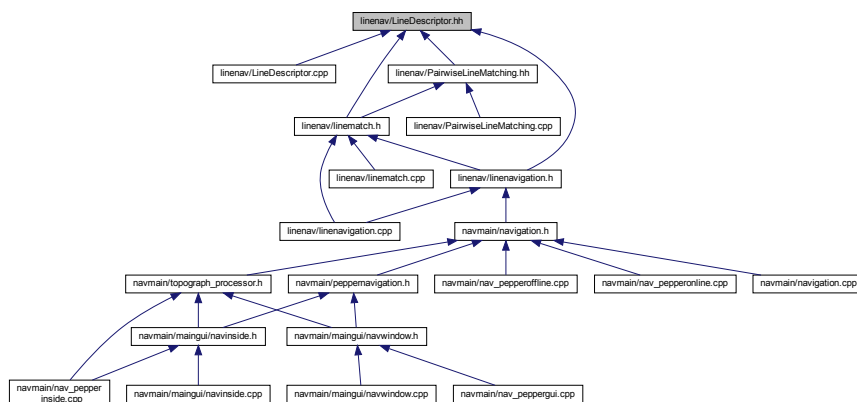
16.31 linenav/LineDescriptor.hh File Reference

```
#include "EDLineDetector.hh"
#include "LineStructure.hh"
#include <opencv2/line_descriptor.hpp>
#include <map>
```

Include dependency graph for LineDescriptor.hh:



This graph shows which files directly or indirectly include this file:



Classes

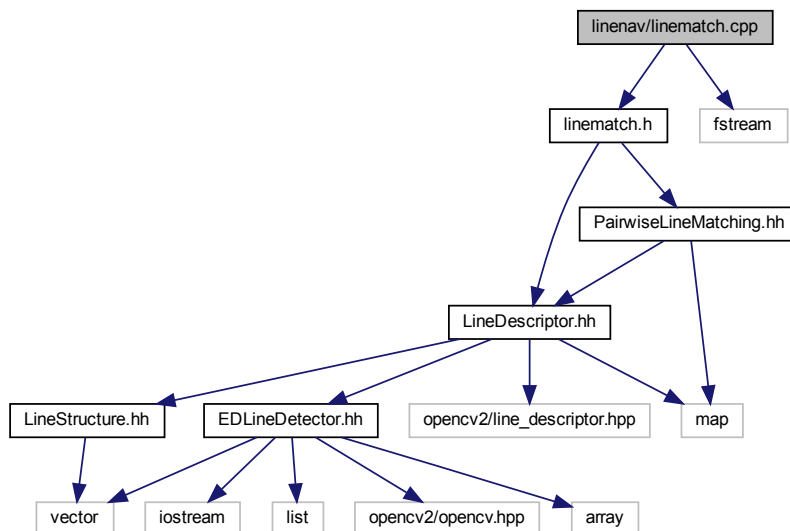
- struct [OctaveLine](#)
- class [LineDescriptor](#)

16.32 linenav/linematch.cpp File Reference

```
#include "linematch.h"
```

```
#include <fstream>
```

Include dependency graph for linematch.cpp:

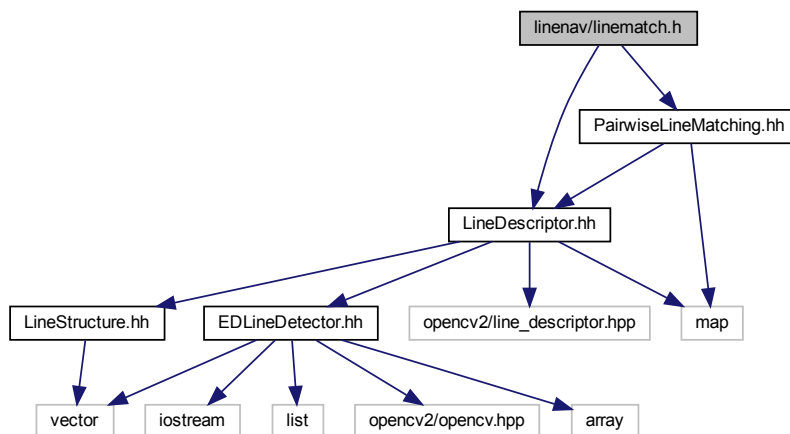


16.33 linenav/linematch.h File Reference

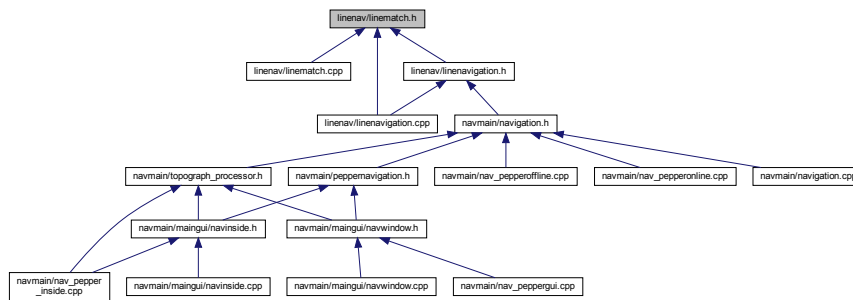
```
#include "LineDescriptor.hh"
```

```
#include "PairwiseLineMatching.hh"
```

Include dependency graph for linematch.h:



This graph shows which files directly or indirectly include this file:



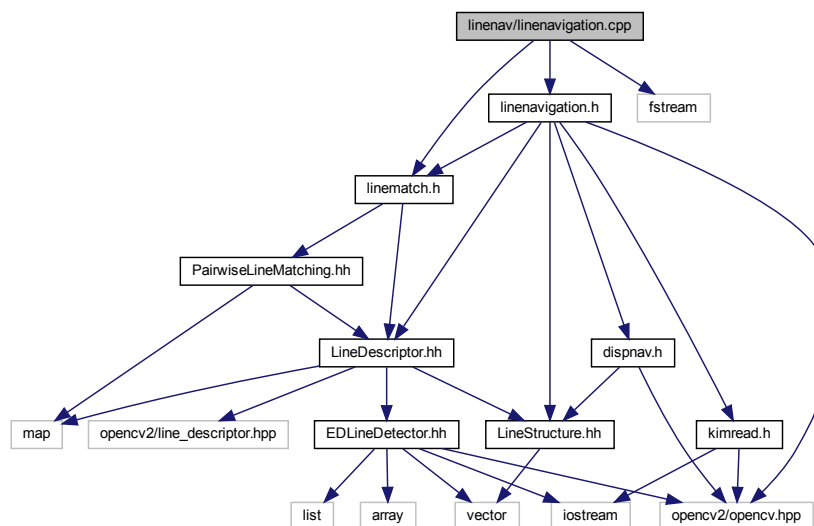
Classes

- class [linematch](#)

16.34 linenav/linenavigation.cpp File Reference

```
#include "linenavigation.h"
#include "linematch.h"
#include <fstream>
```

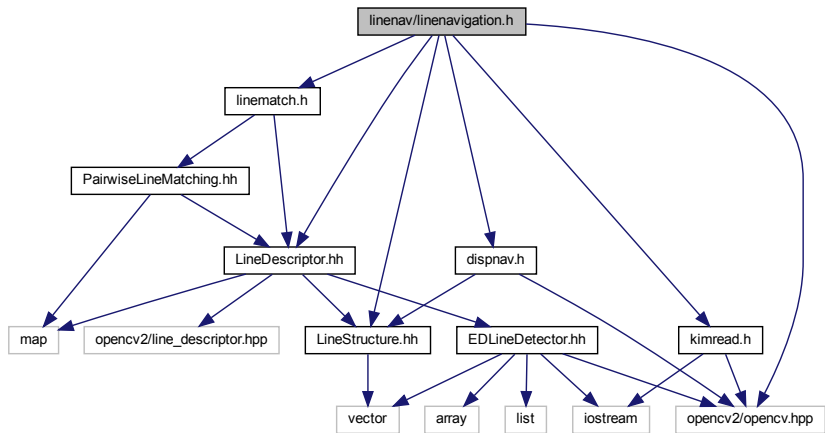
Include dependency graph for linenavigation.cpp:



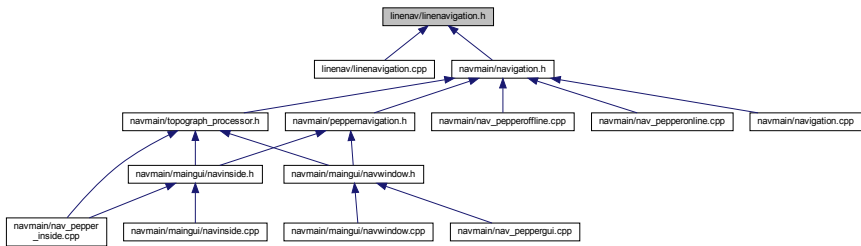
16.35 linenav/linenavigation.h File Reference

```
#include "LineStructure.hh"
#include "LineDescriptor.hh"
#include "linematch.h"
#include <opencv2/opencv.hpp>
```

```
#include "kimread.h"
#include "dispnave.h"
Include dependency graph for linenavigation.h:
```



This graph shows which files directly or indirectly include this file:

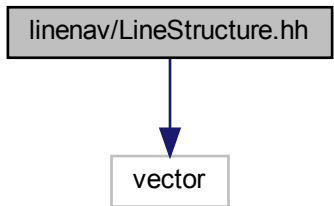


Classes

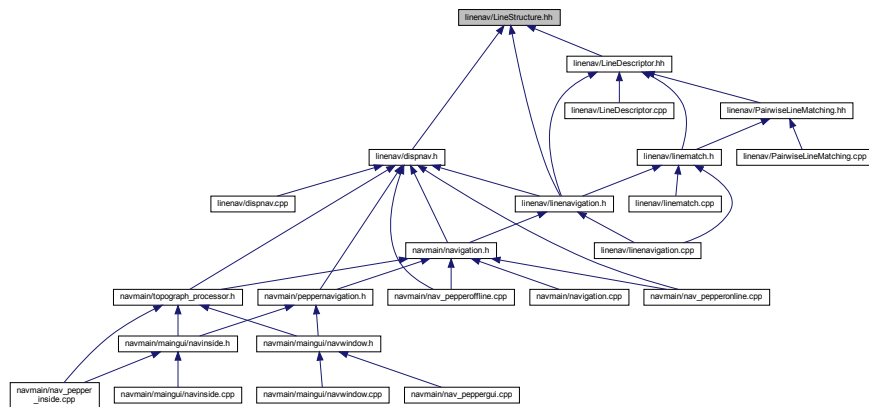
- class [linenavigation](#)

16.36 linenav/LineStructure.hh File Reference

```
#include <vector>
Include dependency graph for LineStructure.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [SingleLine](#)
- struct [OctaveSingleLine](#)

Typedefs

- typedef std::vector< [SingleLine](#) > [Lines_list](#)
- typedef std::vector< [OctaveSingleLine](#) > [LinesVec](#)
- typedef std::vector< [LinesVec](#) > [ScaleLines](#)

16.36.1 Typedef Documentation

16.36.1.1 Lines_list

```
typedef std::vector<SingleLine> Lines\_list
```

Definition at line 46 of file LineStructure.hh.

16.36.1.2 LinesVec

```
typedef std::vector<OctaveSingleLine> LinesVec
```

Definition at line 78 of file LineStructure.hh.

16.36.1.3 ScaleLines

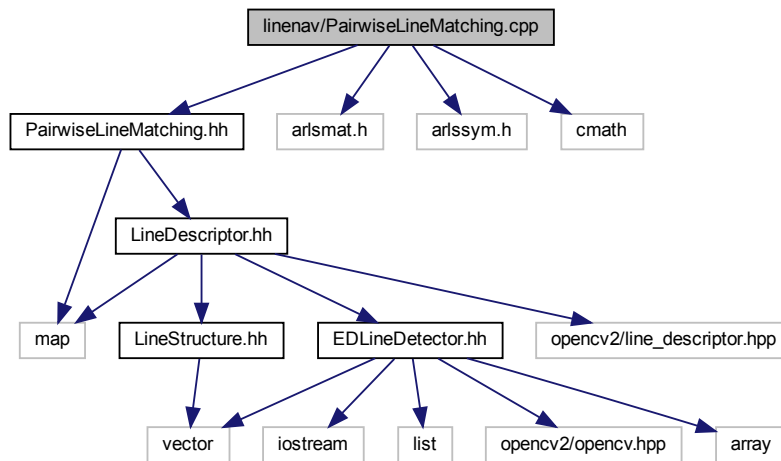
```
typedef std::vector<LinesVec> ScaleLines
```

Definition at line 80 of file LineStructure.hh.

16.37 linenav/PairwiseLineMatching.cpp File Reference

```
#include "PairwiseLineMatching.hh"
#include "arlsmat.h"
#include "arlssym.h"
#include <cmath>
```

Include dependency graph for PairwiseLineMatching.cpp:



Macros

- `#define` `Inf` 1e10
- `#define` `ResolutionScale` 20
- `#define` `AcceptableAngleHistogramDifference` 0.49
- `#define` `AcceptableLengthVectorDifference` 0.4
- `#define` `LengthDifThreshold` 4
- `#define` `AngleDifferenceThreshold` 0.7854
- `#define` `DescriptorDifThreshold` 0.35
- `#define` `RelativeAngleDifferenceThreshold` 0.7854
- `#define` `IntersectionRationDifThreshold` 1
- `#define` `ProjectionRationDifThreshold` 1
- `#define` `WeightOfMeanEigenVec` 0.1

Functions

- void `normalize` (std::vector< double > &vect)
- double `normL2` (std::vector< double > &vect)

16.37.1 Macro Definition Documentation

16.37.1.1 AcceptableAngleHistogramDifference

```
#define AcceptableAngleHistogramDifference 0.49
```

Definition at line 21 of file PairwiseLineMatching.cpp.

16.37.1.2 AcceptableLengthVectorDifference

```
#define AcceptableLengthVectorDifference 0.4
```

Definition at line 22 of file PairwiseLineMatching.cpp.

16.37.1.3 AngleDifferenceThreshold

```
#define AngleDifferenceThreshold 0.7854
```

Definition at line 29 of file PairwiseLineMatching.cpp.

16.37.1.4 DescriptorDifThreshold

```
#define DescriptorDifThreshold 0.35
```

Definition at line 30 of file PairwiseLineMatching.cpp.

16.37.1.5 Inf

```
#define Inf 1e10
```

Definition at line 15 of file PairwiseLineMatching.cpp.

16.37.1.6 IntersectionRationDifThreshold

```
#define IntersectionRationDifThreshold 1
```

Definition at line 35 of file PairwiseLineMatching.cpp.

16.37.1.7 LengthDifThreshold

```
#define LengthDifThreshold 4
```

Definition at line 28 of file PairwiseLineMatching.cpp.

16.37.1.8 ProjectionRationDifThreshold

```
#define ProjectionRationDifThreshold 1
```

Definition at line 36 of file PairwiseLineMatching.cpp.

16.37.1.9 RelativeAngleDifferenceThreshold

```
#define RelativeAngleDifferenceThreshold 0.7854
```

Definition at line 34 of file PairwiseLineMatching.cpp.

16.37.1.10 ResolutionScale

```
#define ResolutionScale 20
```

Definition at line 17 of file PairwiseLineMatching.cpp.

16.37.1.11 WeightOfMeanEigenVec

```
#define WeightOfMeanEigenVec 0.1
```

Definition at line 40 of file PairwiseLineMatching.cpp.

16.37.2 Function Documentation

16.37.2.1 normalize()

```
void normalize (  
    std::vector< double > & vect ) [inline]
```

Definition at line 41 of file PairwiseLineMatching.cpp.

Typedefs

- typedef std::vector< [Node](#) > [Nodes_list](#)
- typedef std::multimap< double, unsigned int, [CompareL](#) > [EigenMAP](#)
- typedef std::multimap< double, unsigned int, [CompareS](#) > [DISMAP](#)

16.38.1 Typedef Documentation

16.38.1.1 DISMAP

```
typedef std::multimap<double,unsigned int,CompareS> DISMAP
```

Definition at line 35 of file PairwiseLineMatching.hh.

16.38.1.2 EigenMAP

```
typedef std::multimap<double,unsigned int,CompareL> EigenMAP
```

Definition at line 30 of file PairwiseLineMatching.hh.

16.38.1.3 Nodes_list

```
typedef std::vector<Node> Nodes\_list
```

Definition at line 24 of file PairwiseLineMatching.hh.

16.39 mapping/generate_configfile.m File Reference

Functions

- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL [DAMAGES](#) (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF % *SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS % *INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF [LIABILITY](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT OR [TORT](#) (INCLUDING NEGLIGENCE OR OTHERWISE) % *ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE
- % % % add line matching lib path [addpath](#) ('./linematching')
- [fprintf](#) (fileID,'% s % 2s\n', [nodesname](#))
- [fprintf](#) (fileID,'% s % s % s\n', [edges](#))
- [fprintf](#) (fileID,'% s \n', [hd3](#))

- `if (s < 7 && d == 'c') fold`
- `elseif (s < 7 && d == 'a') fold = ['Edge_', num2str(e), num2str(e-1)]`
- `if (dr == 's') kim = indd + 1`
- `elseif (dr == 'e') kim`
- `if (l == 1) flagx = 0`
- `elseif ((l == L) || (linect(l-1) > linect(l+1))) flagx`
- `elseif (linect(l-1) <= linect(l+1)) flagx = 0`
- `end if (flagx == 1) indd`
- `elseif (flagx == 0) indd`
- `end matches (i)`
- `end branch (:, end)`
- `fprintf (fileID, '% s % s % s % s\n', branch')`
- `system ('rm matched.lines')`

Variables

- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED WARRANTIES
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED INCLUDING
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED TO
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY DIRECT
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY INDIRECT
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY INCIDENTAL
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY SPECIAL
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY EXEMPLARY
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % * CONTRACT
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT LIABILITY
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT OR EVEN IF ADVISED OF THE % *POSSIBILITY OF SUCH DAMAGE % % clc
- `close all`
- % % % base folder where topo graph lies bf
- % % % hd1

- [Harvey](#)
- [Rob1](#)
- [Peter](#)
- [ROb2](#)
- [Cartman](#)
- [Belinda](#)
- [Manipulation](#)
- [Robotics](#)
- `nodes = 0:length(nodesname)-1`
- `nodesname`
- `% % hd2`
- `edges`
- `hd3`
- `edgelist = []`
- `for i`
- `d = bb(end)`
- `e = str2double(bb(end-1))`
- `s = str2double(bb(end-2))`
- `sf = fold`
- `df = bb`
- `dr = 's'`
- `end bb`
- `linect=[]`
- `kl = kimlists`
- `end L = length(kl)`
- `for j`
- `status`
- `end [M I] = max(linect)`
- `flagx = -1`

16.39.1 Function Documentation

16.39.1.1 `addpath()`

```
% % % add line matching lib path addpath (
    './linematching' )
```

16.39.1.2 `branch()`

```
end branch (
    : ,
    end )
```

16.39.1.3 DAMAGES()

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL DAMAGES (
    INCLUDING ,
    BUT NOT LIMITED TO,
    PROCUREMENT OF % *SUBSTITUTE GOODS OR SERVICES;LOSS OF USE,
    DATA ,
    OR PROFITS;OR BUSINESS % * INTERRUPTION )
```

16.39.1.4 elseif() [1/5]

```
elseif (
    (I==L)|| (linect(I-1) > linect(I+1)) )
```

16.39.1.5 elseif() [2/5]

```
elseif (
    dr == 'e' )
```

16.39.1.6 elseif() [3/5]

```
elseif (
    flagx == 0 )
```

16.39.1.7 elseif() [4/5]

```
elseif (
    linect(I-1)<=linect(I+1) ) [pure virtual]
```

16.39.1.8 elseif() [5/5]

```
elseif ( ) = ['Edge_',num2str(e),num2str(e-1)]
```

16.39.1.9 fprintf() [1/4]

```
fprintf (
    fileID ,
    '% s % 2s\n' ,
    nodesname'  )
```

16.39.1.10 fprintf() [2/4]

```
fprintf (
    fileID ,
    '% s % s % s % s\n' ,
    branch'  )
```

16.39.1.11 fprintf() [3/4]

```
fprintf (
    fileID ,
    '% s % s % s\n' ,
    edges'  )
```

16.39.1.12 fprintf() [4/4]

```
fprintf (
    fileID ,
    '% s \n' ,
    hd3  )
```

16.39.1.13 if() [1/4]

```
end if (
    dr  == 's' ) = indd+1
```

16.39.1.14 if() [2/4]

```
end if (
    flagx  = =1 )
```

16.39.1.15 if() [3/4]

```
if (
    I  = =I ) [pure virtual]
```

16.39.1.16 if() [4/4]

```
if ( )
```

16.39.1.17 matches()

```
end matches (
    i )
```

16.39.1.18 system()

```
system (
    'rm matched.lines' )
```

16.39.1.19 TORT()

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL WHETHER IN % STRICT OR TORT (
    INCLUDING NEGLIGENCE OR OTHERWISE )
```

16.39.2 Variable Documentation**16.39.2.1 all**

```
close all
```

Definition at line 23 of file generate_configfile.m.

16.39.2.2 bb

```
end bb
```

Initial value:

```
= [sf,' ',df,' ', dr]
```

```
[imlist kimlists] = getImages([bf sf])
```

Definition at line 129 of file generate_configfile.m.

16.39.2.3 Belinda

```
Belinda
```

Definition at line 43 of file generate_configfile.m.

16.39.2.4 bf

```
% % % base filder where topo grapg lies bf
```

Initial value:

```
= '/home/suman/soft/pepper_navigation/data/tmap/'  
fileID = fopen([bf,'conf.txt'],'w')
```

Definition at line 31 of file generate_configfile.m.

16.39.2.5 Cartman

```
Cartman
```

Definition at line 42 of file generate_configfile.m.

16.39.2.6 clc

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR  
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵  
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*  
CONSEQUENTIAL WHETHER IN % STRICT OR EVEN IF ADVISED OF THE %* POSSIBILITY OF SUCH DAMAGE %* % clc
```

Definition at line 22 of file generate_configfile.m.

16.39.2.7 CONTRACT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR←
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL WHETHER IN %* CONTRACT
```

Definition at line 10 of file generate_configfile.m.

16.39.2.8 d

```
d = bb(end)
```

Definition at line 102 of file generate_configfile.m.

16.39.2.9 df

```
df = bb
```

Definition at line 109 of file generate_configfile.m.

16.39.2.10 DIRECT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR←
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY DIRE←
CT
```

Definition at line 6 of file generate_configfile.m.

16.39.2.11 dr

```
dr = 's'
```

Definition at line 110 of file generate_configfile.m.

16.39.2.12 e

```
e = str2double(bb(end-1))
```

Definition at line 103 of file generate_configfile.m.

16.39.2.13 edgelist

```
edgelist = []
```

Definition at line 96 of file generate_configfile.m.

16.39.2.14 edges

```
edges
```

Initial value:

```
= [
"Edge_01" "0" "1"
"Edge_12" "1" "2"
"Edge_23" "2" "3"
"Edge_34" "3" "4"
"Edge_45" "4" "5"
"Edge_56" "5" "6"
"Edge_60" "6" "0"
"Edge_10" "1" "0"
"Edge_21" "2" "1"
"Edge_32" "3" "2"
"Edge_43" "4" "3"
"Edge_54" "5" "4"
"Edge_65" "6" "5"
"Edge_06" "0" "6"
]
```

Definition at line 56 of file generate_configfile.m.

16.39.2.15 end

```
end[M I] = max(linect)
```

Definition at line 155 of file generate_configfile.m.

16.39.2.16 EXEMPLARY

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY EXEM↵
PLARY
```

Definition at line 6 of file generate_configfile.m.

16.39.2.17 flagx

```
flagx = -1
```

Definition at line 156 of file generate_configfile.m.

16.39.2.18 Harvey

Harvey

Definition at line 38 of file generate_configfile.m.

16.39.2.19 hd1

```
% % hd1
```

Initial value:

```
= "Nodes"
nodesname =
```

Definition at line 35 of file generate_configfile.m.

16.39.2.20 hd2

```
% % hd2
```

Initial value:

```
= "Edges"
fprintf(fileID, '%s\n', hd2)
```

Definition at line 54 of file generate_configfile.m.

16.39.2.21 hd3

hd3

Initial value:

```
= "Branches"
branch = [
"Branch_17c" "1" "7" "18"
"Branch_75c" "7" "5" "4"
"Branch_28c" "2" "8" "11"
"Branch_84c" "8" "4" "3"
"Branch_57c" "5" "7" "9"
"Branch_71c" "7" "1" "3"
"Branch_48c" "4" "8" "17"
"Branch_82c" "8" "2" "4"
"Branch_17a" "1" "7" "10"
"Branch_75a" "7" "5" "3"
"Branch_28a" "2" "8" "15"
"Branch_84a" "8" "4" "1"
"Branch_57a" "5" "7" "21"
"Branch_71a" "7" "1" "0"
"Branch_48a" "4" "8" "11"
"Branch_82a" "8" "2" "4"
]
```

Definition at line 73 of file generate_configfile.m.

16.39.2.22 i

```
for i
```

Initial value:

```
=1:length(branch)

    bb=sscanf(branch(i,1),'%c')
```

Definition at line 99 of file generate_configfile.m.

16.39.2.23 INCIDENTAL

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY INCI↵
DENTAL
```

Definition at line 6 of file generate_configfile.m.

16.39.2.24 INCLUDING

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED INCLUDING
```

Definition at line 3 of file generate_configfile.m.

16.39.2.25 INDIRECT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY INDI↵
RECT
```

Definition at line 6 of file generate_configfile.m.

16.39.2.26 j

```
for j
```

Initial value:

```
=1:L

    kim2 = k1{j}
```

Definition at line 148 of file generate_configfile.m.

16.39.2.27 kl

```
kl = kimlists
```

Definition at line 139 of file generate_configfile.m.

16.39.2.28 L

```
end L = length(kl)
```

Definition at line 147 of file generate_configfile.m.

16.39.2.29 LIABILITY

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR  
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-  
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*  
CONSEQUENTIAL WHETHER IN % STRICT LIABILITY
```

Definition at line 10 of file generate_configfile.m.

16.39.2.30 linect

```
linect =[]
```

Definition at line 135 of file generate_configfile.m.

16.39.2.31 Manipulation

```
Manipulation
```

Definition at line 44 of file generate_configfile.m.

16.39.2.32 nodes

```
nodes = 0:length(nodesname)-1
```

Definition at line 48 of file generate_configfile.m.

16.39.2.33 nodesname

nodesname

Initial value:

```
= [nodesname, nodes']  
fprintf(fileID, '%s\n', hd1)
```

Definition at line 49 of file generate_configfile.m.

16.39.2.34 Peter

Peter

Definition at line 40 of file generate_configfile.m.

16.39.2.35 Rob1

Rob1

Definition at line 39 of file generate_configfile.m.

16.39.2.36 ROb2

ROb2

Definition at line 41 of file generate_configfile.m.

16.39.2.37 Robotics

Robotics

Definition at line 46 of file generate_configfile.m.

16.39.2.38 s

```
s = str2double(bb(end-2))
```

Definition at line 104 of file generate_configfile.m.

16.39.2.39 sf

```
sf = fold
```

Definition at line 108 of file generate_configfile.m.

16.39.2.40 SPECIAL

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY SPEC↵
IAL
```

Definition at line 6 of file generate_configfile.m.

16.39.2.41 status

```
end match lines if status
```

Initial value:

```
= matchLines(kim,kim2,'matched.lines')
    matchindex = load('matched.lines')
```

Definition at line 150 of file generate_configfile.m.

16.39.2.42 TO

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED TO
```

Definition at line 3 of file generate_configfile.m.

16.39.2.43 WARRANTIES

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED WARRANTIES
```

Definition at line 3 of file generate_configfile.m.

16.40 mapping/getImages.m File Reference**Variables**

- % % By suman raj bista % function to get name of imagesfiles from [folder function](#) [ListOfImageNames TextFileList]

16.40.1 Variable Documentation

16.40.1.1 function

```
% % By suman raj bista % function to get name of imagesfiles from folder function[ListOfImageNames
TextFileList]
```

Initial value:

```
= getImages(folder)
%
global fodseq
```

Definition at line 4 of file getImages.m.

16.41 mapping/select_ReferencelImages.m File Reference

Functions

- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF % *SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS % *INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) % *ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE
- % % % set path for linematching and trifocal tensor estimation addpath('./tftLine')
- addpath('./linematching')
- if (size(gim, 3) == 3) gim
- imwrite(gim, 'tmpimg.pgm')
- save('lines_l.tmp', 'lines', '-ascii', '-tabs')
- system(['cp ', im, ' ', KFD])
- % save([KFD, '/kl_', im(end-7:end-3), 'txt'], 'lines', '-ascii', '-tabs')
- dlmwrite([KFD, '/kl_', im(end-8:end-3), 'txt'], lines, 'delimiter', '\t', 'precision', '%.12f')
- disp('New Key Image Added')
- dlmwrite('lines_r.tmp', lines, 'delimiter', '\t', 'precision', '%.12f')
- if length(matchindex) < 10 % status
- if (sfc > 2) ct=0
- % save('lines_t.tmp', 'linesprev', '-ascii', '-tabs')
- dlmwrite('lines_l.tmp', linesprev, 'delimiter', '\t', 'precision', '%.12f')
- if (length(validindex) < thres || (cr < 0.5 && pr < 0.5)) ct=0

Variables

- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED [WARRANTIES](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED [INCLUDING](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED [TO](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY [DIRECT](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY [INDIRECT](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY [INCIDENTAL](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY [SPECIAL](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY [EXEMPLARY](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % * [CONTRACT](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT [LIABILITY](#)
- % % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS % *AND ANY EXPRESS OR IMPLIED BUT NOT LIMITED THE % *IMPLIED [WARRANTIES](#) OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE % *ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE % *LIABLE FOR ANY OR % *CONSEQUENTIAL WHETHER IN % STRICT OR EVEN IF ADVISED OF THE % *POSSIBILITY OF SUCH DAMAGE % % clear [all](#)
- [clc](#)
- % % % [folder](#) that contains [all](#) the images in png % path of basefolder % % [bf](#) = '../roboroom'
- % % threshold no of minimum match [lines](#) [thres](#) = 20
- % % % image seq is inside imgs [folder](#) [\[imList\]](#) =getImages([\[bf](#) '/imgs_acquired'])
- % key reference images will be stored in kfls [folder](#) [KFD](#) = [\[bf](#) '/ref_imgs']
- % % [mp](#) = []
- [matchedlines](#) = []
- [indxmatch](#) =[]
- [aaa](#) =[]
- [lc](#) = []
- [ct](#) = 0
- global [imSize](#) = [640 480]
- [sfc](#) = 0
- % % for [j](#)
- % % detect [lines](#) [status](#) = detectLines('tmpimg.pgm','edlines.out')
- [lines](#) = load('edlines.out')
- [gim_t](#) = gim
- [im_t](#) = im

- `linest = lines`
- `linesmatch = []`
- `pr = 1`
- `matchindex = load('matched.lines')`
- `linesprev = linest(matchindex(:,1),:)`
- `% if view line m`
- `% if view lineatching fialed Add new ref image continue`
- `end % inliers = validindex<0.1`
- `cr = sum(inliers)/length(validindex)`

16.41.1 Function Documentation

16.41.1.1 `addpath()` [1/2]

```
addpath (
    './linematching' )
```

16.41.1.2 `addpath()` [2/2]

```
% % % set path for linematching and trifocal tensor estimation addpath (
    './tftLine' )
```

16.41.1.3 `DAMAGES()`

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↔
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL DAMAGES (
    INCLUDING ,
    BUT NOT LIMITED TO,
    PROCUREMENT OF % *SUBSTITUTE GOODS OR SERVICES;LOSS OF USE,
    DATA ,
    OR PROFITS;OR BUSINESS % * INTERRUPTION )
```

16.41.1.4 `disp()`

```
disp (
    'New Key Image Added' )
```

16.41.1.5 dlmwrite() [1/3]

```
dlmwrite (
    'lines_l.tmp' ,
    linesprev ,
    'delimiter' ,
    '\t' ,
    'precision' ,
    '%.12f' )
```

16.41.1.6 dlmwrite() [2/3]

```
dlmwrite (
    'lines_r.tmp' ,
    lines ,
    'delimiter' ,
    '\t' ,
    'precision' ,
    '%.12f' )
```

16.41.1.7 dlmwrite() [3/3]

```
dlmwrite (
    lines ,
    'delimiter' ,
    '\t' ,
    'precision' ,
    '%.12f' )
```

16.41.1.8 if() [1/3]

```
if ( ) [pure virtual]
```

16.41.1.9 if() [2/3]

```
if (
    sfc ,
    2 ) [pure virtual]
```

16.41.1.10 if() [3/3]

```
if (
    size(gim, 3) == 3 )
```

16.41.1.11 imwrite()

```
imwrite (
    gim ,
    'tmpimg.pgm' )
```

16.41.1.12 length()

```
if length (
    matchindex )
```

16.41.1.13 save() [1/3]

```
save (
    'lines_l.tmp' ,
    'lines' ,
    '-ascii' ,
    '-tabs' )
```

16.41.1.14 save() [2/3]

```
% save (
    'lines_t.tmp' ,
    'linesprev' ,
    '-ascii' ,
    '-tabs' )
```

16.41.1.15 save() [3/3]

```
else save (
    'lines' ,
    '-ascii' ,
    '-tabs' )
```

16.41.1.16 system()

```
system ( )
```

16.41.1.17 TORT()

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR←
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL WHETHER IN % STRICT OR TORT (
    INCLUDING NEGLIGENCE OR OTHERWISE )
```

16.41.2 Variable Documentation

16.41.2.1 aaa

```
aaa =[]
```

Definition at line 52 of file select_ReferenceImages.m.

16.41.2.2 all

```
close all
```

Definition at line 24 of file select_ReferenceImages.m.

16.41.2.3 bf

```
% % % folder that contains all the images in png % path of basefolder % % bf = '../roboroom'
```

Definition at line 36 of file select_ReferenceImages.m.

16.41.2.4 clc

```
clc
```

Definition at line 26 of file select_ReferenceImages.m.

16.41.2.5 continue

```
% if view line atching fialed Add new ref image continue
```

Definition at line 130 of file select_ReferenceImages.m.

16.41.2.6 CONTRACT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*
CONSEQUENTIAL WHETHER IN %* CONTRACT
```

Definition at line 10 of file select_ReferencelImages.m.

16.41.2.7 cr

```
cr = sum(inliers)/length(validindex)
```

Definition at line 135 of file select_ReferencelImages.m.

16.41.2.8 ct

```
catch ct = 0
```

Definition at line 54 of file select_ReferencelImages.m.

16.41.2.9 DIRECT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY DIRE↵
CT
```

Definition at line 6 of file select_ReferencelImages.m.

16.41.2.10 EXEMPLARY

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY EXEM↵
PLARY
```

Definition at line 6 of file select_ReferencelImages.m.

16.41.2.11 folder

```
% % % image seq is inside imgs folder[imList] =getImages([bf '/imgs_acquired'])
```

Definition at line 42 of file select_ReferenceImages.m.

16.41.2.12 gim_t

```
gim_t = gim
```

Definition at line 80 of file select_ReferenceImages.m.

16.41.2.13 im_t

```
im_t = im
```

Definition at line 81 of file select_ReferenceImages.m.

16.41.2.14 imSize

```
end imSize = [640 480]
```

Definition at line 55 of file select_ReferenceImages.m.

16.41.2.15 INCIDENTAL

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR↵
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY INCI↵
DENTAL
```

Definition at line 6 of file select_ReferenceImages.m.

16.41.2.16 INCLUDING

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED INCLUDING
```

Definition at line 3 of file select_ReferenceImages.m.

16.41.2.17 INDIRECT

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY INDI-
RECT
```

Definition at line 6 of file select_ReferencelImages.m.

16.41.2.18 indxmatch

```
indxmatch =[]
```

Definition at line 51 of file select_ReferencelImages.m.

16.41.2.19 inliers

```
end % inliers = validindex<0.1
```

Definition at line 133 of file select_ReferencelImages.m.

16.41.2.20 j

```
% % for j
```

Initial value:

```
=1:length(imList)
    %% read images
    im = imList{j}
    gim = imread(imList{j})
```

Definition at line 61 of file select_ReferencelImages.m.

16.41.2.21 KFD

```
% key reference images will be stored in kfls folder KFD = [bf '/ref_imgs']
```

Definition at line 45 of file select_ReferencelImages.m.

16.41.2.22 lc

```
lc = []
```

Definition at line 53 of file select_ReferencelImages.m.

16.41.2.23 LIABILITY

```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR  
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-  
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY OR %*  
CONSEQUENTIAL WHETHER IN % STRICT LIABILITY
```

Definition at line 10 of file select_ReferenceImages.m.

16.41.2.24 lines

```
lines = load('edlines.out')
```

Definition at line 73 of file select_ReferenceImages.m.

16.41.2.25 linesmatch

```
linesmatch =[]
```

Definition at line 87 of file select_ReferenceImages.m.

16.41.2.26 linesprev

```
linesprev = linest(matchindex(:,1),:)
```

Definition at line 118 of file select_ReferenceImages.m.

16.41.2.27 linest

```
linest = lines
```

Definition at line 82 of file select_ReferenceImages.m.

16.41.2.28 m

```
% if view line m
```

Definition at line 129 of file select_ReferenceImages.m.

16.41.2.29 matchedlines

```
if ct try matchedlines = []
```

Definition at line 50 of file select_ReferencelImages.m.

16.41.2.30 matchindex

```
matchindex = load('matched.lines')
```

Definition at line 98 of file select_ReferencelImages.m.

16.41.2.31 mp

```
% % mp = []
```

Definition at line 49 of file select_ReferencelImages.m.

16.41.2.32 pr

```
pr = 1
```

Definition at line 90 of file select_ReferencelImages.m.

16.41.2.33 sfc

```
end else sfc = 0
```

Definition at line 57 of file select_ReferencelImages.m.

16.41.2.34 SPECIAL

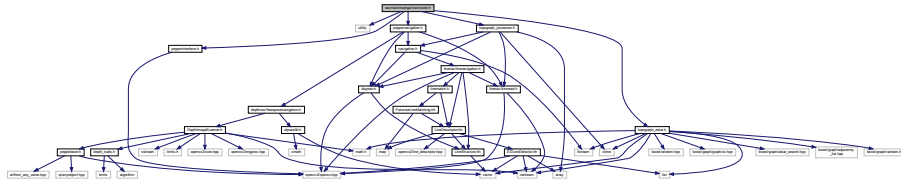
```
% % % THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS %* AND ANY EXPRESS OR  
IMPLIED BUT NOT LIMITED THE %* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR←  
POSE %* ARE DISCLAIMED IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE %* LIABLE FOR ANY SPEC←  
IAL
```

Definition at line 6 of file select_ReferencelImages.m.

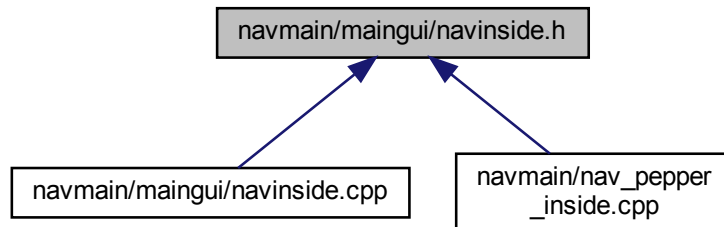
16.43 navmain/maingui/navinside.h File Reference

```
#include <utility>
#include "pepperInterface.h"
#include "peppernavigation.h"
#include "topograph_astar.h"
#include "topograph_processor.h"
```

Include dependency graph for navinside.h:



This graph shows which files directly or indirectly include this file:



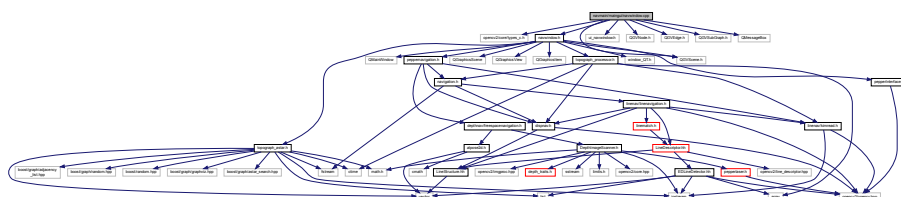
Classes

- class [NavInside](#)

16.44 navmain/maingui/navwindow.cpp File Reference

```
#include <opencv2/core/types_c.h>
#include "navwindow.h"
#include "ui_navwindow.h"
#include "QGVScene.h"
#include "QGVNode.h"
#include "QGVEdge.h"
#include "QGVSubGraph.h"
#include <QMessageBox>
```

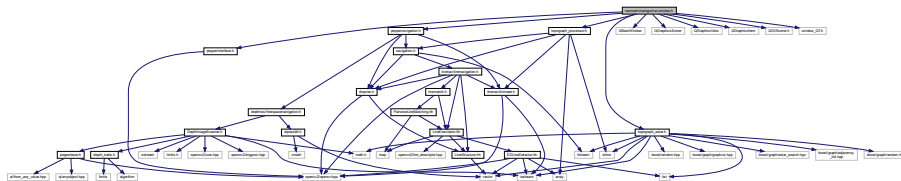
Include dependency graph for navwindow.cpp:



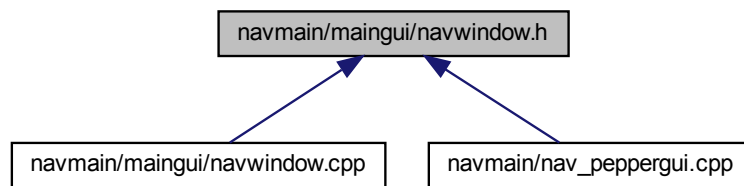
16.45 navmain/maingui/navwindow.h File Reference

```
#include "pepperInterface.h"
#include "peppernavigation.h"
#include <QMainWindow>
#include "topograph_astar.h"
#include <QGraphicsScene>
#include <QGraphicsView>
#include <QGraphicsItem>
#include "QGVScene.h"
#include "topograph_processor.h"
#include "window_QT.h"
```

Include dependency graph for navwindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [NavWindow](#)

Namespaces

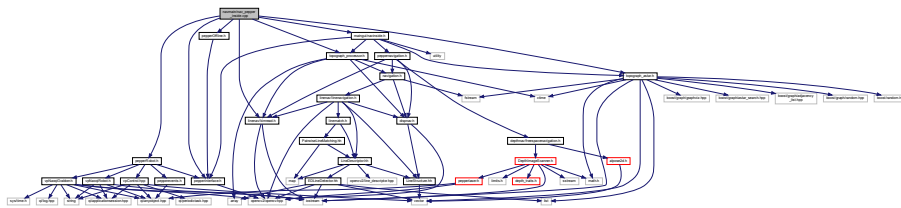
- [Ui](#)

16.46 navmain/nav_pepper_inside.cpp File Reference

```
#include "pepperInterface.h"
#include "pepperRobot.h"
#include "pepperOffline.h"
#include "topograph_astar.h"
#include "linenav/kimread.h"
#include "topograph_processor.h"
```

```
#include "maingui/navinside.h"
```

Include dependency graph for nav_pepper_inside.cpp:



Functions

- void [usage](#) (int argc, char **argv)
- int [main](#) (int argc, char **argv)

16.46.1 Function Documentation

16.46.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

Define Nodes

Define Edges

Definition at line 38 of file nav_pepper_inside.cpp.

16.46.1.2 usage()

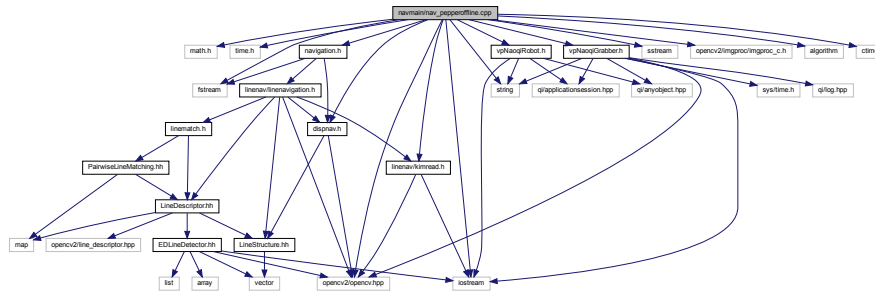
```
void usage (
    int argc,
    char ** argv )
```

Definition at line 33 of file nav_pepper_inside.cpp.

16.48 navmain/nav_pepperoffline.cpp File Reference

```
#include <math.h>
#include <time.h>
#include <fstream>
#include <opencv2/opencv.hpp>
#include <string>
#include <sstream>
#include <opencv2/imgproc/imgproc_c.h>
#include <iostream>
#include <algorithm>
#include <ctime>
#include "linenav/kimread.h"
#include "navigation.h"
#include "linenav/dispsnav.h"
#include "vpNaoqiRobot.h"
#include "vpNaoqiGrabber.h"
```

Include dependency graph for nav_pepperoffline.cpp:



Functions

- void [usage](#) (int argc, char **argv)
- int [main](#) (int argc, char **argv)

16.48.1 Function Documentation

16.48.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

Definition at line 35 of file nav_pepperoffline.cpp.

16.48.1.2 usage()

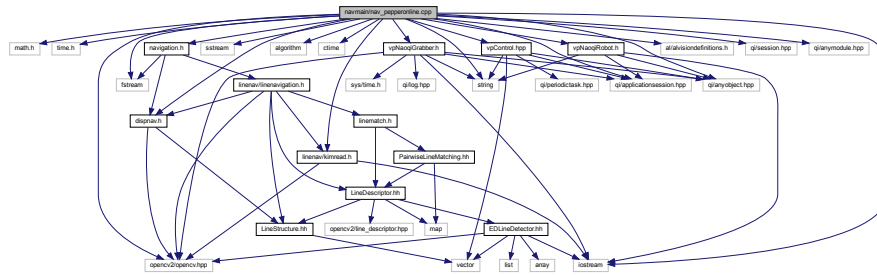
```
void usage (
    int argc,
    char ** argv )
```

Definition at line 28 of file nav_pepperoffline.cpp.

16.49 navmain/nav_pepperonline.cpp File Reference

```
#include <math.h>
#include <time.h>
#include <fstream>
#include <opencv2/opencv.hpp>
#include <string>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <ctime>
#include "linenav/kimread.h"
#include "navigation.h"
#include "linenav/dispsnav.h"
#include <vpNaoqiGrabber.h>
#include <vpNaoqiRobot.h>
#include <al/alvisiondefinitions.h>
#include <qi/session.hpp>
#include <qi/applicationsession.hpp>
#include <qi/anymodule.hpp>
#include <qi/anyobject.hpp>
#include "vpControl.hpp"
```

Include dependency graph for nav_pepperonline.cpp:



Functions

- void [usage](#) (int argc, char **argv)
- const std::string [currentDateTime](#) ()
- int [main](#) (int argc, char **argv)

16.49.1 Function Documentation

16.49.1.1 currentDateTime()

```
const std::string currentDateTime ( )
```

Definition at line 39 of file nav_pepperonline.cpp.

16.49.1.2 main()

```
int main (
    int argc,
    char ** argv )
```

Definition at line 53 of file nav_pepperonline.cpp.

16.49.1.3 usage()

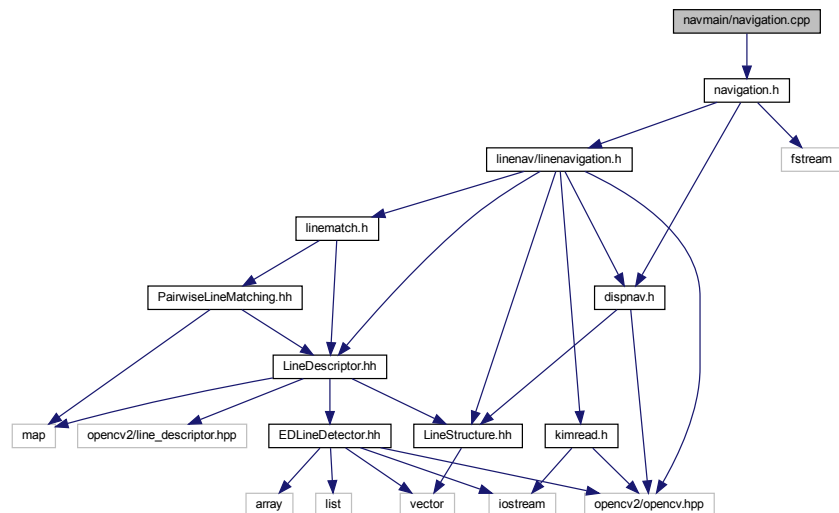
```
void usage (
    int argc,
    char ** argv )
```

Definition at line 35 of file nav_pepperonline.cpp.

16.50 navmain/navigation.cpp File Reference

```
#include "navigation.h"
```

Include dependency graph for navigation.cpp:

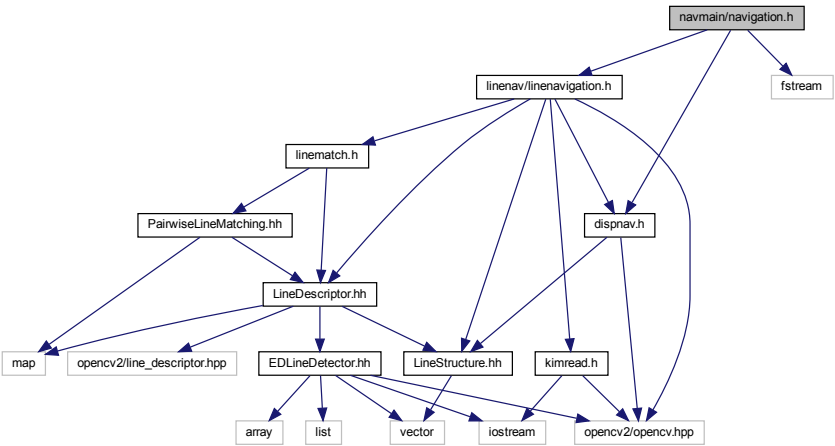


16.51 navmain/navigation.h File Reference

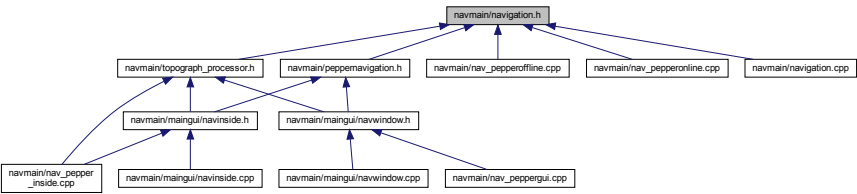
```
#include "linenav/linenavigation.h"
#include "linenav/dispnav.h"
```

```
#include <fstream>
```

Include dependency graph for navigation.h:



This graph shows which files directly or indirectly include this file:



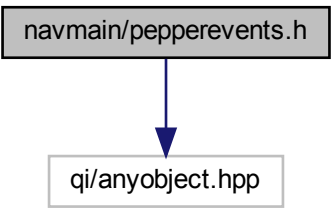
Classes

- class [navigation](#)
The navigation class.

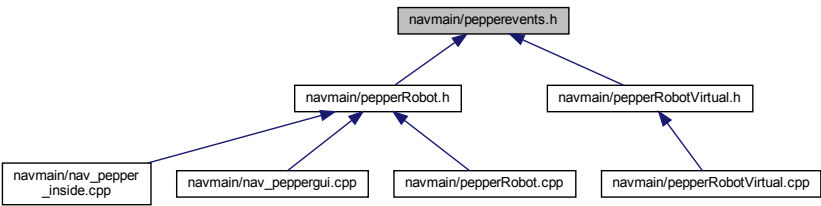
16.52 navmain/pepperevents.h File Reference

```
#include <qi/anyobject.hpp>
```

Include dependency graph for pepperevents.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `pepperServices`
The `pepperServices` class.

Functions

- `QI_REGISTER_OBJECT` (`pepperServices`, `moveCallback`, `armsCallback`)

16.52.1 Function Documentation

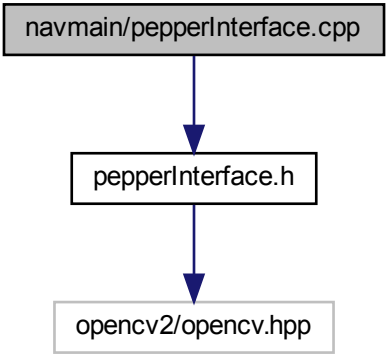
16.52.1.1 QI_REGISTER_OBJECT()

```
QI_REGISTER_OBJECT (
    pepperServices ,
    moveCallback ,
    armsCallback )
```

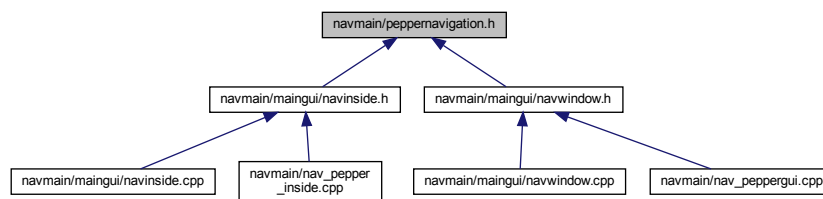
Register Callback Services

16.53 navmain/pepperInterface.cpp File Reference

`#include "pepperInterface.h"`
Include dependency graph for `pepperInterface.cpp`:



This graph shows which files directly or indirectly include this file:



Classes

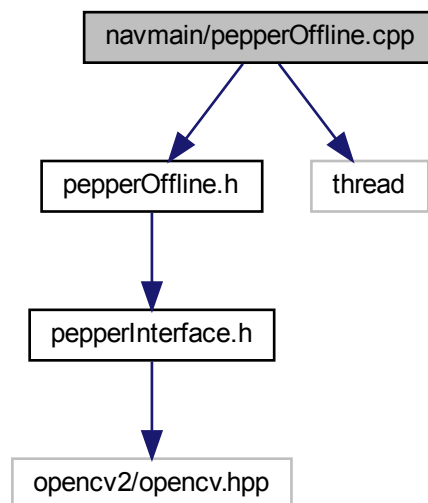
- class [pepperNavigation](#)
The [pepperNavigation](#) class.

16.56 navmain/pepperOffline.cpp File Reference

```
#include "pepperOffline.h"
```

```
#include <thread>
```

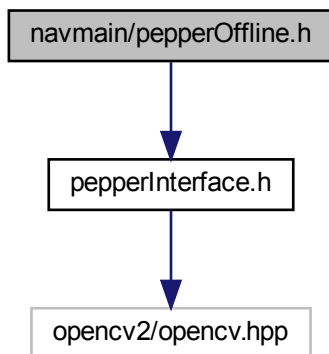
Include dependency graph for pepperOffline.cpp:



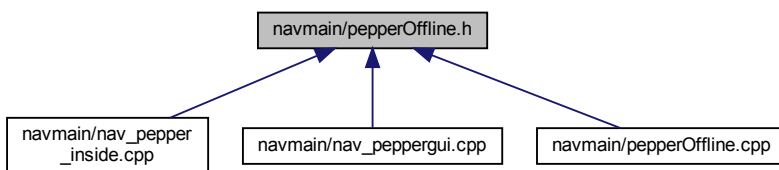
16.57 navmain/pepperOffline.h File Reference

```
#include "pepperInterface.h"
```

Include dependency graph for pepperOffline.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [pepperOffline](#)

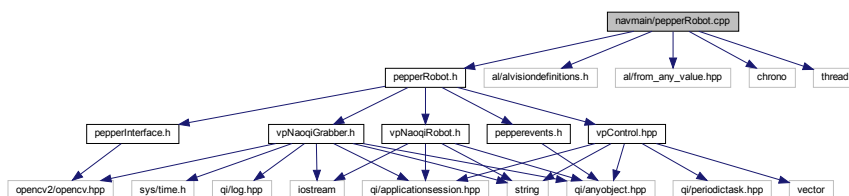
16.58 navmain/pepperRobot.cpp File Reference

```

#include "pepperRobot.h"
#include <al/alvisiondefinitions.h>
#include <al/from_any_value.hpp>
#include <chrono>
#include <thread>

```

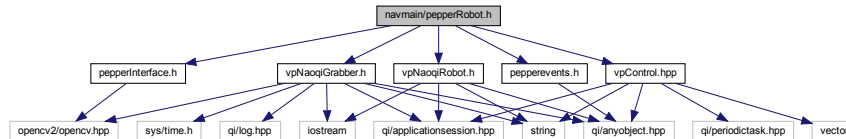
Include dependency graph for pepperRobot.cpp:



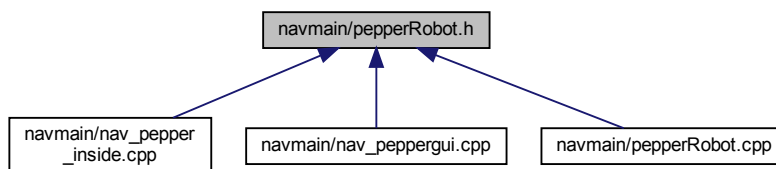
16.59 navmain/pepperRobot.h File Reference

```
#include "pepperInterface.h"
#include "vpNaoqiRobot.h"
#include "vpNaoqiGrabber.h"
#include "vpControl.hpp"
#include "pepperevents.h"
```

Include dependency graph for pepperRobot.h:



This graph shows which files directly or indirectly include this file:



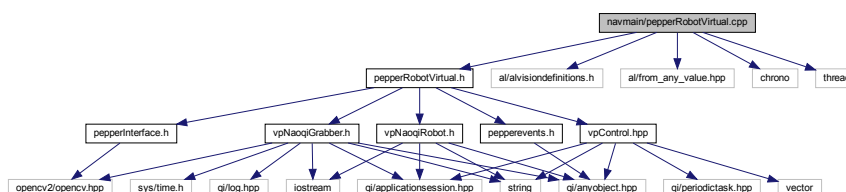
Classes

- class [pepperRobot](#)

16.60 navmain/pepperRobotVirtual.cpp File Reference

```
#include "pepperRobotVirtual.h"
#include <al/alvisiondefinitions.h>
#include <al/from_any_value.hpp>
#include <chrono>
#include <thread>
```

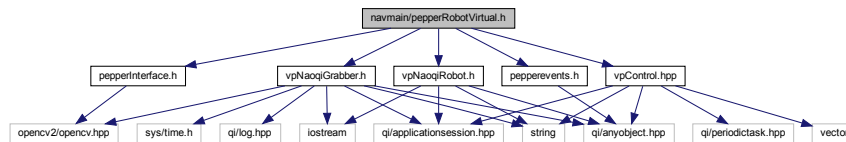
Include dependency graph for pepperRobotVirtual.cpp:



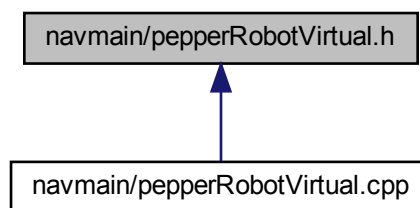
16.61 navmain/pepperRobotVirtual.h File Reference

```
#include "pepperInterface.h"
#include "vpNaoqiRobot.h"
#include "vpNaoqiGrabber.h"
#include "vpControl.hpp"
#include "pepperevents.h"
```

Include dependency graph for pepperRobotVirtual.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [pepperRobotVirtual](#)

16.62 navmain/topograph_astar.h File Reference

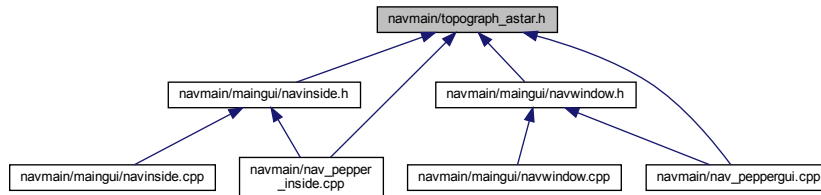
```
#include <boost/graph/astar_search.hpp>
#include <boost/graph/adjacency_list.hpp>
#include <boost/graph/random.hpp>
#include <boost/random.hpp>
#include <boost/graph/graphviz.hpp>
#include <ctime>
#include <vector>
#include <list>
#include <iostream>
#include <fstream>
```

```
#include <math.h>
```

Include dependency graph for topograph_astar.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [astar::location](#)
- class [astar::graph_writer< Name, LocMap >](#)
- class [astar::weight_writer< WeightMap >](#)
- class [astar::distance_heuristic< Graph, CostType, LocMap >](#)
- class [astar::heuristic< Graph, CostType, LocMap >](#)
- struct [astar::found_goal](#)
- class [astar::astar_goal_visitor< Vertex >](#)
- class [astar::astar](#)

The astar class.

Namespaces

- [astar](#)

Typedefs

- typedef float [astar::cost](#)
- typedef boost::adjacency_list< boost::listS, boost::vecS, boost::undirectedS, boost::no_property, boost::property< boost::edge_weight_t, cost > > [astar::mygraph_t](#)
- typedef boost::property_map< mygraph_t, boost::edge_weight_t >::type [astar::WeightMap](#)
- typedef mygraph_t::vertex_descriptor [astar::vertex](#)
- typedef mygraph_t::edge_descriptor [astar::edge_descriptor](#)
- typedef std::pair< int, int > [astar::edge](#)

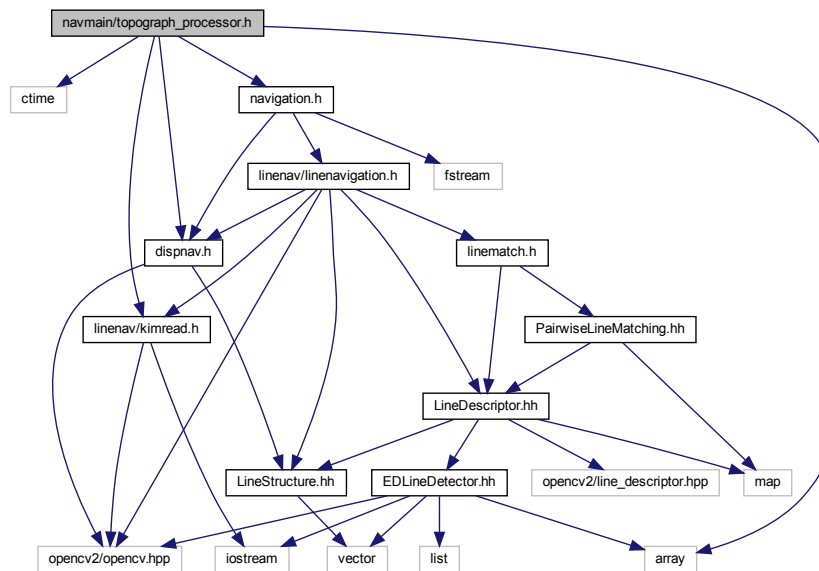
Variables

- const typedef char * [astar::node](#)

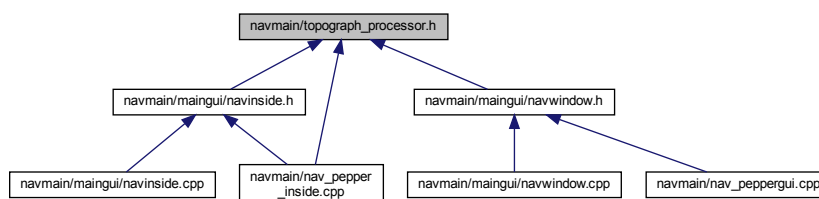
16.63 navmain/topograph_processor.h File Reference

```
#include <ctime>
#include "linenav/kimread.h"
#include "navigation.h"
#include "linenav/dispnave.h"
#include <array>
```

Include dependency graph for topograph_processor.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [tgraph::node](#)

The node struct each node has nodeid starting from 0. nodename : higher level id for node.

- struct [tgraph::edge](#)
- struct [tgraph::branch](#)
- struct [tgraph::topograph](#)

The topograph struct.

- class [tgraph::topmapprocessor](#)

The topmapprocessor class.

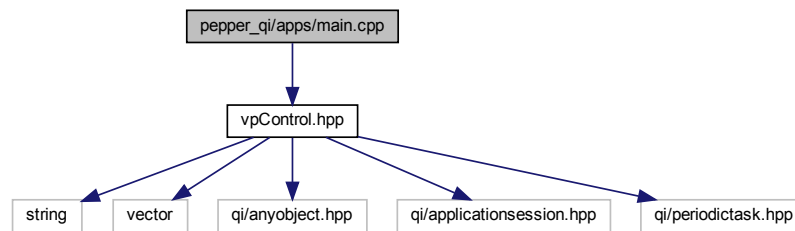
Namespaces

- [tgraph](#)

16.64 pepper_qi/apps/main.cpp File Reference

```
#include "vpControl.hpp"
```

Include dependency graph for main.cpp:



Functions

- int [main](#) (int argc, char **argv)

16.64.1 Function Documentation

16.64.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

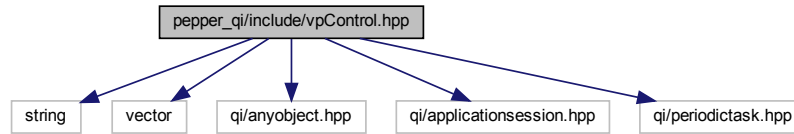
Definition at line 36 of file main.cpp.

16.65 pepper_qi/include/vpControl.hpp File Reference

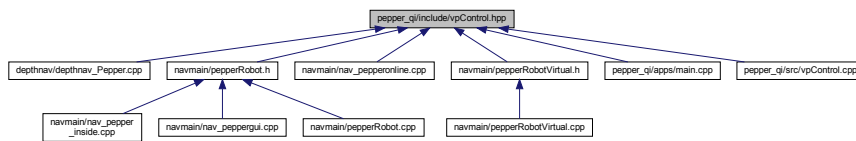
```
#include <string>
#include <vector>
#include <qi/anyobject.hpp>
#include <qi/applicationsession.hpp>
```

```
#include <qi/periodictask.hpp>
```

Include dependency graph for `vpControl.hpp`:



This graph shows which files directly or indirectly include this file:



Classes

- class `vpControl`

Functions

- `QI_REGISTER_OBJECT` (`vpControl`, `getJointValues`, `printTime`, `setDesJointVelocity`, `setOneDesJointVelocity`, `start`, `stop`, `stopJoint`)

16.65.1 Function Documentation

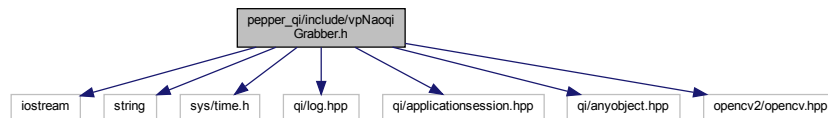
16.65.1.1 QI_REGISTER_OBJECT()

```
QI_REGISTER_OBJECT (
    vpControl ,
    getJointValues ,
    printTime ,
    setDesJointVelocity ,
    setOneDesJointVelocity ,
    start ,
    stop ,
    stopJoint )
```

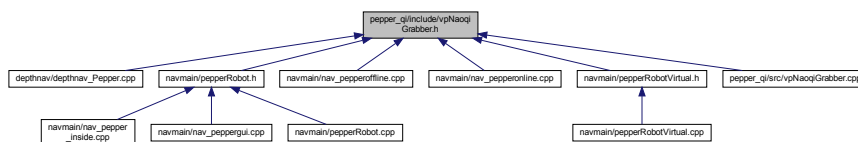
16.66 pepper_qi/include/vpNaoqiGrabber.h File Reference

```
#include <iostream>
#include <string>
#include <sys/time.h>
#include <qi/log.hpp>
#include <qi/applicationsession.hpp>
#include <qi/anyobject.hpp>
#include <opencv2/opencv.hpp>
```

Include dependency graph for vpNaoqiGrabber.h:



This graph shows which files directly or indirectly include this file:



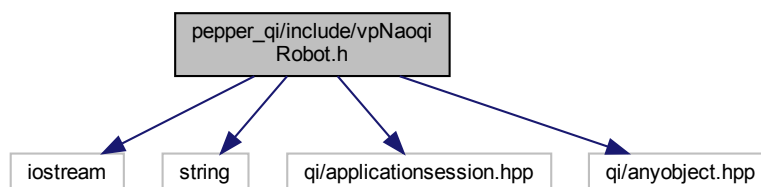
Classes

- class [vpNaoqiGrabber](#)

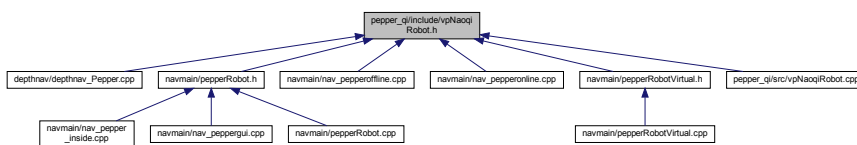
16.67 pepper_qi/include/vpNaoqiRobot.h File Reference

```
#include <iostream>
#include <string>
#include <qi/applicationsession.hpp>
#include <qi/anyobject.hpp>
```

Include dependency graph for vpNaoqiRobot.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [vpNaoqiRobot](#)

Macros

- `#define` [BOOST_SIGNALS_NO_DEPRECATION_WARNING](#)

16.67.1 Macro Definition Documentation

16.67.1.1 BOOST_SIGNALS_NO_DEPRECATION_WARNING

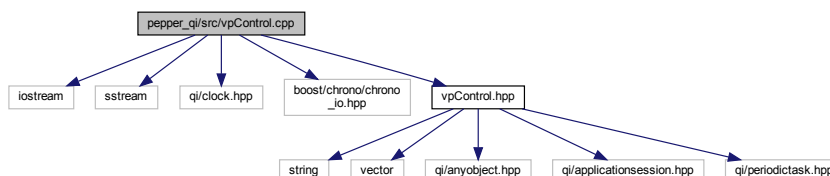
```
#define BOOST_SIGNALS_NO_DEPRECATION_WARNING
```

Definition at line 57 of file `vpNaoqiRobot.h`.

16.68 pepper_qi/src/vpControl.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <qi/clock.hpp>
#include <boost/chrono/chrono_io.hpp>
#include "vpControl.hpp"
```

Include dependency graph for `vpControl.cpp`:



Macros

- #define [FLAGACC](#) 1
- #define [FLAGCTE](#) 2
- #define [FLAGDEC](#) 3
- #define [FLAGSTO](#) 4
- #define [DELTAQMIN](#) 0.0001
- #define [OFFSET_BUTEE](#) 0

16.68.1 Macro Definition Documentation

16.68.1.1 DELTAQMIN

```
#define DELTAQMIN 0.0001
```

Definition at line 50 of file vpControl.cpp.

16.68.1.2 FLAGACC

```
#define FLAGACC 1
```

Definition at line 45 of file vpControl.cpp.

16.68.1.3 FLAGCTE

```
#define FLAGCTE 2
```

Definition at line 46 of file vpControl.cpp.

16.68.1.4 FLAGDEC

```
#define FLAGDEC 3
```

Definition at line 47 of file vpControl.cpp.

16.68.1.5 FLAGSTO

```
#define FLAGSTO 4
```

Definition at line 48 of file vpControl.cpp.

16.68.1.6 OFFSET_BUTEE

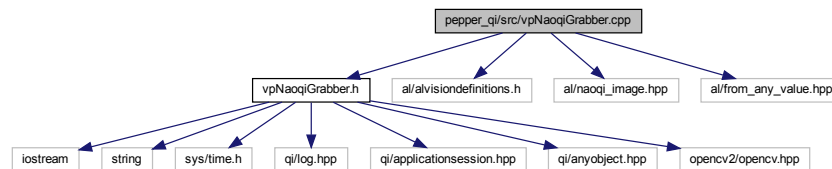
```
#define OFFSET_BUTEE 0
```

Definition at line 51 of file vpControl.cpp.

16.69 pepper_qi/src/vpNaoqiGrabber.cpp File Reference

```
#include "vpNaoqiGrabber.h"
#include "al/alvisiondefinitions.h"
#include "al/naoqi_image.hpp"
#include "al/from_any_value.hpp"
```

Include dependency graph for vpNaoqiGrabber.cpp:



16.70 pepper_qi/src/vpNaoqiRobot.cpp File Reference

```
#include "vpNaoqiRobot.h"
#include "al/from_any_value.hpp"
```

Include dependency graph for vpNaoqiRobot.cpp:

