Primitive Identification Tagging & Tracking (PITT). The "Object Table Segmentation" package

v1.1

Generated by Doxygen 1.8.6

Thu Mar 3 2016 18:56:11

Contents

Chapter 1

The mainpage documentation

To do !!!

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 pcm Namespace Reference

Classes

- class PCManager

Functions

- static search::KdTree
  < PointXYZ >::Ptr tree (new search::KdTree< PointXYZ >())

Variables

- static NormalEstimation
  < PointXYZ, Normal > ne
- static VoxelGrid< PointXYZ > sor

5.1.1 Function Documentation

5.1.1.1 static search::KdTree< PointXYZ>::Ptr pcm::tree (new search::KdTree< PointXYZ > *()*) `[static]`

Referenced by clusterize(), and pcm::PCManager::estimateNormal().

5.1.2 Variable Documentation

5.1.2.1 NormalEstimation< PointXYZ, Normal> pcm::ne `[static]`

Definition at line 35 of file PCManager.cpp.

Referenced by pcm::PCManager::estimateNormal().

5.1.2.2 VoxelGrid< PointXYZ> pcm::sor `[static]`

Definition at line 38 of file PCManager.cpp.

Referenced by pcm::PCManager::downSampling().

5.2 pcp Namespace Reference

Classes

- class PCPrimitive

Chapter 6

Class Documentation

6.1 pcm::PCManager Class Reference

```
#include <PCManager.h>
```

Public Member Functions

- PCManager ()
- PCManager (bool visualizationFlag)
- virtual ∼PCManager ()
- vector< PCLCloudPtr > getCloudFromIdx (PrimitiveIdxPtr indices)
- void visualize ()
- PCPrimitivePtr getPrimitiveShape (int idx)
- int addPrimitiveShape (string shapeName, PCLCloudPtr cloud, PCLNormalPtr norms, bool visualFlag)
- int clearPtimitiveShape ()
- PCLCloudPtr getOriginalCloud ()
- PointCloud2 getOriginalCloudRosMsg ()
- PCLNormalPtr getOriginalNormal ()
- PointCloud2 getOriginalNormalRosMsg ()
- bool getVisualizationFlag ()
- PCLVisualizer getVisor ()
- void setOriginalCloud (PCLCloudPtr cloud)
- void setOriginalCloud (PCLCloudPtr cloud, int normSearch, float downSpanX, float downSpanY, float downSpanZ)
- void setOriginalCloud (PointCloud2Ptr cloud)
- void setOriginalCloud (PointCloud2Ptr cloud, int normSearch, float downSpanX, float downSpanY, float downSpanZ)
- void setVisualizationFlag (bool flag)

Static Public Member Functions

- static PCLCloudPtr copyCloud (PCLCloudPtr input)
- static PCLNormalPtr copyNormals (PCLNormalPtr input)
- static ModelCoefficients::Ptr copyCoefficients (ModelCoefficients::Ptr input)
- static PCLCloudPtr downSampling (PCLCloudPtr input)
- static PCLCloudPtr downSampling (PCLCloudPtr input, float span)
- static PCLCloudPtr downSampling (PCLCloudPtr input, float spanX, float spanY, float spanZ)
- static PCLNormalPtr estimateNormal (PCLCloudPtr input)
- static PCLNormalPtr estimateNormal (PCLCloudPtr input, int search)

- static PointCloud2 cloudToRosMsg (PCLCloudPtr input)
- static PCLCloudPtr cloudForRosMsg (PointCloud2 input)
- static PCLCloudPtr cloudForRosMsg (PointCloud2Ptr input)
- static PointCloud2 normToRosMsg (PCLNormalPtr input)
- static PCLNormalPtr normForRosMsg (PointCloud2 input)
- static vector< int > inlierToVectorMsg (PointIndices::Ptr inliers)
- static vector< float > coefficientToVectorMsg (ModelCoefficients::Ptr coefficients)
- static PCLVisualizer createVisor (string title)
- static void updateVisor (PCLVisualizer viewer, PCLCloudPtr cloud, int R, int G, int B, string name)
- static void updateVisor (PCLVisualizer viewer, PCLCloudPtr cloud, string name)
- static void updateVisor (PCLVisualizer viewer, PCLCloudPtr cloud, PCLNormalPtr normals, int R, int G, int B, string name)
- static void updateVisor (PCLVisualizer viewer, PCLCloudPtr cloud, PCLNormalPtr normals, string name)
- static void updateVisor (PCLVisualizer viewer, PointXYZ point, int R, int G, int B, string name)
- static void updateVisor (PCLVisualizer viewer, PointXYZ point, string name)
- static void clearVisor (PCLVisualizer viewer)
- static vector< PCLCloudPtr > getCloudFromIdx (PCLCloudPtr originalCloud, PrimitiveIdxPtr indices)
- static bool writeToFile (string txt, string filePath)

Static Public Attributes

- static const bool DEFAULT_VISUALIZATION_FLAG = false
- static const int VISUALIZER_POINT_SIZE = 3
- static const int VISUALIZER_POINT_SIZE_BIG = 10
- static const string DEFAULT_CLOUD_NAME_SUFFIX = "_cloud"
- static const string DEFAULT_NORM_NAME_SUFFIX = "_normal"
- static const string DEFAULT_ORIGINAL_CLOUD_VIEWER_NAME = "original"
- static const int DEFAULT_NORM_LEVEL = 5
- static const float DEFAULT_NORM_SCALE = 0.02f
- static const string DEFAULT_VISUALIZER_TITLE = "PointCloud manager"
- static const int DEFAULT_NORM_SEARCH = 50
- static const float DEFAULT_DOWSEAMPLIG_RATE = 0.01f
- static const string DEEP_FILTER_SERVICE_NAME = "deepFilterSrv"
- static const string SUPPORT_FILTER_SERVICE_NAME = "supportSegmentationSrv"
- static const string CUSTER_FILTER_SERVICE_NAME = "clusterSegmentationSrv"
- static const string ARM_FILTER_SERVICE_NAME = "robotArmCloudFiltering"
- static const string RANSAC_SPHERE_FILTER_SERVICE_NAME = "sphereSegmentationSrv"
- static const string RANSAC_CYLINDER_FILTER_SERVICE_NAME = "cylinderSegmentationSrv"
- static const string RANSAC_CONE_FILTER_SERVICE_NAME = "coneSegmentationSrv"
- static const string RANSAC_PLANE_FILTER_SERVICE_NAME = "planeSegmentationSrv"
- static const string SEMANTIC_SCENE_RECOGNITION_SERVICE_NAME = "semanticSceneRecognitonSrv"
- static const float DEFAULT_SERVICE_PARAMETER_REQUEST = -1.0f

Private Member Functions

- void initialize (bool visualizationFlag)

Private Attributes

- PCLCloudPtr originalCloud
- PCLNormalPtr originalNorms
- PCLVisualizer visor
- bool visualizationFlag
- vector< PCPrimitivePtr > primitiveList

6.1.1 Detailed Description

Definition at line 31 of file PCManager.h.

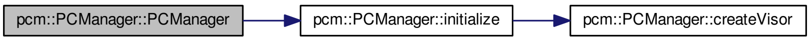
6.1.2 Constructor & Destructor Documentation

6.1.2.1 pcm::PCManager::PCManager ( )

Definition at line 239 of file PCManager.cpp.

References DEFAULT_VISUALIZATION_FLAG, and initialize().

Here is the call graph for this function:

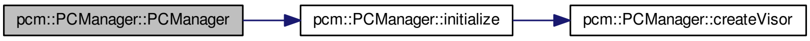

pcm::PCManager::PCManager
pcm::PCManager::initialize
pcm::PCManager::createVisor

6.1.2.2 pcm::PCManager::PCManager (bool *visualizationFlag*)

Definition at line 242 of file PCManager.cpp.

References initialize().

Here is the call graph for this function:


pcm::PCManager::PCManager
pcm::PCManager::initialize
pcm::PCManager::createVisor

6.1.2.3 pcm::PCManager::∼PCManager ( ) [virtual]

Definition at line 247 of file PCManager.cpp.

6.1.3 Member Function Documentation

6.1.3.1 int pcm::PCManager::addPrimitiveShape (string *shapeName,* PCLCloudPtr *cloud,* PCLNormalPtr *norms,* bool *visualFlag*)

6.1.3.2 int pcm::PCManager::clearPtimitiveShape ( )

6.1.3.3 void pcm::PCManager::clearVisor (PCLVisualizer *viewer*) [static]

Definition at line 179 of file PCManager.cpp.

6.1.3.4 PCLCloudPtr pcm::PCManager::cloudForRosMsg (PointCloud2 *input*) [static]

Definition at line 101 of file PCManager.cpp.

Referenced by clusterize(), ransacConeDetaction(), ransacCylinderDetaction(), ransacPlaneDetaction(), ransacSphereDetaction(), and setOriginalCloud().

6.1.3.5 PCLCloudPtr pcm::PCManager::cloudForRosMsg (PointCloud2Ptr *input*) [static]

Definition at line 96 of file PCManager.cpp.

6.1.3.6 PointCloud2 pcm::PCManager::cloudToRosMsg (PCLCloudPtr *input*) [static]

Definition at line 91 of file PCManager.cpp.

Referenced by getOriginalCloudRosMsg().

6.1.3.7 vector< float > pcm::PCManager::coefficientToVectorMsg (ModelCoefficients::Ptr *coefficients*) [static]

Definition at line 123 of file PCManager.cpp.

Referenced by ransacConeDetaction(), ransacCylinderDetaction(), ransacPlaneDetaction(), and ransacSphereDetaction().

6.1.3.8 PCLCloudPtr pcm::PCManager::copyCloud (PCLCloudPtr *input*) [static]

Definition at line 41 of file PCManager.cpp.

6.1.3.9 ModelCoefficients::Ptr pcm::PCManager::copyCoefficients (ModelCoefficients::Ptr *input*) [static]

Definition at line 60 of file PCManager.cpp.

6.1.3.10 PCLNormalPtr pcm::PCManager::copyNormals (PCLNormalPtr *input*) [static]

Definition at line 54 of file PCManager.cpp.

6.1.3.11 PCLVisualizer pcm::PCManager::createVisor (string *title*) [static]

Definition at line 132 of file PCManager.cpp.

Referenced by initialize(), main(), and setVisualizationFlag().

6.1.3.12 PCLCloudPtr pcm::PCManager::downSampling (PCLCloudPtr *input*) [static]

Definition at line 66 of file PCManager.cpp.

References DEFAULT_DOWSEAMPLIG_RATE.

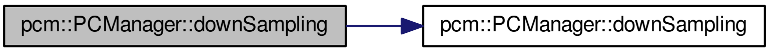
Referenced by downSampling(), and setOriginalCloud().

6.1.3.13 PCLCloudPtr pcm::PCManager::downSampling (PCLCloudPtr *input,* float *span*) [static]

Definition at line 69 of file PCManager.cpp.

References downSampling().

Here is the call graph for this function:



6.1.3.14 **PCLCloudPtr** pcm::PCManager::downSampling (**PCLCloudPtr** *input,* float *spanX,* float *spanY,* float *spanZ*) [static]

Definition at line 72 of file PCManager.cpp.

References pcm::sor.

6.1.3.15 **PCLNormalPtr** pcm::PCManager::estimateNormal (**PCLCloudPtr** *input*) [static]

Definition at line 79 of file PCManager.cpp.

References DEFAULT_NORM_SEARCH.

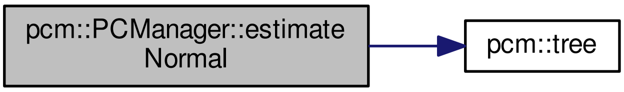
Referenced by setOriginalCloud().

6.1.3.16 **PCLNormalPtr** pcm::PCManager::estimateNormal (**PCLCloudPtr** *input,* int *search*) [static]

Definition at line 82 of file PCManager.cpp.

References pcm::ne, and pcm::tree().

Here is the call graph for this function:



6.1.3.17 vector< **PCLCloudPtr** > pcm::PCManager::getCloudFromIdx (**PCLCloudPtr** *originalCloud,* **PrimitiveIdxPtr** *indices*) [static]

Definition at line 185 of file PCManager.cpp.

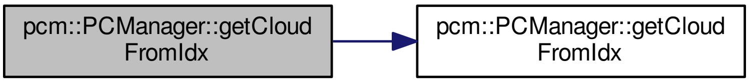
Referenced by getCloudFromIdx().

6.1.3.18 vector< **PCLCloudPtr** > pcm::PCManager::getCloudFromIdx (**PrimitiveIdxPtr** *indices*)

Definition at line 231 of file PCManager.cpp.

References getCloudFromIdx(), and originalCloud.

Here is the call graph for this function:


pcm::PCManager::getCloud FromIdx
pcm::PCManager::getCloud FromIdx

6.1.3.19 PCLCloudPtr pcm::PCManager::getOriginalCloud ( )

Definition at line 285 of file PCManager.cpp.

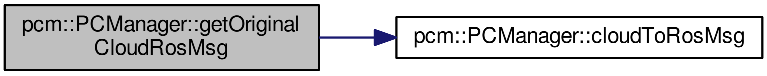
References originalCloud.

6.1.3.20 PointCloud2 pcm::PCManager::getOriginalCloudRosMsg ( )

Definition at line 288 of file PCManager.cpp.

References cloudToRosMsg(), and originalCloud.

Here is the call graph for this function:


pcm::PCManager::getOriginal CloudRosMsg
pcm::PCManager::cloudToRosMsg

6.1.3.21 PCLNormalPtr pcm::PCManager::getOriginalNormal ( )

Definition at line 292 of file PCManager.cpp.

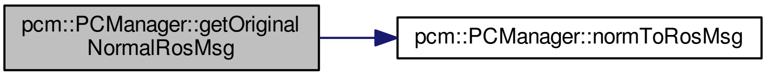
References originalNorms.

6.1.3.22 PointCloud2 pcm::PCManager::getOriginalNormalRosMsg ( )

Definition at line 295 of file PCManager.cpp.

References normToRosMsg(), and originalNorms.

Here is the call graph for this function:


pcm::PCManager::getOriginal NormalRosMsg
pcm::PCManager::normToRosMsg

6.1.3.23 PCPrimitivePtr pcm::PCManager::getPrimitiveShape (int *idx*)

6.1.3.24 PCLVisualizer pcm::PCManager::getVisor ( )

Definition at line 302 of file PCManager.cpp.

References visor.

6.1.3.25 bool pcm::PCManager::getVisualizationFlag ( )

Definition at line 299 of file PCManager.cpp.

References visualizationFlag.

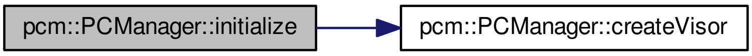
6.1.3.26 void pcm::PCManager::initialize (bool *visualizationFlag*) [private]

Definition at line 343 of file PCManager.cpp.

References createVisor(), DEFAULT_VISUALIZER_TITLE, visor, and visualizationFlag.

Referenced by PCManager().

Here is the call graph for this function:



6.1.3.27 vector< int > pcm::PCManager::inlierToVectorMsg (PointIndices::Ptr *inliers*) [static]

Definition at line 116 of file PCManager.cpp.

Referenced by ransacConeDetaction(), ransacCylinderDetaction(), ransacPlaneDetaction(), and ransacSphereDetaction().

6.1.3.28 PCLNormalPtr pcm::PCManager::normForRosMsg (PointCloud2 *input*) [static]

Definition at line 111 of file PCManager.cpp.

Referenced by ransacConeDetection(), ransacCylinderDetection(), ransacPlaneDetection(), and ransacSphereDetection().

6.1.3.29 PointCloud2 pcm::PCManager::normToRosMsg (PCLNormalPtr *input*) [static]

Definition at line 106 of file PCManager.cpp.

Referenced by getOriginalNormalRosMsg().

6.1.3.30 void pcm::PCManager::setOriginalCloud (PCLCloudPtr *cloud*)

Definition at line 308 of file PCManager.cpp.

References DEFAULT_DOWSEAMPLIG_RATE, and DEFAULT_NORM_SEARCH.

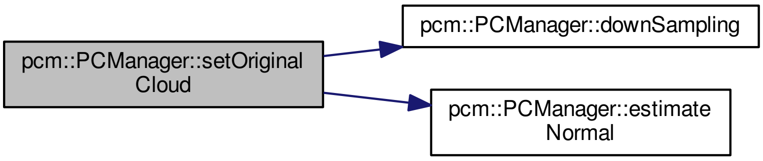
Referenced by setOriginalCloud().

6.1.3.31 void pcm::PCManager::setOriginalCloud (PCLCloudPtr *cloud,* int *normSearch,* float *downSpanX,* float *downSpanY,* float *downSpanZ*)

Definition at line 311 of file PCManager.cpp.

References downSampling(), estimateNormal(), originalCloud, and originalNorms.

Here is the call graph for this function:

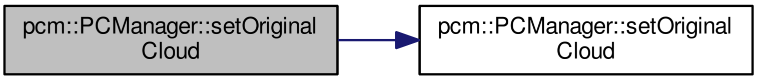

pcm::PCManager::setOriginal Cloud
pcm::PCManager::downSampling
pcm::PCManager::estimate Normal

6.1.3.32 void pcm::PCManager::setOriginalCloud (PointCloud2Ptr *cloud*)

Definition at line 319 of file PCManager.cpp.

References DEFAULT_DOWSEAMPLIG_RATE, DEFAULT_NORM_SEARCH, and setOriginalCloud().

Here is the call graph for this function:

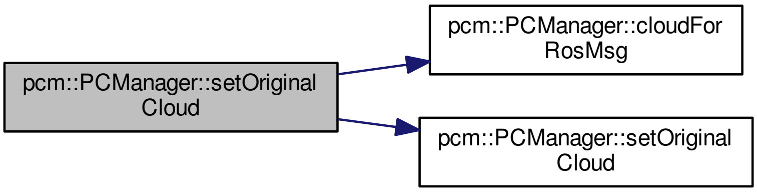

pcm::PCManager::setOriginal Cloud
pcm::PCManager::setOriginal Cloud

6.1.3.33 void pcm::PCManager::setOriginalCloud (PointCloud2Ptr *cloud,* int *normSearch,* float *downSpanX,* float *downSpanY,* float *downSpanZ*)

Definition at line 322 of file PCManager.cpp.

References cloudForRosMsg(), and setOriginalCloud().

Here is the call graph for this function:

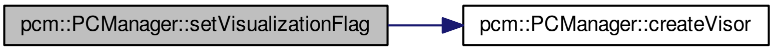

pcm::PCManager::setOriginal Cloud
pcm::PCManager::cloudFor RosMsg
pcm::PCManager::setOriginal Cloud

6.1.3.34 void pcm::PCManager::setVisualizationFlag (bool *flag*)

Definition at line 326 of file PCManager.cpp.

References createVisor(), DEFAULT_VISUALIZER_TITLE, visor, and visualizationFlag.

Here is the call graph for this function:


pcm::PCManager::setVisualizationFlag
pcm::PCManager::createVisor

6.1.3.35 void pcm::PCManager::updateVisor (PCLVisualizer *viewer,* PCLCloudPtr *cloud,* int *R,* int *G,* int *B,* string *name*) [static]

Definition at line 154 of file PCManager.cpp.

References VISUALIZER_POINT_SIZE.

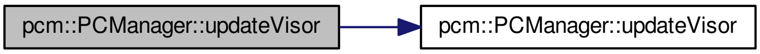
Referenced by ransacConeDetaction(), ransacCylinderDetaction(), and updateVisor().

6.1.3.36 void pcm::PCManager::updateVisor (PCLVisualizer *viewer,* PCLCloudPtr *cloud,* string *name*) [static]

Definition at line 162 of file PCManager.cpp.

References updateVisor().

Here is the call graph for this function:



6.1.3.37 void pcm::PCManager::updateVisor (**PCLVisualizer** *viewer,* **PCLCloudPtr** *cloud,* **PCLNormalPtr** *normals,* int *R,* int *G,* int *B,* string *name*) [static]

Definition at line 166 of file PCManager.cpp.

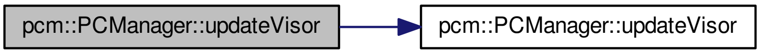
References DEFAULT_NORM_LEVEL, DEFAULT_NORM_NAME_SUFFIX, DEFAULT_NORM_SCALE, and VISUALIZER_POINT_SIZE.

6.1.3.38 void pcm::PCManager::updateVisor (**PCLVisualizer** *viewer,* **PCLCloudPtr** *cloud,* **PCLNormalPtr** *normals,* string *name*) [static]

Definition at line 176 of file PCManager.cpp.

References updateVisor().

Here is the call graph for this function:



6.1.3.39 void pcm::PCManager::updateVisor (**PCLVisualizer** *viewer,* PointXYZ *point,* int *R,* int *G,* int *B,* string *name*) [static]

Definition at line 140 of file PCManager.cpp.

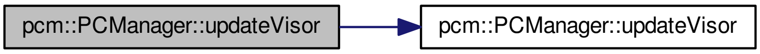
References VISUALIZER_POINT_SIZE_BIG.

6.1.3.40 void pcm::PCManager::updateVisor (**PCLVisualizer** *viewer,* PointXYZ *point,* string *name*) [static]

Definition at line 150 of file PCManager.cpp.

References updateVisor().

Here is the call graph for this function:



6.1.3.41 void pcm::PCManager::visualize ( )

6.1.3.42 bool pcm::PCManager::writeToFile (string *txt*, string *filePath*) [static]

Definition at line 219 of file PCManager.cpp.

6.1.4 Member Data Documentation

6.1.4.1 const string pcm::PCManager::ARM_FILTER_SERVICE_NAME = "robotArmCloudFiltering" [static]

Definition at line 127 of file PCManager.h.

6.1.4.2 const string pcm::PCManager::CUSTER_FILTER_SERVICE_NAME = "clusterSegmentationSrv" [static]

Definition at line 126 of file PCManager.h.

6.1.4.3 const string pcm::PCManager::DEEP_FILTER_SERVICE_NAME = "deepFilterSrv" [static]

Definition at line 124 of file PCManager.h.

6.1.4.4 const string pcm::PCManager::DEFAULT_CLOUD_NAME_SUFFIX = "_cloud" [static]

Definition at line 114 of file PCManager.h.

6.1.4.5 const float pcm::PCManager::DEFAULT_DOWSEAMPLIG_RATE = 0.01f [static]

Definition at line 122 of file PCManager.h.

Referenced by downSampling(), and setOriginalCloud().

6.1.4.6 const int pcm::PCManager::DEFAULT_NORM_LEVEL = 5 [static]

Definition at line 117 of file PCManager.h.

Referenced by updateVisor().

6.1.4.7 const string pcm::PCManager::DEFAULT_NORM_NAME_SUFFIX = "_normal" [static]

Definition at line 115 of file PCManager.h.

Referenced by updateVisor().

6.1.4.8 const float pcm::PCManager::DEFAULT_NORM_SCALE = 0.02f [static]

Definition at line 118 of file PCManager.h.

Referenced by updateVisor().

6.1.4.9 const int pcm::PCManager::DEFAULT_NORM_SEARCH = 50 [static]

Definition at line 121 of file PCManager.h.

Referenced by estimateNormal(), and setOriginalCloud().

6.1.4.10 const string pcm::PCManager::DEFAULT_ORIGINAL_CLOUD_VIEWER_NAME = "original" [static]

Definition at line 116 of file PCManager.h.

6.1.4.11 const float pcm::PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST = -1.0f [static]

Definition at line 134 of file PCManager.h.

6.1.4.12 const bool pcm::PCManager::DEFAULT_VISUALIZATION_FLAG = false [static]

Definition at line 111 of file PCManager.h.

Referenced by PCManager().

6.1.4.13 const string pcm::PCManager::DEFAULT_VISUALIZER_TITLE = "PointCloud manager" [static]

Definition at line 119 of file PCManager.h.

Referenced by initialize(), and setVisualizationFlag().

6.1.4.14 PCLCloudPtr pcm::PCManager::originalCloud [private]

Definition at line 33 of file PCManager.h.

Referenced by getCloudFromIdx(), getOriginalCloud(), getOriginalCloudRosMsg(), and setOriginalCloud().

6.1.4.15 PCLNormalPtr pcm::PCManager::originalNorms [private]

Definition at line 34 of file PCManager.h.

Referenced by getOriginalNormal(), getOriginalNormalRosMsg(), and setOriginalCloud().

6.1.4.16 vector< PCPrimitivePtr > pcm::PCManager::primitiveList [private]

Definition at line 39 of file PCManager.h.

6.1.4.17 const string pcm::PCManager::RANSAC_CONE_FILTER_SERVICE_NAME = "coneSegmentationSrv" [static]

Definition at line 130 of file PCManager.h.

Referenced by main().

6.1.4.18 const string pcm::PCManager::RANSAC_CYLINDER_FILTER_SERVICE_NAME = "cylinderSegmentationSrv" [static]

Definition at line 129 of file PCManager.h.

Referenced by main().

6.1.4.19 const string pcm::PCManager::RANSAC_PLANE_FILTER_SERVICE_NAME = "planeSegmentationSrv" [static]

Definition at line 131 of file PCManager.h.

Referenced by main().

6.1.4.20 const string pcm::PCManager::RANSAC_SPHERE_FILTER_SERVICE_NAME = "sphereSegmentationSrv" [static]

Definition at line 128 of file PCManager.h.

Referenced by main().

6.1.4.21 const string pcm::PCManager::SEMANTIC_SCENE_RECOGNITION_SERVICE_NAME = "semanticSceneRecognitonSrv" [static]

Definition at line 132 of file PCManager.h.

6.1.4.22 const string pcm::PCManager::SUPPORT_FILTER_SERVICE_NAME = "supportSegmentationSrv" [static]

Definition at line 125 of file PCManager.h.

6.1.4.23 PCLVisualizer pcm::PCManager::visor [private]

Definition at line 36 of file PCManager.h.

Referenced by getVisor(), initialize(), and setVisualizationFlag().

6.1.4.24 bool pcm::PCManager::visualizationFlag [private]

Definition at line 37 of file PCManager.h.

Referenced by getVisualizationFlag(), initialize(), and setVisualizationFlag().

6.1.4.25 const int pcm::PCManager::VISUALIZER_POINT_SIZE = 3 [static]

Definition at line 112 of file PCManager.h.

Referenced by updateVisor().

6.1.4.26 const int pcm::PCManager::VISUALIZER_POINT_SIZE_BIG = 10 [static]

Definition at line 113 of file PCManager.h.

Referenced by updateVisor().

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

- PCManager.h
- PCManager.cpp

6.2 pcp::PCPrimitive Class Reference

```
#include <PCPrimitive.h>
```

Public Member Functions

- PCPrimitive (string shapename, int shapeMapidx, bool visualFlag, PCLCloudPtr cloud, PCLNormalPtr norms)
- virtual ∼PCPrimitive ()
- string getShapeName ()
- string getVisualizationName ()
- bool getVisualizationFlag ()
- int getShapeMapidx ()
- PCLCloud getPrimitiveCloud ()
- PCLNormal getPrimitiveNormal ()

Static Public Attributes

- static const string DEFAULT_SHAPE_NAME_PLANE = "plane"
- static const string DEFAULT_SHAPE_NAME_CLUSTER = "cluster"
- static const string DEFAULT_VISUALIZATION_NAME_SEPARATOR = "-"

Private Member Functions

- string getVisualizationNameFromTag (int idx)
- ModelCoefficients copyCoefficients (ModelCoefficients::Ptr input)

Private Attributes

- string shapeName
- string visualizationName
- bool visualizationFlag
- int shapeMapIdx
- PCLCloud primitiveCloud
- PCLNormal primitiveNormals
- ModelCoefficients primitiveCoefficients

6.2.1 Detailed Description

Definition at line 27 of file PCPrimitive.h.

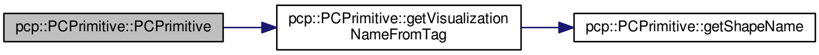
6.2.2 Constructor & Destructor Documentation

6.2.2.1 pcp::PCPrimitive::PCPrimitive (string *shapename,* int *shapeMapidx,* bool *visualFlag,* PCLCloudPtr *cloud,* PCLNormalPtr *norms*)

Definition at line 17 of file PCPrimitive.cpp.

References getVisualizationNameFromTag(), primitiveCloud, primitiveNormals, shapeMapIdx, shapeName, visualizationFlag, and visualizationName.

Here is the call graph for this function:


pcp::PCPrimitive::PCPrimitive
pcp::PCPrimitive::getVisualization NameFromTag
pcp::PCPrimitive::getShapeName

6.2.2.2 pcp::PCPrimitive::∼PCPrimitive ( ) [virtual]

Definition at line 26 of file PCPrimitive.cpp.

6.2.3 Member Function Documentation

6.2.3.1 ModelCoefficients pcp::PCPrimitive::copyCoefficients (ModelCoefficients::Ptr *input*) [private]

Definition at line 64 of file PCPrimitive.cpp.

6.2.3.2 PCLCloud pcp::PCPrimitive::getPrimitiveCloud ( )

Definition at line 88 of file PCPrimitive.cpp.

References primitiveCloud.

6.2.3.3 PCLNormal pcp::PCPrimitive::getPrimitiveNormal ( )

Definition at line 91 of file PCPrimitive.cpp.

References primitiveNormals.

6.2.3.4 int pcp::PCPrimitive::getShapeMapidx ( )

Definition at line 82 of file PCPrimitive.cpp.

References shapeMapIdx.

6.2.3.5 string pcp::PCPrimitive::getShapeName ( )

Definition at line 73 of file PCPrimitive.cpp.

References shapeName.

Referenced by getVisualizationNameFromTag().

6.2.3.6 bool pcp::PCPrimitive::getVisualizationFlag ( )

Definition at line 79 of file PCPrimitive.cpp.

References visualizationFlag.

6.2.3.7 string pcp::PCPrimitive::getVisualizationName ( )

Definition at line 76 of file PCPrimitive.cpp.

References visualizationName.

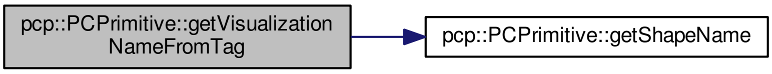
6.2.3.8 string pcp::PCPrimitive::getVisualizationNameFromTag (int *idx*) [private]

Definition at line 35 of file PCPrimitive.cpp.

References DEFAULT_VISUALIZATION_NAME_SEPARATOR, and getShapeName().

Referenced by PCPrimitive().

Here is the call graph for this function:


pcp::PCPrimitive::getVisualization NameFromTag
pcp::PCPrimitive::getShapeName

6.2.4 Member Data Documentation

6.2.4.1 const string pcp::PCPrimitive::DEFAULT_SHAPE_NAME_CLUSTER = "cluster" [static]

Definition at line 74 of file PCPrimitive.h.

6.2.4.2 const string pcp::PCPrimitive::DEFAULT_SHAPE_NAME_PLANE = "plane" [static]

Definition at line 73 of file PCPrimitive.h.

6.2.4.3 const string pcp::PCPrimitive::DEFAULT_VISUALIZATION_NAME_SEPARATOR = "-" [static]

Definition at line 76 of file PCPrimitive.h.

Referenced by getVisualizationNameFromTag().

6.2.4.4 PCLCloud pcp::PCPrimitive::primitiveCloud [private]

Definition at line 37 of file PCPrimitive.h.

Referenced by getPrimitiveCloud(), and PCPrimitive().

6.2.4.5 ModelCoefficients pcp::PCPrimitive::primitiveCoefficients [private]

Definition at line 39 of file PCPrimitive.h.

6.2.4.6 PCLNormal pcp::PCPrimitive::primitiveNormals [private]

Definition at line 38 of file PCPrimitive.h.

Referenced by getPrimitiveNormal(), and PCPrimitive().

6.2.4.7 int pcp::PCPrimitive::shapeMapIdx [private]

Definition at line 34 of file PCPrimitive.h.

Referenced by getShapeMapidx(), and PCPrimitive().

6.2.4.8 string pcp::PCPrimitive::shapeName [private]

Definition at line 30 of file PCPrimitive.h.

Referenced by getShapeName(), and PCPrimitive().

6.2.4.9 bool pcp::PCPrimitive::visualizationFlag [private]

Definition at line 32 of file PCPrimitive.h.

Referenced by getVisualizationFlag(), and PCPrimitive().

6.2.4.10 string pcp::PCPrimitive::visualizationName [private]

Definition at line 31 of file PCPrimitive.h.

Referenced by getVisualizationName(), and PCPrimitive().

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

- PCPrimitive.h
- PCPrimitive.cpp

6.3 vector3d Struct Reference

Public Attributes

- float x
- float y
- float z

6.3.1 Detailed Description

Definition at line 25 of file coneSegmentationServer.cpp.

6.3.2 Member Data Documentation

6.3.2.1 float vector3d::x

Definition at line 26 of file coneSegmentationServer.cpp.

Referenced by getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), ransacConeDetaction(), and ransacCylinderDetaction().

6.3.2.2 float vector3d::y

Definition at line 27 of file coneSegmentationServer.cpp.

Referenced by getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), ransacConeDetaction(), and ransacCylinderDetaction().

6.3.2.3 float vector3d::z

Definition at line 28 of file coneSegmentationServer.cpp.

Referenced by getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), ransacConeDetaction(), and ransacCylinderDetaction().

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

- coneSegmentationServer.cpp
- cylinderSegmentationServer.cpp

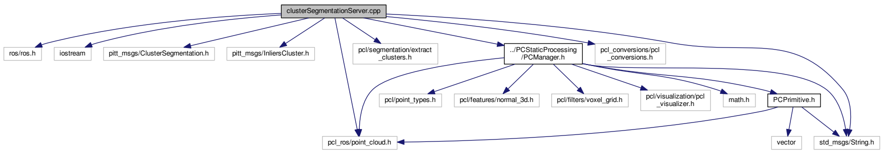
Chapter 7

File Documentation

7.1 clusterSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
#include <iostream>
#include "pitt_msgs/ClusterSegmentation.h"
#include "pitt_msgs/InliersCluster.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/extract_clusters.h>
#include <std_msgs/String.h>
#include <pcl_conversions/pcl_conversions.h>
#include "../PCStaticProcessing/PCManager.h"
```

Include dependency graph for clusterSegmentationServer.cpp:


clusterSegmentationServer.cpp
ros/ros.h
iostream
pitt_msgs/ClusterSegmentation.h
pitt_msgs/InliersCluster.h
pcl/segmentation/extract_clusters.h
../PCStaticProcessing/PCManager.h
pcl_conversions/pcl_conversions.h
pcl/point_types.h
pcl/features/normal_3d.h
pcl/filters/voxel_grid.h
pcl/visualization/pcl_visualizer.h
math.h
PCPrimitive.h
pcl_ros/point_cloud.h
vector
std_msgs/String.h

Functions

- bool clusterize (ClusterSegmentation::Request &req, ClusterSegmentation::Response &res)
- int main (int argc, char ∗∗argv)

Variables

- const float TOLLERANCE_DEFAULT = 0.03f
- const float MIN_CLUSTER_RATE_DEFAULT = 0.01f
- const float MAX_CLUSTER_RATE_DEFAULT = 0.99f
- const float MIN_INPUT_SIZE_DEFAULT = 30.0f

7.1.1 Function Documentation

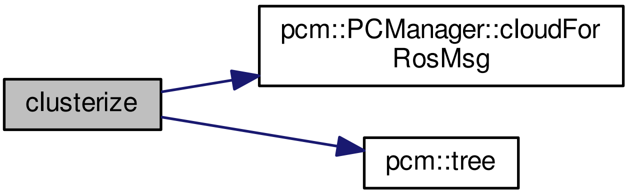
7.1.1.1 bool clusterize (ClusterSegmentation::Request & *req*, ClusterSegmentation::Response & *res*)

Definition at line 33 of file clusterSegmentationServer.cpp.

References pcm::PCManager::cloudForRosMsg(), MAX_CLUSTER_RATE_DEFAULT, MIN_CLUSTER_RATE_DEFAULT, MIN_INPUT_SIZE_DEFAULT, TOLLERANCE_DEFAULT, and pcm::tree().

Referenced by main().

Here is the call graph for this function:

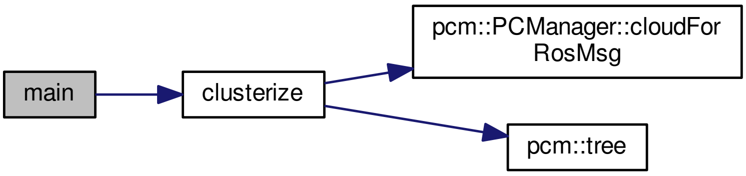


7.1.1.2 int main (int *argc,* char ∗∗ *argv*)

Definition at line 118 of file clusterSegmentationServer.cpp.

References clusterize().

Here is the call graph for this function:



7.1.2 Variable Documentation

7.1.2.1 const float MAX_CLUSTER_RATE_DEFAULT = 0.99f

Definition at line 29 of file clusterSegmentationServer.cpp.

Referenced by clusterize().

7.1.2.2 const float MIN_CLUSTER_RATE_DEFAULT = 0.01f

Definition at line 28 of file clusterSegmentationServer.cpp.

Referenced by clusterize().

7.1.2.3 const float MIN_INPUT_SIZE_DEFAULT = 30.0f

Definition at line 30 of file clusterSegmentationServer.cpp.

Referenced by clusterize().

7.1.2.4 const float TOLLERANCE_DEFAULT = 0.03f

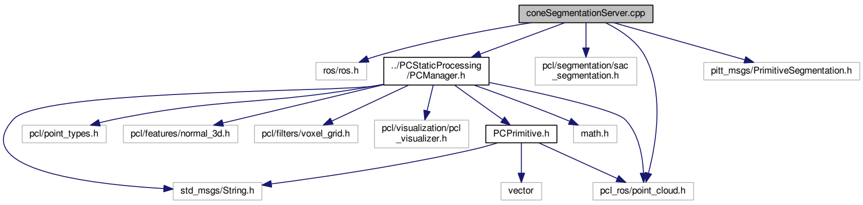
Definition at line 27 of file clusterSegmentationServer.cpp.

Referenced by clusterize().

7.2 coneSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "../PCStaticProcessing/PCManager.h"
#include "pitt_msgs/PrimitiveSegmentation.h"
```

Include dependency graph for coneSegmentationServer.cpp:



Classes

- struct vector3d

Functions

- vector3d getNormalizeAxesDirectionVector (ModelCoefficients::Ptr coefficients)
- vector3d getPointOnAxes (ModelCoefficients::Ptr coefficients, vector3d direction, float t)
- vector3d getVectorBetweenPoints (vector3d p1, vector3d p2)
- bool ransacConeDetaction (PrimitiveSegmentation::Request &req, PrimitiveSegmentation::Response &res)
- int main (int argc, char $**$argv)

Variables

- const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.0006f
- const float DISTANCE_THRESHOLD_DEFAULT = 0.0055f
- const int MAX_ITERATION_DEFAULT = 1000
- const float MIN_RADIUS_LIMIT = 0.001
- const float MAX_RADIUS_LIMIT = 0.500
- const float EPS_ANGLE = 0.4f
- const float MIN_OPENING_ANGLE = 10.0f
- const float MAX_OPENING_ANGLE = 170.0f

- const bool VISUALIZE_RESULT = false
- boost::shared_ptr
  $<$ visualization::PCLVisualizer $>$ vis

7.2.1 Function Documentation

7.2.1.1 vector3d getNormalizeAxesDirectionVector (ModelCoefficients::Ptr *coefficients*)

Definition at line 36 of file coneSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

Referenced by ransacConeDetaction().

7.2.1.2 vector3d getPointOnAxes (ModelCoefficients::Ptr *coefficients,* vector3d *direction,* float *t*)

Definition at line 47 of file coneSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

Referenced by ransacConeDetaction().

7.2.1.3 vector3d getVectorBetweenPoints (vector3d *p1,* vector3d *p2*)

Definition at line 56 of file coneSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

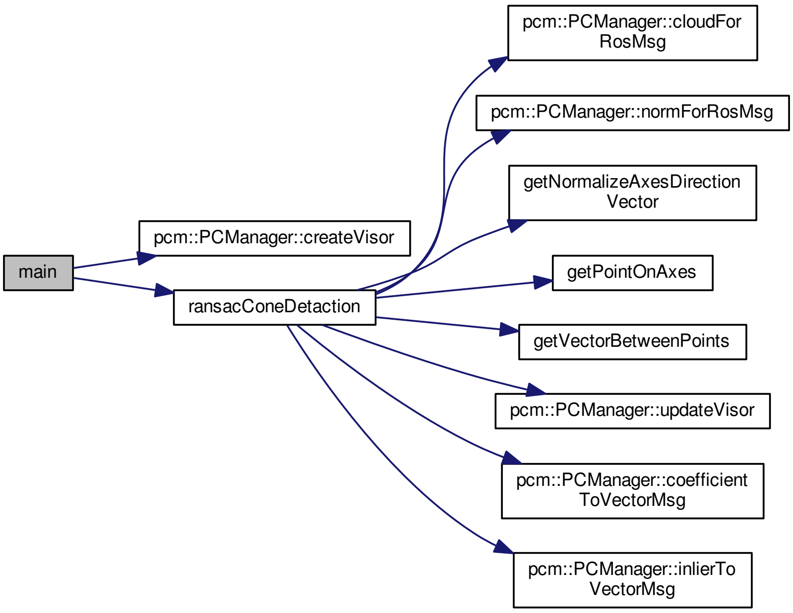
Referenced by ransacConeDetaction().

7.2.1.4 int main (int *argc,* char $**$ *argv*)

Definition at line 217 of file coneSegmentationServer.cpp.

References pcm::PCManager::createVisor(), pcm::PCManager::RANSAC_CONE_FILTER_SERVICE_NAME, ransacConeDetaction(), vis, and VISUALIZE_RESULT.

Here is the call graph for this function:


main
pcm::PCManager::createVisor
ransacConeDetaction
pcm::PCManager::cloudFor RosMsg
pcm::PCManager::normForRosMsg
getNormalizeAxesDirection Vector
getPointOnAxes
getVectorBetweenPoints
pcm::PCManager::updateVisor
pcm::PCManager::coefficient ToVectorMsg
pcm::PCManager::inlierTo VectorMsg

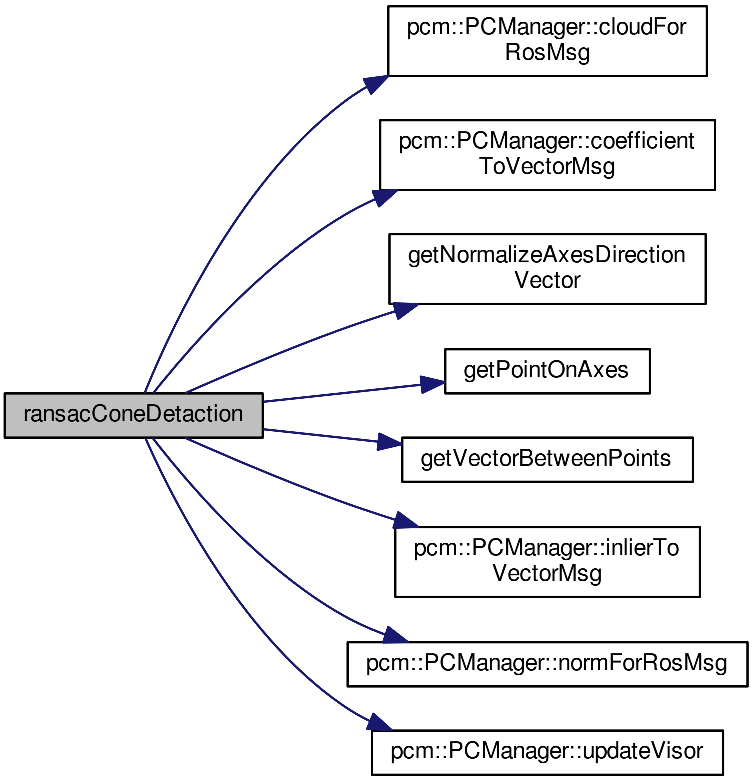
7.2.1.5 bool ransacConeDetaction (PrimitiveSegmentation::Request & *req,* PrimitiveSegmentation::Response & *res*)

Definition at line 65 of file coneSegmentationServer.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), DISTANCE_THRESHOLD_DEFAULT, EPS_ANGLE, getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), pcm::PCManager::inlierToVectorMsg(), MAX_ITERATION_DEFAULT, MAX_OPENING_ANGLE, MAX_RADIUS_LIMIT, MIN_OPENING_ANGLE, MIN_RADIUS_LIMIT, NORMAL_DISTANCE_WEIGHT_DEFAULT, pcm::PCManager::normForRosMsg(), seg, pcm::PCManager::updateVisor(), vis, VISUALIZE_RESULT, vector3d::x, vector3d::y, and vector3d::z.

Referenced by main().

Here is the call graph for this function:


ransacConeDetaction
pcm::PCManager::cloudForRosMsg
pcm::PCManager::coefficientToVectorMsg
getNormalizeAxesDirectionVector
getPointOnAxes
getVectorBetweenPoints
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::normForRosMsg
pcm::PCManager::updateVisor

7.2.2 Variable Documentation

7.2.2.1 const float DISTANCE_THRESHOLD_DEFAULT = 0.0055f

Definition at line 16 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.2 const float EPS_ANGLE = 0.4f

Definition at line 20 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.3 const int MAX_ITERATION_DEFAULT = 1000

Definition at line 17 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.4 const float MAX_OPENING_ANGLE = 170.0f

Definition at line 22 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.5 const float MAX_RADIUS_LIMIT = 0.500

Definition at line 19 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.6 const float MIN_OPENING_ANGLE = 10.0f

Definition at line 21 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.7 const float MIN_RADIUS_LIMIT = 0.001

Definition at line 18 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.8 const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.0006f

Definition at line 15 of file coneSegmentationServer.cpp.

Referenced by ransacConeDetaction().

7.2.2.9 boost::shared_ptr< visualization::PCLVisualizer> vis

Definition at line 33 of file coneSegmentationServer.cpp.

Referenced by main(), and ransacConeDetaction().

7.2.2.10 const bool VISUALIZE_RESULT = false

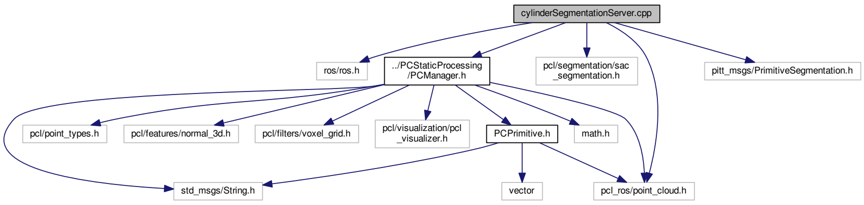
Definition at line 32 of file coneSegmentationServer.cpp.

Referenced by main(), and ransacConeDetaction().

7.3 cylinderSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "../PCStaticProcessing/PCManager.h"
#include "pitt_msgs/PrimitiveSegmentation.h"
```

Include dependency graph for cylinderSegmentationServer.cpp:


cylinderSegmentationServer.cpp
ros/ros.h
../PCStaticProcessing /PCManager.h
pcl/segmentation/sac _segmentation.h
pitt_msgs/PrimitiveSegmentation.h
pcl/point_types.h
pcl/features/normal_3d.h
pcl/filters/voxel_grid.h
pcl/visualization/pcl _visualizer.h
PCPrimitive.h
math.h
std_msgs/String.h
vector
pcl_ros/point_cloud.h

Classes

- struct vector3d

Functions

- vector3d getNormalizeAxesDirectionVector (ModelCoefficients::Ptr coefficients)
- vector3d getPointOnAxes (ModelCoefficients::Ptr coefficients, vector3d direction, float t)
- vector3d getVectorBetweenPoints (vector3d p1, vector3d p2)
- bool ransacCylinderDetaction (PrimitiveSegmentation::Request &req, PrimitiveSegmentation::Response &res)
- int main (int argc, char ∗∗argv)

Variables

- const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f
- const float DISTANCE_THRESHOLD_DEFAULT = 0.008f
- const int MAX_ITERATION_DEFAULT = 1000
- const float MIN_RADIUS_LIMIT = 0.005
- const float MAX_RADIUS_LIMIT = 0.500
- const float EPS_ANGLE = 0.0001f
- const float MIN_OPENING_ANGLE = 50.0f
- const float MAX_OPENING_ANGLE = 180.0f
- const bool VISUALIZE_RESULT = false
- boost::shared_ptr
  < visualization::PCLVisualizer > vis

7.3.1 Function Documentation

7.3.1.1 vector3d getNormalizeAxesDirectionVector (ModelCoefficients::Ptr *coefficients*)

Definition at line 37 of file cylinderSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

Referenced by ransacCylinderDetaction().

7.3.1.2 vector3d getPointOnAxes (ModelCoefficients::Ptr *coefficients,* vector3d *direction,* float *t*)

Definition at line 48 of file cylinderSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

Referenced by ransacCylinderDetaction().

7.3.1.3 vector3d getVectorBetweenPoints (vector3d *p1*, vector3d *p2*)

Definition at line 57 of file cylinderSegmentationServer.cpp.

References vector3d::x, vector3d::y, and vector3d::z.

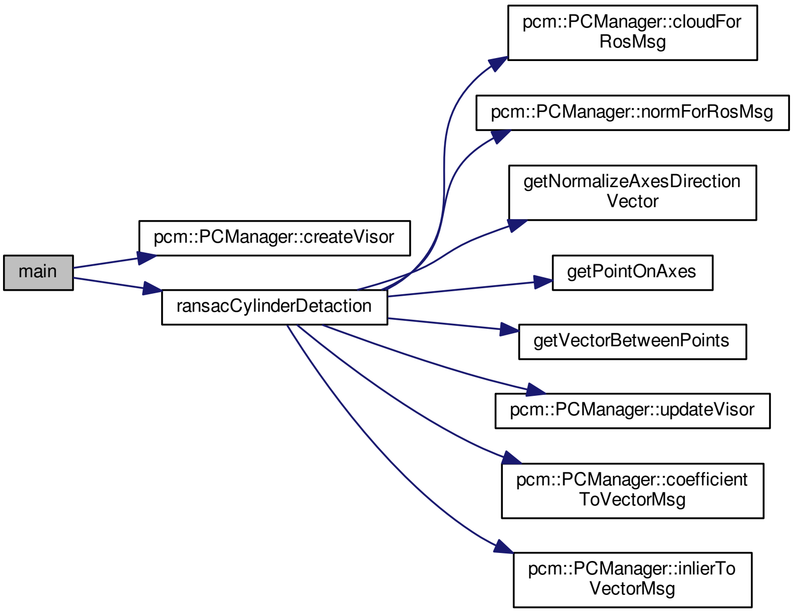
Referenced by ransacCylinderDetaction().

7.3.1.4 int main (int *argc*, char ∗∗ *argv*)

Definition at line 220 of file cylinderSegmentationServer.cpp.

References pcm::PCManager::createVisor(), pcm::PCManager::RANSAC_CYLINDER_FILTER_SERVICE_NAME, ransacCylinderDetaction(), vis, and VISUALIZE_RESULT.

Here is the call graph for this function:


main
pcm::PCManager::createVisor
ransacCylinderDetaction
pcm::PCManager::cloudFor RosMsg
pcm::PCManager::normForRosMsg
getNormalizeAxesDirection Vector
getPointOnAxes
getVectorBetweenPoints
pcm::PCManager::updateVisor
pcm::PCManager::coefficient ToVectorMsg
pcm::PCManager::inlierTo VectorMsg

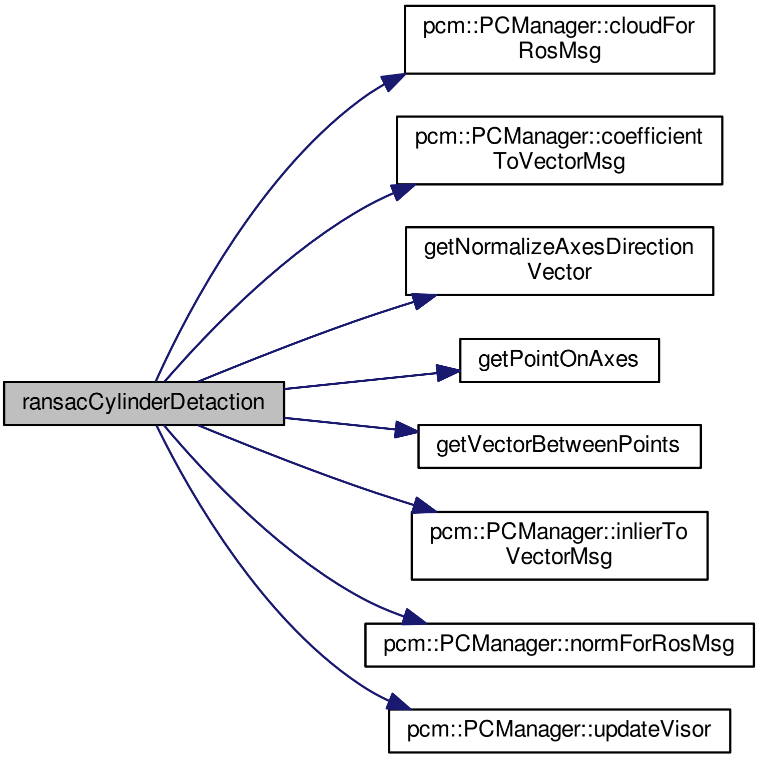
7.3.1.5 bool ransacCylinderDetaction (PrimitiveSegmentation::Request & *req*, PrimitiveSegmentation::Response & *res*)

Definition at line 66 of file cylinderSegmentationServer.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), DISTANCE_THRESHOLD_DEFAULT, EPS_ANGLE, getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), pcm::PCManager::inlierToVectorMsg(), MAX_ITERATION_DEFAULT, MAX_OPENING_ANGLE, MAX_RADIUS_LIMIT, MIN_OPENING_ANGLE, MIN_RADIUS_LIMIT, NORMAL_DISTANCE_WEIGHT_DEFAULT, pcm::PCManager::normForRosMsg(), seg, pcm::PCManager::updateVisor(), vis, VISUALIZE_RESULT, vector3d::x, vector3d::y, and vector3d::z.

Referenced by main().

Here is the call graph for this function:


pcm::PCManager::cloudForRosMsg
pcm::PCManager::coefficientToVectorMsg
getNormalizeAxesDirectionVector
getPointOnAxes
ransacCylinderDetection
getVectorBetweenPoints
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::normForRosMsg
pcm::PCManager::updateVisor

7.3.2 Variable Documentation

7.3.2.1 const float DISTANCE_THRESHOLD_DEFAULT = 0.008f

Definition at line 17 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetection().

7.3.2.2 const float EPS_ANGLE = 0.0001f

Definition at line 21 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetection().

7.3.2.3 const int MAX_ITERATION_DEFAULT = 1000

Definition at line 18 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetection().

7.3.2.4 const float MAX_OPENING_ANGLE = 180.0f

Definition at line 23 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetaction().

7.3.2.5 const float MAX_RADIUS_LIMIT = 0.500

Definition at line 20 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetaction().

7.3.2.6 const float MIN_OPENING_ANGLE = 50.0f

Definition at line 22 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetaction().

7.3.2.7 const float MIN_RADIUS_LIMIT = 0.005

Definition at line 19 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetaction().

7.3.2.8 const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f

Definition at line 16 of file cylinderSegmentationServer.cpp.

Referenced by ransacCylinderDetaction().

7.3.2.9 boost::shared_ptr< visualization::PCLVisualizer > vis

Definition at line 34 of file cylinderSegmentationServer.cpp.

Referenced by main(), and ransacCylinderDetaction().

7.3.2.10 const bool VISUALIZE_RESULT = false

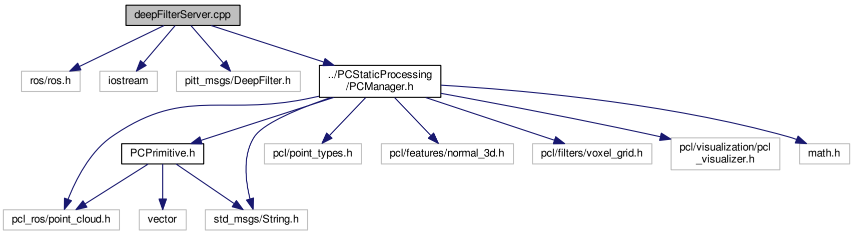
Definition at line 33 of file cylinderSegmentationServer.cpp.

Referenced by main(), and ransacCylinderDetaction().

7.4 deepFilterServer.cpp File Reference

```
#include "ros/ros.h"
#include <iostream>
#include "pitt_msgs/DeepFilter.h"
#include "../PCStaticProcessing/PCManager.h"
```

Include dependency graph for deepFilterServer.cpp:



Functions

- bool deepFiltering (DeepFilter::Request &req, DeepFilter::Response &res)
- int main (int argc, char ∗∗argv)

Variables

- static const float DEPTH_THRESHOULD = 3.000f

7.4.1 Function Documentation

7.4.1.1 bool deepFiltering (DeepFilter::Request & *req,* DeepFilter::Response & *res*)

Definition at line 29 of file deepFilterServer.cpp.

References DEPTH_THRESHOULD.

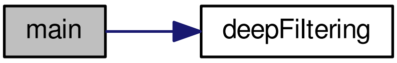
Referenced by main().

7.4.1.2 int main (int *argc,* char ∗∗ *argv*)

Definition at line 64 of file deepFilterServer.cpp.

References deepFiltering().

Here is the call graph for this function:



7.4.2 Variable Documentation

7.4.2.1 const float DEPTH_THRESHOULD = 3.000f [static]

Definition at line 24 of file deepFilterServer.cpp.

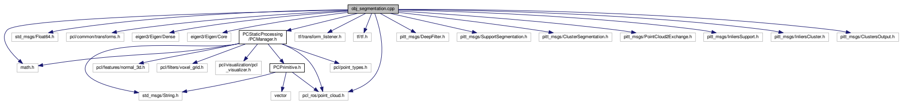
Referenced by deepFiltering().

7.5 introduction.dox File Reference

7.6 obj_segmentation.cpp File Reference

```
#include <pcl_ros/point_cloud.h>
#include <std_msgs/Float64.h>
#include <pcl/common/transforms.h>
#include <eigen3/Eigen/Dense>
#include <eigen3/Eigen/Core>
#include <math.h>
#include <tf/transform_listener.h>
#include <tf/tf.h>
#include "PCStaticProcessing/PCManager.h"
#include "pitt_msgs/DeepFilter.h"
#include "pitt_msgs/SupportSegmentation.h"
#include "pitt_msgs/ClusterSegmentation.h"
#include "pitt_msgs/PointCloud2Exchange.h"
#include "pitt_msgs/InliersSupport.h"
#include "pitt_msgs/InliersCluster.h"
#include "pitt_msgs/ClustersOutput.h"
```

Include dependency graph for obj_segmentation.cpp:



Typedefs

- typedef boost::shared_ptr
  < vector< InliersSupport > > InlierSupportsPtr
- typedef vector< InliersSupport > InlierSupports
- typedef boost::shared_ptr
  < vector< InliersCluster > > InlierClusterPtr
- typedef vector< InliersCluster > InlierClusters

Functions

- bool callDeepFilter (PCLCloudPtr &cloud)
- bool callArmFilter (PCLCloudPtr &cloud)
- InlierSupportsPtr callSupportFilter (PCLCloudPtr inputCloud, PCLNormalPtr normal)
- InlierClusterPtr callClusterSegmentation (PCLCloudPtr cloud)
- void logAvarageCentroid (InliersCluster clusterObject)
- void depthAcquisition (const PointCloud2Ptr &input)
- int main (int argc, char $**$argv)

Variables

- bool SHOW_ORIGINAL_CLOUD = false

- bool SHOW_SUPPORTS = false
- bool SHOW_OBJECT_ON_SUPPORT = false
- bool SHOW_CLUSTERS = true
- static const int MIN_POINT_IN_ORIGINAL_CLOUD = 100
- static const float DEEPTH_THRESHOULD = 1.3f
- static const float MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float MIN_PLANE_PERCENTAGE_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float MAX_VARIANCE_TH_FOR_HORIZONTAL = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int RANSAC_MAX_ITERATION_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float RANSAC_TH_DISTANCE_POINT_SHAPE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float RANSAC_NORMAL_DISTANCE_WEIGHT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static float HORIZONTAL_AXIS [1] = { PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST}
- static const float TABLE_EDGE_OFSET = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CLUSTER_TOLLERANCE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float MAX_CLUSTER_SIZE_RATE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float MIN_CLUSTER_SIZE_RATE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float MIN_CLUSTER_INPUT_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- pcm::PCManager ∗ manager = new pcm::PCManager(false)
- boost::shared_ptr
  < visualization::PCLVisualizer > vis
- Publisher clusterPub
- float avarageCentroid_X = 0
- float avarageCentroid_Y = 0
- float avarageCentroid_Z = 0
- int scanCount = 0
- Eigen::Matrix4f pclTransform
- static string centroidFileLog

7.6.1 Typedef Documentation

7.6.1.1 typedef boost::shared_ptr< vector< InliersCluster> > InlierClusterPtr

Definition at line 39 of file obj_segmentation.cpp.

7.6.1.2 typedef vector< InliersCluster> InlierClusters

Definition at line 40 of file obj_segmentation.cpp.

7.6.1.3 typedef vector< InliersSupport> InlierSupports

Definition at line 38 of file obj_segmentation.cpp.

7.6.1.4 typedef boost::shared_ptr< vector< InliersSupport> > InlierSupportsPtr

Definition at line 37 of file obj_segmentation.cpp.

7.6.2 Function Documentation

7.6.2.1 bool callArmFilter (PCLCloudPtr & *cloud*)

Definition at line 100 of file obj_segmentation.cpp.

Referenced by depthAcquisition().

7.6.2.2 InlierClusterPtr callClusterSegmentation (PCLCloudPtr *cloud*)

Definition at line 154 of file obj_segmentation.cpp.

References CLUSTER_TOLLERANCE, MAX_CLUSTER_SIZE_RATE, MIN_CLUSTER_INPUT_SIZE, and MIN_-CLUSTER_SIZE_RATE.

Referenced by depthAcquisition().

7.6.2.3 bool callDeepFilter (PCLCloudPtr & *cloud*)

Definition at line 74 of file obj_segmentation.cpp.

References DEEPTH_THRESHOULD.

Referenced by depthAcquisition().

7.6.2.4 InlierSupportsPtr callSupportFilter (PCLCloudPtr *inputCloud,* PCLNormalPtr *normal*)

Definition at line 118 of file obj_segmentation.cpp.

References HORIZONTAL_AXIS, MAX_VARIANCE_TH_FOR_HORIZONTAL, MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE, MIN_PLANE_PERCENTAGE_SIZE, RANSAC_MAX_ITERATION_TH, RANSAC_NORMAL_DISTANCE_WEIGHT, RANSAC_TH_DISTANCE_POINT_SHAPE, and TABLE_EDGE_OFSET.

Referenced by depthAcquisition().

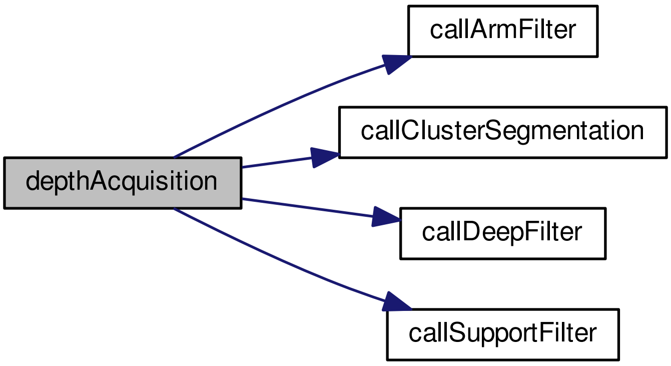
7.6.2.5 void depthAcquisition (const PointCloud2Ptr & *input*)

Definition at line 192 of file obj_segmentation.cpp.

References callArmFilter(), callClusterSegmentation(), callDeepFilter(), callSupportFilter(), centroidFileLog, clusterPub, MIN_POINT_IN_ORIGINAL_CLOUD, pclTransform, SHOW_CLUSTERS, SHOW_OBJECT_ON_SUPPORT, SHOW_ORIGINAL_CLOUD, SHOW_SUPPORTS, and vis.

Referenced by main().

Here is the call graph for this function:



7.6.2.6 void logAvarageCentroid (InliersCluster *clusterObject*)

Definition at line 180 of file obj_segmentation.cpp.

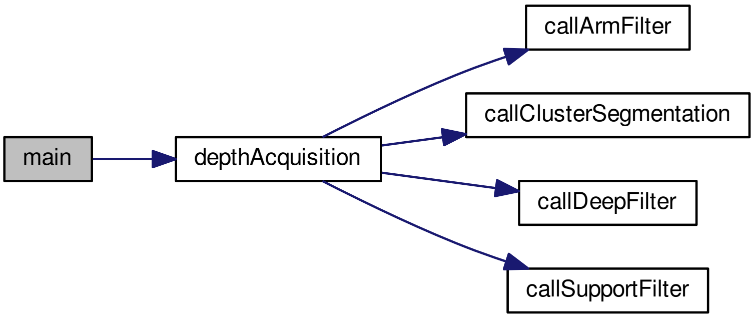
References avarageCentroid_X, avarageCentroid_Y, avarageCentroid_Z, and scanCount.

7.6.2.7 int main (int *argc,* char $**$ *argv*)

Definition at line 280 of file obj_segmentation.cpp.

References centroidFileLog, clusterPub, depthAcquisition(), pclTransform, SHOW_CLUSTERS, SHOW_OBJECT_ON_SUPPORT, SHOW_ORIGINAL_CLOUD, SHOW_SUPPORTS, and vis.

Here is the call graph for this function:



7.6.3 Variable Documentation

7.6.3.1 float avarageCentroid_X = 0

Definition at line 178 of file obj_segmentation.cpp.

Referenced by logAvarageCentroid().

7.6.3.2 float avarageCentroid_Y = 0

Definition at line 178 of file obj_segmentation.cpp.

Referenced by logAvarageCentroid().

7.6.3.3 float avarageCentroid_Z = 0

Definition at line 178 of file obj_segmentation.cpp.

Referenced by logAvarageCentroid().

7.6.3.4 string centroidFileLog [static]

Definition at line 191 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.5 const float CLUSTER_TOLLERANCE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 62 of file obj_segmentation.cpp.

Referenced by callClusterSegmentation().

7.6.3.6 Publisher clusterPub

Definition at line 70 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.7 const float DEEPTH_THRESHOULD = 1.3f [static]

Definition at line 51 of file obj_segmentation.cpp.

Referenced by callDeepFilter().

7.6.3.8 float HORIZONTAL_AXIS[1] = { PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST} [static]

Definition at line 59 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.9 pcm::PCManager* manager = new pcm::PCManager(false)

Definition at line 68 of file obj_segmentation.cpp.

7.6.3.10 const float MAX_CLUSTER_SIZE_RATE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 63 of file obj_segmentation.cpp.

Referenced by callClusterSegmentation().

7.6.3.11 const float MAX_VARIANCE_TH_FOR_HORIZONTAL = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 55 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.12 const float MIN_CLUSTER_INPUT_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 65 of file obj_segmentation.cpp.

Referenced by callClusterSegmentation().

7.6.3.13 const float MIN_CLUSTER_SIZE_RATE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 64 of file obj_segmentation.cpp.

Referenced by callClusterSegmentation().

7.6.3.14 const float MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 53 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.15 const float MIN_PLANE_PERCENTAGE_SIZE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 54 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.16 const int MIN_POINT_IN_ORIGINAL_CLOUD = 100 [static]

Definition at line 49 of file obj_segmentation.cpp.

Referenced by depthAcquisition().

7.6.3.17 Eigen::Matrix4f pclTransform

Definition at line 190 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.18 const int RANSAC_MAX_ITERATION_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 56 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.19 const float RANSAC_NORMAL_DISTANCE_WEIGHT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 58 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.20 const float RANSAC_TH_DISTANCE_POINT_SHAPE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 57 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.21 int scanCount = 0

Definition at line 179 of file obj_segmentation.cpp.

Referenced by logAvarageCentroid().

7.6.3.22 bool SHOW_CLUSTERS = true

Definition at line 47 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.23 bool SHOW_OBJECT_ON_SUPPORT = false

Definition at line 46 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.24 bool SHOW_ORIGINAL_CLOUD = false

Definition at line 44 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.25 bool SHOW_SUPPORTS = false

Definition at line 45 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.6.3.26 const float TABLE_EDGE_OFSET = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 60 of file obj_segmentation.cpp.

Referenced by callSupportFilter().

7.6.3.27 boost::shared_ptr< visualization::PCLVisualizer > vis

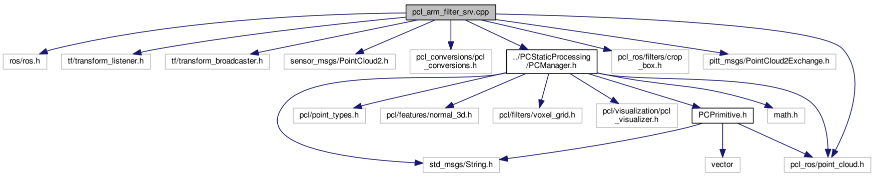
Definition at line 69 of file obj_segmentation.cpp.

Referenced by depthAcquisition(), and main().

7.7 pcl_arm_filter_srv.cpp File Reference

```
#include <ros/ros.h>
#include <tf/transform_listener.h>
#include <tf/transform_broadcaster.h>
#include <sensor_msgs/PointCloud2.h>
#include <pcl_conversions/pcl_conversions.h>
#include <pcl_ros/point_cloud.h>
#include <pcl_ros/filters/crop_box.h>
#include <pitt_msgs/PointCloud2Exchange.h>
#include "../PCStaticProcessing/PCManager.h"
```

Include dependency graph for pcl_arm_filter_srv.cpp:


pcl_arm_filter_srv.cpp
ros/ros.h
tf/transform_listener.h
tf/transform_broadcaster.h
sensor_msgs/PointCloud2.h
pcl_conversions/pcl _conversions.h
../PCStaticProcessing /PCManager.h
pcl_ros/filters/crop _box.h
pitt_msgs/PointCloud2Exchange.h
pcl/point_types.h
pcl/features/normal_3d.h
pcl/filters/voxel_grid.h
pcl/visualization/pcl _visualizer.h
PCPrimitive.h
math.h
std_msgs/String.h
vector
pcl_ros/point_cloud.h

Typedefs

- typedef pcl::PointCloud
  < pcl::PointXYZ > PCLCloud
- typedef pcl::PointCloud
  < pcl::PointXYZ >::Ptr PCLCloudPtr

Functions

- PCLCloudPtr armFiltering (PCLCloudPtr original, Eigen::Vector4f minValues, Eigen::Vector4f maxValues, tf::StampedTransform frame)
- bool filter (PointCloud2ExchangeRequest &input, PointCloud2ExchangeResponse &output)
- int main (int argc, char $**$argv)

Variables

- double roll
- double pitch
- double yaw
- Eigen::Vector3f translation
- Eigen::Vector3f rotation
- Eigen::Affine3f trans = Eigen::Affine3f::Identity()
- CropBox< PointXYZ > cropFilter
- Eigen::Vector4f forearmMinValue
- Eigen::Vector4f elbowMinValue
- Eigen::Vector4f forearmMaxValue
- Eigen::Vector4f elbowMaxValue
- tf::StampedTransform left_lower_forearm_frame
- tf::StampedTransform right_lower_forearm_frame
- tf::StampedTransform left_lower_elbow_frame
- tf::StampedTransform right_lower_elbow_frame
- tf::Quaternion rotQuat
- tf::Matrix3x3 rotMat
- bool tfError = false

7.7.1 Typedef Documentation

7.7.1.1 typedef pcl::PointCloud<pcl::PointXYZ> PCLCloud

Definition at line 24 of file pcl_arm_filter_srv.cpp.

7.7.1.2 typedef pcl::PointCloud<pcl::PointXYZ>::Ptr PCLCloudPtr

Definition at line 25 of file pcl_arm_filter_srv.cpp.

7.7.2 Function Documentation

7.7.2.1 PCLCloudPtr armFiltering (PCLCloudPtr *original,* Eigen::Vector4f *minValues,* Eigen::Vector4f *maxValues,* tf::StampedTransform *frame*)

Definition at line 53 of file pcl_arm_filter_srv.cpp.

References cropFilter, pitch, roll, rotation, rotMat, rotQuat, trans, translation, and yaw.

Referenced by filter().

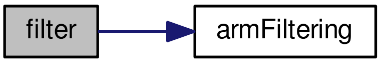
7.7.2.2 bool filter (PointCloud2ExchangeRequest & *input,* PointCloud2ExchangeResponse & *output*)

Definition at line 86 of file pcl_arm_filter_srv.cpp.

References armFiltering(), elbowMaxValue, elbowMinValue, forearmMaxValue, forearmMinValue, left_lower_elbow-_frame, left_lower_forearm_frame, right_lower_elbow_frame, right_lower_forearm_frame, and tfError.

Referenced by main().

Here is the call graph for this function:

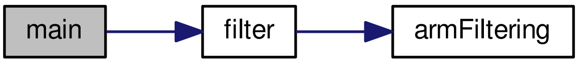


7.7.2.3 int main (int *argc,* char ** *argv*)

Definition at line 130 of file pcl_arm_filter_srv.cpp.

References elbowMaxValue, elbowMinValue, filter(), forearmMaxValue, forearmMinValue, left_lower_elbow_frame, left_lower_forearm_frame, right_lower_elbow_frame, right_lower_forearm_frame, and tfError.

Here is the call graph for this function:



7.7.3 Variable Documentation

7.7.3.1 CropBox< PointXYZ > cropFilter

Definition at line 34 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.2 Eigen::Vector4f elbowMaxValue

Definition at line 42 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.3 Eigen::Vector4f elbowMinValue

Definition at line 38 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.4 Eigen::Vector4f forearmMaxValue

Definition at line 41 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.5 Eigen::Vector4f forearmMinValue

Definition at line 37 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.6 tf::StampedTransform left_lower_elbow_frame

Definition at line 46 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.7 tf::StampedTransform left_lower_forearm_frame

Definition at line 46 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.8 double pitch

Definition at line 27 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.9 tf::StampedTransform right_lower_elbow_frame

Definition at line 46 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.10 tf::StampedTransform right_lower_forearm_frame

Definition at line 46 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.11 double roll

Definition at line 27 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.12 Eigen::Vector3f rotation

Definition at line 31 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.13 tf::Matrix3x3 rotMat

Definition at line 48 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.14 tf::Quaternion rotQuat

Definition at line 47 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.15 bool tfError = false

Definition at line 49 of file pcl_arm_filter_srv.cpp.

Referenced by filter(), and main().

7.7.3.16 Eigen::Affine3f trans = Eigen::Affine3f::Identity()

Definition at line 32 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.17 Eigen::Vector3f translation

Definition at line 30 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.7.3.18 double yaw

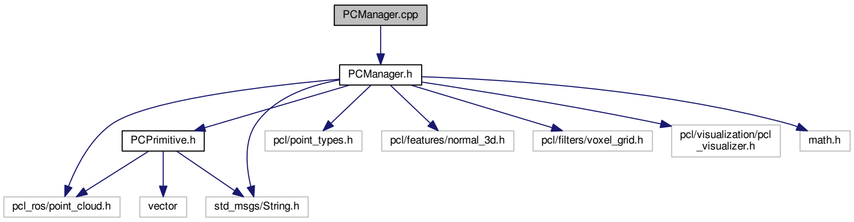
Definition at line 27 of file pcl_arm_filter_srv.cpp.

Referenced by armFiltering().

7.8 PCManager.cpp File Reference

```
#include "PCManager.h"
```

Include dependency graph for PCManager.cpp:



Namespaces

- pcm

Functions

- static search::KdTree
  < PointXYZ >::Ptr pcm::tree (new search::KdTree< PointXYZ >())

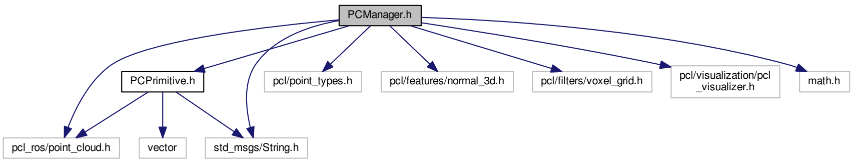
Variables

- static NormalEstimation
  < PointXYZ, Normal > pcm::ne
- static VoxelGrid< PointXYZ > pcm::sor

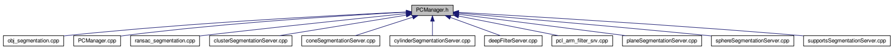
7.9 PCManager.h File Reference

```
#include <pcl_ros/point_cloud.h>
#include <pcl/point_types.h>
#include <pcl/features/normal_3d.h>
#include <pcl/filters/voxel_grid.h>
#include <pcl/visualization/pcl_visualizer.h>
#include <std_msgs/String.h>
#include <math.h>
#include "PCPrimitive.h"
```

Include dependency graph for PCManager.h:



This graph shows which files directly or indirectly include this file:


PCManager.h
obj_segmentation.cpp
PCManager.cpp
ransac_segmentation.cpp
clusterSegmentationServer.cpp
coneSegmentationServer.cpp
cylinderSegmentationServer.cpp
deepFilterServer.cpp
pcl_arm_filter_srv.cpp
planeSegmentationServer.cpp
sphereSegmentationServer.cpp
supportsSegmentationServer.cpp

Classes

- class pcm::PCManager

Namespaces

- pcm

Typedefs

- typedef boost::shared_ptr < pcp::PCPrimitive > PCPrimitivePtr
- typedef boost::shared_ptr < visualization::PCLVisualizer > PCLVisualizer

7.9.1 Typedef Documentation

7.9.1.1 typedef boost::shared_ptr< visualization::PCLVisualizer> PCLVisualizer

Definition at line 23 of file PCManager.h.

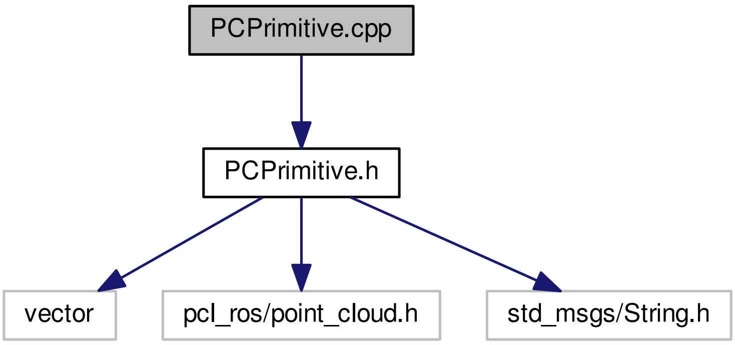
7.9.1.2 typedef boost::shared_ptr< pcp::PCPrimitive> PCPrimitivePtr

Definition at line 22 of file PCManager.h.

7.10 PCPrimitive.cpp File Reference

```
#include "PCPrimitive.h"
```

Include dependency graph for PCPrimitive.cpp:



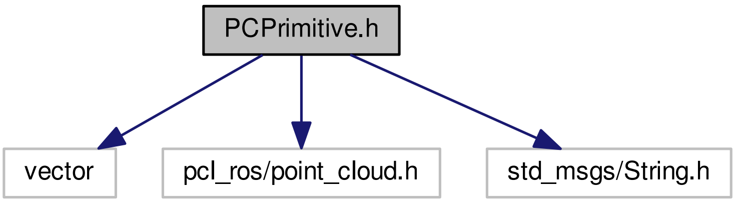
Namespaces

- pcp

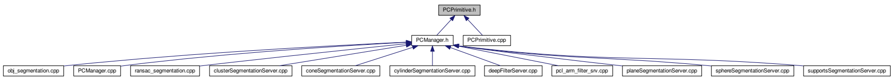
7.11 PCPrimitive.h File Reference

```
#include <vector>
#include <pcl_ros/point_cloud.h>
#include <std_msgs/String.h>
```

Include dependency graph for PCPrimitive.h:



This graph shows which files directly or indirectly include this file:


PCPrimitive.h
PCManager.h
PCPrimitive.cpp
obj_segmentation.cpp
PCManager.cpp
ransac_segmentation.cpp
clusterSegmentationServer.cpp
coneSegmentationServer.cpp
cylinderSegmentationServer.cpp
deepFilterServer.cpp
pcl_arm_filter_srv.cpp
planeSegmentationServer.cpp
sphereSegmentationServer.cpp
supportsSegmentationServer.cpp

Classes

- class pcp::PCPrimitive

Namespaces

- pcp

Typedefs

- typedef std::vector< int > PrimitiveIdx
- typedef boost::shared_ptr
  < std::vector< int > > PrimitiveIdxPtr
- typedef pcl::PointCloud
  < pcl::PointXYZ > PCLCloud
- typedef pcl::PointCloud
  < pcl::PointXYZ >::Ptr PCLCloudPtr
- typedef pcl::PointCloud
  < pcl::Normal > PCLNormal
- typedef pcl::PointCloud
  < pcl::Normal >::Ptr PCLNormalPtr

7.11.1 Typedef Documentation

7.11.1.1 typedef pcl::PointCloud< pcl::PointXYZ> PCLCloud

Definition at line 20 of file PCPrimitive.h.

7.11.1.2 typedef pcl::PointCloud< pcl::PointXYZ>::Ptr PCLCloudPtr

Definition at line 21 of file PCPrimitive.h.

7.11.1.3 typedef pcl::PointCloud< pcl::Normal> PCLNormal

Definition at line 22 of file PCPrimitive.h.

7.11.1.4 typedef pcl::PointCloud< pcl::Normal>::Ptr PCLNormalPtr

Definition at line 23 of file PCPrimitive.h.

7.11.1.5 typedef std::vector< int> PrimitiveIdx

Definition at line 18 of file PCPrimitive.h.

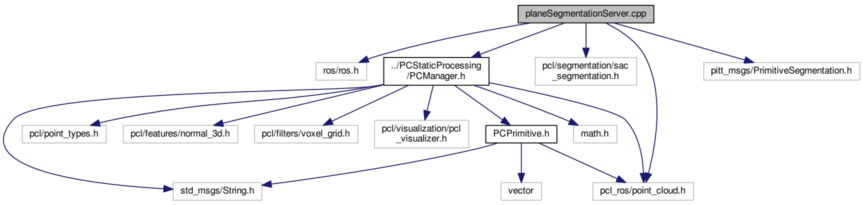
7.11.1.6 typedef boost::shared_ptr< std::vector< int> > PrimitiveIdxPtr

Definition at line 19 of file PCPrimitive.h.

7.12 planeSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "../PCStaticProcessing/PCManager.h"
#include "pitt_msgs/PrimitiveSegmentation.h"
```

Include dependency graph for planeSegmentationServer.cpp:


planeSegmentationServer.cpp
ros/ros.h
../PCStaticProcessing /PCManager.h
pcl/segmentation/sac _segmentation.h
pitt_msgs/PrimitiveSegmentation.h
pcl/point_types.h
pcl/features/normal_3d.h
pcl/filters/voxel_grid.h
pcl/visualization/pcl _visualizer.h
PCPrimitive.h
math.h
std_msgs/String.h
vector
pcl_ros/point_cloud.h

Functions

- bool ransacPlaneDetaction (PrimitiveSegmentation::Request &req, PrimitiveSegmentation::Response &res)
- int main (int argc, char ∗∗argv)

Variables

- const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f
- const float DISTANCE_THRESHOLD_DEFAULT = 0.007f
- const int MAX_ITERATION_DEFAULT = 1000
- const float MIN_RADIUS_LIMIT = -1.0f
- const float MAX_RADIUS_LIMIT = -1.0f
- const float EPS_ANGLE = 0.0f
- const float MIN_OPENING_ANGLE = 0.0f
- const float MAX_OPENING_ANGLE = 10.0f

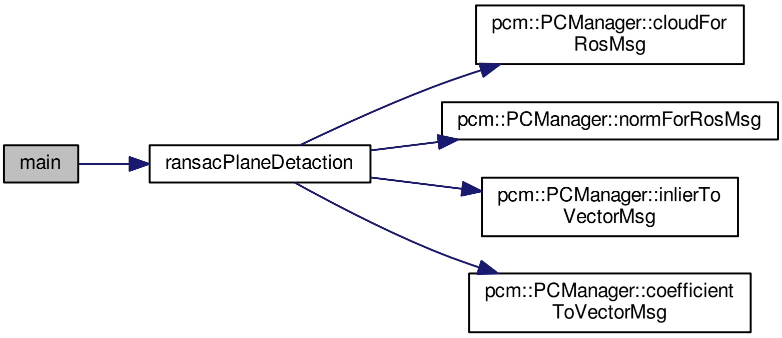
7.12.1 Function Documentation

7.12.1.1 int main (int *argc,* char ∗∗ *argv*)

Definition at line 97 of file planeSegmentationServer.cpp.

References pcm::PCManager::RANSAC_PLANE_FILTER_SERVICE_NAME, and ransacPlaneDetaction().

Here is the call graph for this function:


main
ransacPlaneDetaction
pcm::PCManager::cloudForRosMsg
pcm::PCManager::normForRosMsg
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::coefficientToVectorMsg

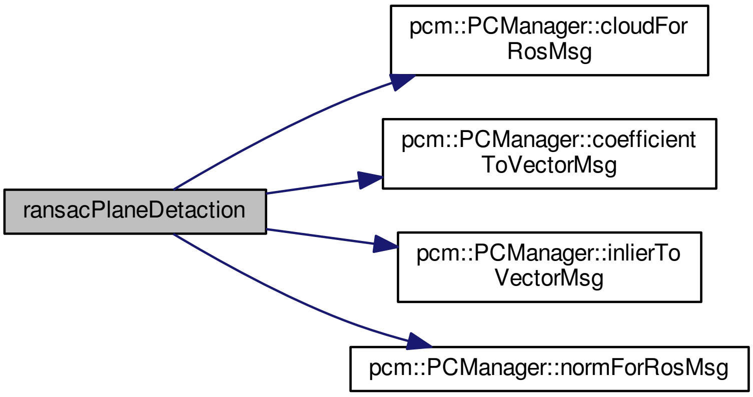
7.12.1.2 bool ransacPlaneDetaction (PrimitiveSegmentation::Request & *req,* PrimitiveSegmentation::Response & *res*)

Definition at line 25 of file planeSegmentationServer.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), DISTANCE_THRESHOLD_DEFAULT, EPS_ANGLE, pcm::PCManager::inlierToVectorMsg(), MAX_ITERATION_DEFAULT, MAX_OPENING_ANGLE, MAX_RADIUS_LIMIT, MIN_OPENING_ANGLE, MIN_RADIUS_LIMIT, NORMAL_DISTANCE_WEIGHT_DEFAULT, pcm::PCManager::normForRosMsg(), and seg.

Referenced by main().

Here is the call graph for this function:


ransacPlaneDetaction
pcm::PCManager::cloudForRosMsg
pcm::PCManager::coefficientToVectorMsg
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::normForRosMsg

7.12.2 Variable Documentation

7.12.2.1 const float DISTANCE_THRESHOLD_DEFAULT = 0.007f

Definition at line 16 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.2 const float EPS_ANGLE = 0.0f

Definition at line 20 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.3 const int MAX_ITERATION_DEFAULT = 1000

Definition at line 17 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.4 const float MAX_OPENING_ANGLE = 10.0f

Definition at line 22 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.5 const float MAX_RADIUS_LIMIT = -1.0f

Definition at line 19 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.6 const float MIN_OPENING_ANGLE = 0.0f

Definition at line 21 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.7 const float MIN_RADIUS_LIMIT = -1.0f

Definition at line 18 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.12.2.8 const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f

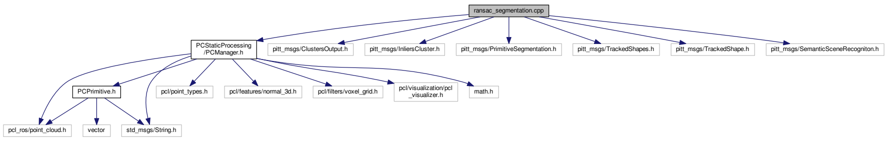
Definition at line 15 of file planeSegmentationServer.cpp.

Referenced by ransacPlaneDetaction().

7.13 ransac_segmentation.cpp File Reference

```
#include "PCStaticProcessing/PCManager.h"
#include "pitt_msgs/ClustersOutput.h"
#include "pitt_msgs/InliersCluster.h"
#include "pitt_msgs/PrimitiveSegmentation.h"
#include "pitt_msgs/TrackedShapes.h"
#include "pitt_msgs/TrackedShape.h"
#include "pitt_msgs/SemanticSceneRecogniton.h"
```

Include dependency graph for ransac_segmentation.cpp:



Typedefs

- typedef vector< InliersCluster > InliersClusters
- typedef boost::shared_ptr
  < InliersClusters > InliersClustersPtr
- typedef boost::shared_ptr
  < PrimitiveSegmentation > PrimitiveSegmentationPtr

Functions

- bool callRansacSphereSegmentation (PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr &out)
- void printSphereInfo (PrimitiveSegmentationPtr info, int idx)
- bool callRansacCylinderSegmentation (PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr &out)
- void printCylinderInfo (PrimitiveSegmentationPtr info, int idx)
- bool callRansacConeSegmentation (PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr &out)
- void printConeInfo (PrimitiveSegmentationPtr info, int idx)
- bool callRansacPlaneSegmentation (PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr &out)
- void printPlaneInfo (PrimitiveSegmentationPtr info, int idx)
- string returnPrimitiveNameFromTag (int primitiveTag)
- bool callSemanticSceneRecognitionServer (TrackedShapes::Ptr outShapes)
- void clustersAcquisition (const ClustersOutputConstPtr &clusterObj)
- int main (int argc, char $**$argv)

Variables

- boost::shared_ptr
  < visualization::PCLVisualizer > vis
- Publisher pub
- static const float SPHERE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float SPHERE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float SPHERE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float SPHERE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int SPHERE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float SPHERE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float SPHERE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST

- static const float SPHERE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int SPHERE_MIN_INLIERS = 40
- static const float CYLINDER_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int CYLINDER_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CYLINDER_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int CYLINDER_MIN_INLIERS = 40
- static const float CONE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int CONE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float CONE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int CONE_MIN_INLIERS = 40
- static const float PLANE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int PLANE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const float PLANE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST
- static const int PLANE_MIN_INLIERS = 40
- static const bool SHOW_PRIMITIVE = true
- static const float CONE_TO_CYLINDER_PRIORITY = 0.9f
- static const int TXT_UNKNOWN_SHAPE_TAG = 0
- static const int TXT_PLANE_SHAPE_TAG = 1
- static const int TXT_SPHERE_SHAPE_TAG = 2
- static const int TXT_CONE_SHAPE_TAG = 3
- static const int TXT_CYLINDER_SHAPE_TAG = 4
- string centroidFileLog

7.13.1 Typedef Documentation

7.13.1.1 typedef vector< InliersCluster> InliersClusters

Definition at line 15 of file ransac_segmentation.cpp.

7.13.1.2 typedef boost::shared_ptr< InliersClusters> InliersClustersPtr

Definition at line 16 of file ransac_segmentation.cpp.

7.13.1.3 typedef boost::shared_ptr< PrimitiveSegmentation> PrimitiveSegmentationPtr

Definition at line 17 of file ransac_segmentation.cpp.

7.13.2 Function Documentation

7.13.2.1 bool callRansacConeSegmentation (PCLCloudPtr *cloud,* PCLNormalPtr *norm,* PrimitiveSegmentationPtr & *out*)

Definition at line 161 of file ransac_segmentation.cpp.

References CONE_DISTANZE_TH, CONE_EPS_ANGLE_TH, CONE_MAX_ITERATION_LIMIT, CONE_MAX_O-PENING_ANGLE_DEGREE, CONE_MAX_RADIUS_LIMIT, CONE_MIN_INLIERS, CONE_MIN_OPENING_ANG-LE_DEGREE, CONE_MIN_RADIUS_LIMIT, and CONE_NORMAL_DISTANCE_WEIGTH.

Referenced by clustersAcquisition().

7.13.2.2 bool callRansacCylinderSegmentation (PCLCloudPtr *cloud,* PCLNormalPtr *norm,* PrimitiveSegmentationPtr & *out*)

Definition at line 115 of file ransac_segmentation.cpp.

References CYLINDER_DISTANZE_TH, CYLINDER_EPS_ANGLE_TH, CYLINDER_MAX_ITERATION_LIMIT, C-YLINDER_MAX_OPENING_ANGLE_DEGREE, CYLINDER_MAX_RADIUS_LIMIT, CYLINDER_MIN_INLIERS, C-YLINDER_MIN_OPENING_ANGLE_DEGREE, CYLINDER_MIN_RADIUS_LIMIT, and CYLINDER_NORMAL_DI-STANCE_WEIGTH.

Referenced by clustersAcquisition().

7.13.2.3 bool callRansacPlaneSegmentation (PCLCloudPtr *cloud,* PCLNormalPtr *norm,* PrimitiveSegmentationPtr & *out*)

Definition at line 207 of file ransac_segmentation.cpp.

References PLANE_DISTANZE_TH, PLANE_EPS_ANGLE_TH, PLANE_MAX_ITERATION_LIMIT, PLANE_MAX-_OPENING_ANGLE_DEGREE, PLANE_MAX_RADIUS_LIMIT, PLANE_MIN_INLIERS, PLANE_MIN_OPENING_-ANGLE_DEGREE, PLANE_MIN_RADIUS_LIMIT, and PLANE_NORMAL_DISTANCE_WEIGTH.

Referenced by clustersAcquisition().

7.13.2.4 bool callRansacSphereSegmentation (PCLCloudPtr *cloud,* PCLNormalPtr *norm,* PrimitiveSegmentationPtr & *out*)

Definition at line 72 of file ransac_segmentation.cpp.

References SPHERE_DISTANZE_TH, SPHERE_EPS_ANGLE_TH, SPHERE_MAX_ITERATION_LIMIT, SPHERE_MAX_OPENING_ANGLE_DEGREE, SPHERE_MAX_RADIUS_LIMIT, SPHERE_MIN_INLIERS, SPHERE_MIN_OPENING_ANGLE_DEGREE, SPHERE_MIN_RADIUS_LIMIT, and SPHERE_NORMAL_DISTANCE_WEIGTH.

Referenced by clustersAcquisition().

7.13.2.5 bool callSemanticSceneRecognitionServer (TrackedShapes::Ptr *outShapes*)

Definition at line 262 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition().

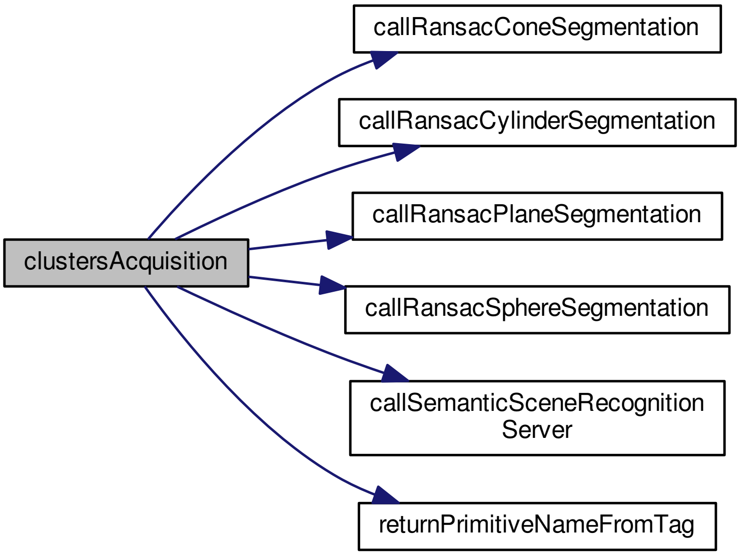
7.13.2.6 void clustersAcquisition (const ClustersOutputConstPtr & *clusterObj*)

Definition at line 284 of file ransac_segmentation.cpp.

References callRansacConeSegmentation(), callRansacCylinderSegmentation(), callRansacPlaneSegmentation(), callRansacSphereSegmentation(), callSemanticSceneRecognitionServer(), CONE_TO_CYLINDER_PRIORITY, pub, returnPrimitiveNameFromTag(), SHOW_PRIMITIVE, TXT_CONE_SHAPE_TAG, TXT_CYLINDER_SHAPE_TAG, TXT_PLANE_SHAPE_TAG, TXT_SPHERE_SHAPE_TAG, TXT_UNKNOWN_SHAPE_TAG, and vis.

Referenced by main().

Here is the call graph for this function:

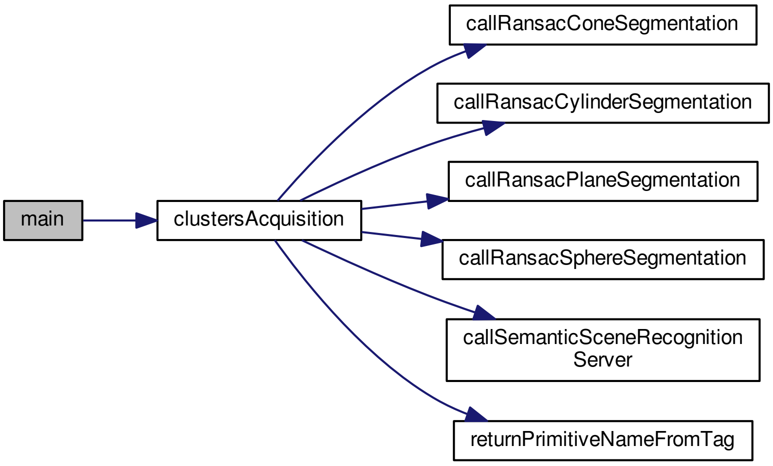

clustersAcquisition
callRansacConeSegmentation
callRansacCylinderSegmentation
callRansacPlaneSegmentation
callRansacSphereSegmentation
callSemanticSceneRecognition Server
returnPrimitiveNameFromTag

7.13.2.7 int main (int *argc,* char ∗∗ *argv*)

Definition at line 402 of file ransac_segmentation.cpp.

References centroidFileLog, clustersAcquisition(), pub, SHOW_PRIMITIVE, and vis.

Here is the call graph for this function:


main
clustersAcquisition
callRansacConeSegmentation
callRansacCylinderSegmentation
callRansacPlaneSegmentation
callRansacSphereSegmentation
callSemanticSceneRecognition Server
returnPrimitiveNameFromTag

7.13.2.8 void printConeInfo (PrimitiveSegmentationPtr *info*, int *idx*)

Definition at line 193 of file ransac_segmentation.cpp.

7.13.2.9 void printCylinderInfo (PrimitiveSegmentationPtr *info*, int *idx*)

Definition at line 148 of file ransac_segmentation.cpp.

7.13.2.10 void printPlaneInfo (PrimitiveSegmentationPtr *info*, int *idx*)

Definition at line 241 of file ransac_segmentation.cpp.

7.13.2.11 void printSphereInfo (PrimitiveSegmentationPtr *info*, int *idx*)

Definition at line 105 of file ransac_segmentation.cpp.

7.13.2.12 string returnPrimitiveNameFromTag (int *primitiveTag*)

Definition at line 250 of file ransac_segmentation.cpp.

References TXT_CONE_SHAPE_TAG, TXT_CYLINDER_SHAPE_TAG, TXT_PLANE_SHAPE_TAG, TXT_SPHERE_SHAPE_TAG, and TXT_UNKNOWN_SHAPE_TAG.

Referenced by clustersAcquisition().

7.13.3 Variable Documentation

7.13.3.1 string centroidFileLog

Definition at line 283 of file ransac_segmentation.cpp.

Referenced by main().

7.13.3.2 const float CONE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 44 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.3 const float CONE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 48 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.4 const int CONE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 47 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.5 const float CONE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 50 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.6 const float CONE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 46 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.7 const int CONE_MIN_INLIERS = 40 [static]

Definition at line 51 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.8 const float CONE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 49 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.9 const float CONE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 45 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.10 const float CONE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 43 of file ransac_segmentation.cpp.

Referenced by callRansacConeSegmentation().

7.13.3.11 const float CONE_TO_CYLINDER_PRIORITY = 0.9f [static]

Definition at line 64 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition().

7.13.3.12 const float CYLINDER_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 34 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.13 const float CYLINDER_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 38 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.14 const int CYLINDER_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 37 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.15 const float CYLINDER_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 40 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.16 const float CYLINDER_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 36 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.17 const int CYLINDER_MIN_INLIERS = 40 [static]

Definition at line 41 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.18 const float CYLINDER_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 39 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.19 const float CYLINDER_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 35 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.20 const float CYLINDER_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 33 of file ransac_segmentation.cpp.

Referenced by callRansacCylinderSegmentation().

7.13.3.21 const float PLANE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 54 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.22 const float PLANE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 58 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.23 const int PLANE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 57 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.24 const float PLANE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 60 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.25 const float PLANE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 56 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.26 const int PLANE_MIN_INLIERS = 40 [static]

Definition at line 61 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.27 const float PLANE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 59 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.28 const float PLANE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 55 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.29 const float PLANE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 53 of file ransac_segmentation.cpp.

Referenced by callRansacPlaneSegmentation().

7.13.3.30 Publisher pub

Definition at line 20 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and main().

7.13.3.31 const bool SHOW_PRIMITIVE = true [static]

Definition at line 63 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and main().

7.13.3.32 const float SPHERE_DISTANZE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 24 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.33 const float SPHERE_EPS_ANGLE_TH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 28 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.34 const int SPHERE_MAX_ITERATION_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 27 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.35 const float SPHERE_MAX_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 30 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.36 const float SPHERE_MAX_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 26 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.37 const int SPHERE_MIN_INLIERS = 40 [static]

Definition at line 31 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.38 const float SPHERE_MIN_OPENING_ANGLE_DEGREE = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 29 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.39 const float SPHERE_MIN_RADIUS_LIMIT = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 25 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.40 const float SPHERE_NORMAL_DISTANCE_WEIGTH = PCManager::DEFAULT_SERVICE_PARAMETER_REQUEST [static]

Definition at line 23 of file ransac_segmentation.cpp.

Referenced by callRansacSphereSegmentation().

7.13.3.41 const int TXT_CONE_SHAPE_TAG = 3 [static]

Definition at line 68 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and returnPrimitiveNameFromTag().

7.13.3.42 const int TXT_CYLINDER_SHAPE_TAG = 4 [static]

Definition at line 69 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and returnPrimitiveNameFromTag().

7.13.3.43 const int TXT_PLANE_SHAPE_TAG = 1 [static]

Definition at line 66 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and returnPrimitiveNameFromTag().

7.13.3.44 const int TXT_SPHERE_SHAPE_TAG = 2 [static]

Definition at line 67 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and returnPrimitiveNameFromTag().

7.13.3.45 const int TXT_UNKNOWN_SHAPE_TAG = 0 [static]

Definition at line 65 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and returnPrimitiveNameFromTag().

7.13.3.46 boost::shared_ptr< visualization::PCLVisualizer > vis

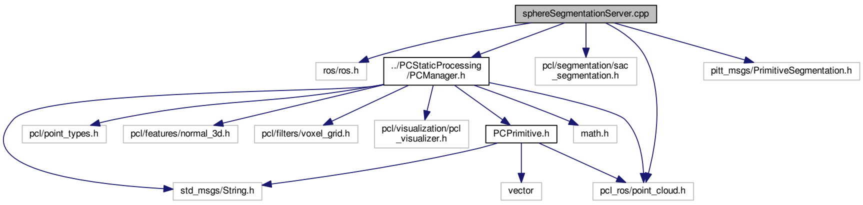
Definition at line 19 of file ransac_segmentation.cpp.

Referenced by clustersAcquisition(), and main().

7.14 sphereSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "../PCStaticProcessing/PCManager.h"
#include "pitt_msgs/PrimitiveSegmentation.h"
```

Include dependency graph for sphereSegmentationServer.cpp:



Functions

- bool ransacSphereDetaction (PrimitiveSegmentation::Request &req, PrimitiveSegmentation::Response &res)
- int main (int argc, char ∗∗argv)

Variables

- const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f
- const float DISTANCE_THRESHOLD_DEFAULT = 0.007f
- const int MAX_ITERATION_DEFAULT = 1000
- const float MIN_RADIUS_LIMIT = 0.005
- const float MAX_RADIUS_LIMIT = 0.500
- const float EPS_ANGLE = 0.0f
- const float MIN_OPENING_ANGLE = 100.0f
- const float MAX_OPENING_ANGLE = 180.0f

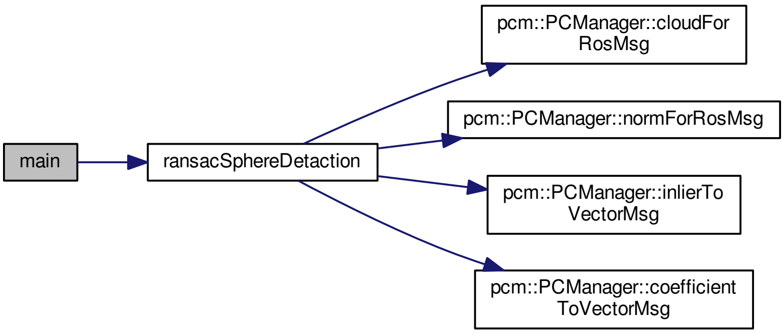
7.14.1 Function Documentation

7.14.1.1 int main (int *argc,* char ∗∗ *argv*)

Definition at line 113 of file sphereSegmentationServer.cpp.

References pcm::PCManager::RANSAC_SPHERE_FILTER_SERVICE_NAME, and ransacSphereDetaction().

Here is the call graph for this function:


main
ransacSphereDetaction
pcm::PCManager::cloudForRosMsg
pcm::PCManager::normForRosMsg
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::coefficientToVectorMsg

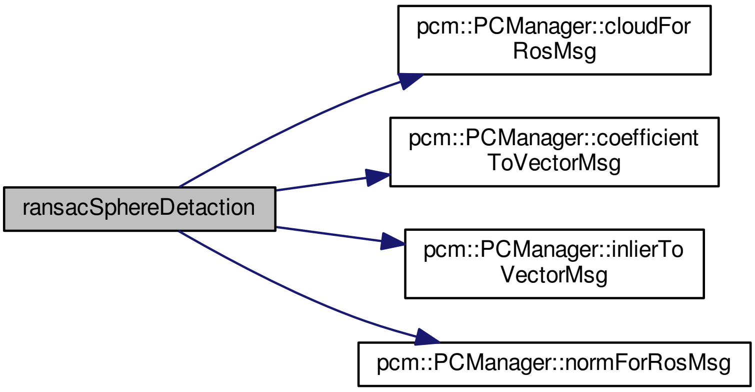
7.14.1.2 bool ransacSphereDetaction (PrimitiveSegmentation::Request & *req*, PrimitiveSegmentation::Response & *res*)

Definition at line 26 of file sphereSegmentationServer.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), DISTANCE_THRESHOLD_DEFAULT, EPS_ANGLE, pcm::PCManager::inlierToVectorMsg(), MAX_ITERATION_DEFAULT, MAX_OPENING_ANGLE, MAX_RADIUS_LIMIT, MIN_OPENING_ANGLE, MIN_RADIUS_LIMIT, NORMAL_DISTANCE_WEIGHT_DEFAULT, pcm::PCManager::normForRosMsg(), and seg.

Referenced by main().

Here is the call graph for this function:


ransacSphereDetaction
pcm::PCManager::cloudForRosMsg
pcm::PCManager::coefficientToVectorMsg
pcm::PCManager::inlierToVectorMsg
pcm::PCManager::normForRosMsg

7.14.2 Variable Documentation

7.14.2.1 const float DISTANCE_THRESHOLD_DEFAULT = 0.007f

Definition at line 17 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.2 const float EPS_ANGLE = 0.0f

Definition at line 21 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.3 const int MAX_ITERATION_DEFAULT = 1000

Definition at line 18 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.4 const float MAX_OPENING_ANGLE = 180.0f

Definition at line 23 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.5 const float MAX_RADIUS_LIMIT = 0.500

Definition at line 20 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.6 const float MIN_OPENING_ANGLE = 100.0f

Definition at line 22 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.7 const float MIN_RADIUS_LIMIT = 0.005

Definition at line 19 of file sphereSegmentationServer.cpp.

Referenced by ransacSphereDetaction().

7.14.2.8 const float NORMAL_DISTANCE_WEIGHT_DEFAULT = 0.001f

Definition at line 16 of file sphereSegmentationServer.cpp.

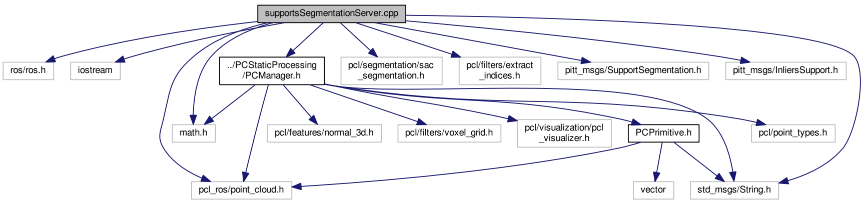
Referenced by ransacSphereDetaction().

7.15 supportsSegmentationServer.cpp File Reference

```
#include "ros/ros.h"
```

```
#include <iostream>
#include "math.h"
#include <pcl_ros/point_cloud.h>
#include <std_msgs/String.h>
#include <pcl/segmentation/sac_segmentation.h>
#include <pcl/filters/extract_indices.h>
#include "pitt_msgs/SupportSegmentation.h"
#include "pitt_msgs/InliersSupport.h"
#include "../PCStaticProcessing/PCManager.h"
```

Include dependency graph for supportsSegmentationServer.cpp:



Macros

- #define LESS_INF -9999999.0
- #define INF 9999999.0

Functions

- void initializeParameters (SupportSegmentation::Request &req)
- void ransacPlaneSegmentator (PCLCloudPtr inputCloud, PCLNormalPtr normals, PointIndices::Ptr &inlierOutput, ModelCoefficients::Ptr &coefficientOutput)
- ExtractIndices< PointXYZ > extract (true)
- void removePlaneInliner (PCLCloudPtr inputCloud, PointIndices::Ptr &removeIndex, PCLCloudPtr output)
- bool valueBelongsToArray (int value, PointIndices::Ptr inliers)
- PrimitiveIdxPtr createNewIdxMap (PrimitiveIdxPtr previousInliersMap, PointIndices::Ptr inliers, int level)
- bool isHorizontalPlane (PCLNormalPtr normal, ModelCoefficients::Ptr coefficients, const float referimentAxis[3])
- PCLCloudPtr getPointOnPlane (PCLCloudPtr plane, PrimitiveIdxPtr inlierIdx, int mapLevel)
- bool findSupports (SupportSegmentation::Request &req, SupportSegmentation::Response &res)
- int main (int argc, char **argv)

Variables

- static const float MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE = 0.030f
- static const float MIN_PLANE_PERCENTAGE_SIZE = 0.030f
- static const float MAX_VARIANCE_TH_FOR_HORIZONTAL = 0.09f
- static const int RANSAC_MAX_ITERATION_TH = 10
- static const float RANSAC_TH_DISTANCE_POINT_SHAPE = 0.02f
- static const float RANSAC_NORMAL_DISTANCE_WEIGHT = 0.9f
- static const float HORIZONTAL_AXIS [3] = { 0.0f, 0.0f, -1.0f}
- static const float SUPPORT_EDG_REMOVEE_OFSET = 0.02
- float minIterativeCloudPercentage
- float minPlanePercentageSize
- float minVarianceThForHorizontal

- float maxVarianceThForHorizontal
- float ransacThDistancePointShape
- float ransacNormaleDistanceWeigth
- float supportEdgeRemoveOfset
- float horizontalAxis [3]
- int ransacMaxIteration
- PCLVisualizer visual
- PCLCloudPtr originalCloud
- PCLNormalPtr originalNorms
- SACSegmentationFromNormals < PointXYZ, Normal > seg

7.15.1 Macro Definition Documentation

7.15.1.1 #define INF 9999999.0

Definition at line 188 of file supportsSegmentationServer.cpp.

Referenced by getPointOnPlane().

7.15.1.2 #define LESS_INF -9999999.0

Definition at line 187 of file supportsSegmentationServer.cpp.

Referenced by getPointOnPlane().

7.15.2 Function Documentation

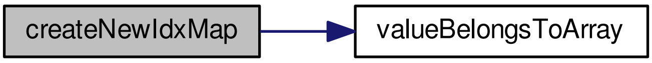
7.15.2.1 PrimitiveIdxPtr createNewIdxMap (PrimitiveIdxPtr *previousInliersMap,* PointIndices::Ptr *inliers,* int *level*)

Definition at line 144 of file supportsSegmentationServer.cpp.

References valueBelongsToArray().

Referenced by findSupports().

Here is the call graph for this function:



7.15.2.2 ExtractIndices< PointXYZ> extract (true)

Referenced by removePlaneInliner().

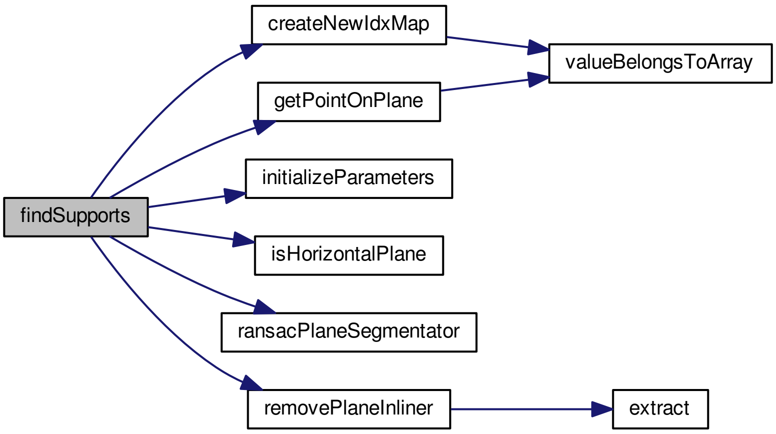
7.15.2.3 bool findSupports (SupportSegmentation::Request & *req,* SupportSegmentation::Response & *res*)

Definition at line 243 of file supportsSegmentationServer.cpp.

References createNewIdxMap(), getPointOnPlane(), HORIZONTAL_AXIS, horizontalAxis, initializeParameters(), isHorizontalPlane(), maxVarianceThForHorizontal, minIterativeCloudPercentage, minPlanePercentageSize, minVarianceThForHorizontal, originalCloud, originalNorms, ransacMaxIteration, ransacNormaleDistanceWeigth, ransacPlaneSegmentator(), ransacThDistancePointShape, removePlaneInliner(), and supportEdgeRemoveOfset.

Referenced by main().

Here is the call graph for this function:



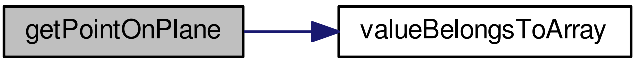
7.15.2.4 PCLCloudPtr getPointOnPlane (PCLCloudPtr *plane,* PrimitiveIdxPtr *inlierIdx,* int *mapLevel*)

Definition at line 189 of file supportsSegmentationServer.cpp.

References INF, LESS_INF, originalCloud, supportEdgeRemoveOfset, and valueBelongsToArray().

Referenced by findSupports().

Here is the call graph for this function:



7.15.2.5 void initializeParameters (SupportSegmentation::Request & *req*)

Definition at line 46 of file supportsSegmentationServer.cpp.

References HORIZONTAL_AXIS, horizontalAxis, MAX_VARIANCE_TH_FOR_HORIZONTAL, maxVarianceThForHorizontal, MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE, MIN_PLANE_PERCENTAGE_SIZE, minIterativeCloudPercentage, minPlanePercentageSize, minVarianceThForHorizontal, RANSAC_MAX_ITERATION_TH, RANSAC_NORMAL_DISTANCE_WEIGHT, RANSAC_TH_DISTANCE_POINT_SHAPE, ransacMaxIteration, ransacNormaleDistanceWeigth, ransacThDistancePointShape, SUPPORT_EDG_REMOVEE_OFSET, and supportEdgeRemoveOfset.

Referenced by findSupports().

7.15.2.6 bool isHorizontalPlane (PCLNormalPtr *normal,* ModelCoefficients::Ptr *coefficients,* const float *referimentAxis[3]*)

Definition at line 166 of file supportsSegmentationServer.cpp.

References maxVarianceThForHorizontal, and minVarianceThForHorizontal.

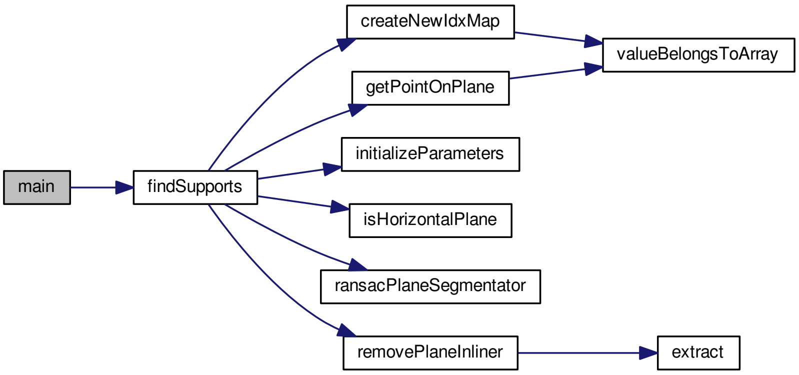
Referenced by findSupports().

7.15.2.7 int main (int *argc,* char ∗∗ *argv*)

Definition at line 360 of file supportsSegmentationServer.cpp.

References findSupports().

Here is the call graph for this function:



7.15.2.8 void ransacPlaneSegmentator (PCLCloudPtr *inputCloud,* PCLNormalPtr *normals,* PointIndices::Ptr & *inlierOutput,* ModelCoefficients::Ptr & *coefficientOutput*)

Definition at line 95 of file supportsSegmentationServer.cpp.

References ransacMaxIteration, ransacNormaleDistanceWeigth, ransacThDistancePointShape, and seg.

Referenced by findSupports().

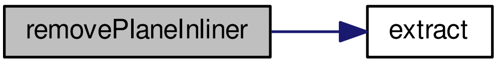
7.15.2.9 void removePlaneInliner (PCLCloudPtr *inputCloud,* PointIndices::Ptr & *removeIndex,* PCLCloudPtr *output*)

Definition at line 120 of file supportsSegmentationServer.cpp.

References extract().

Referenced by findSupports().

Here is the call graph for this function:



7.15.2.10 bool valueBelongsToArray (int *value,* PointIndices::Ptr *inliers*)

Definition at line 136 of file supportsSegmentationServer.cpp.

Referenced by createNewIdxMap(), and getPointOnPlane().

7.15.3 Variable Documentation

7.15.3.1 const float HORIZONTAL_AXIS[3] = { 0.0f, 0.0f, -1.0f} `[static]`

Definition at line 39 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), and initializeParameters().

7.15.3.2 float horizontalAxis[3]

Definition at line 44 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), and initializeParameters().

7.15.3.3 const float MAX_VARIANCE_TH_FOR_HORIZONTAL = 0.09f `[static]`

Definition at line 35 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.4 float maxVarianceThForHorizontal

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), initializeParameters(), and isHorizontalPlane().

7.15.3.5 const float MIN_ITERATIVE_CLOUD_PERCENTUAL_SIZE = 0.030f `[static]`

Definition at line 33 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.6 const float MIN_PLANE_PERCENTAGE_SIZE = 0.030f `[static]`

Definition at line 34 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.7 float minIterativeCloudPercentage

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), and initializeParameters().

7.15.3.8 float minPlanePercentageSize

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), and initializeParameters().

7.15.3.9 float minVarianceThForHorizontal

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), initializeParameters(), and isHorizontalPlane().

7.15.3.10 PCLCloudPtr originalCloud

Definition at line 84 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), and getPointOnPlane().

7.15.3.11 PCLNormalPtr originalNorms

Definition at line 85 of file supportsSegmentationServer.cpp.

Referenced by findSupports().

7.15.3.12 const int RANSAC_MAX_ITERATION_TH = 10 [static]

Definition at line 36 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.13 const float RANSAC_NORMAL_DISTANCE_WEIGHT = 0.9f [static]

Definition at line 38 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.14 const float RANSAC_TH_DISTANCE_POINT_SHAPE = 0.02f [static]

Definition at line 37 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.15 int ransacMaxIteration

Definition at line 45 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), initializeParameters(), and ransacPlaneSegmentator().

7.15.3.16 float ransacNormaleDistanceWeigth

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), initializeParameters(), and ransacPlaneSegmentator().

7.15.3.17 float ransacThDistancePointShape

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), initializeParameters(), and ransacPlaneSegmentator().

7.15.3.18 SACSegmentationFromNormals< PointXYZ, Normal> seg

Definition at line 94 of file supportsSegmentationServer.cpp.

Referenced by ransacConeDetaction(), ransacCylinderDetaction(), ransacPlaneDetaction(), ransacPlaneSegmentator(), and ransacSphereDetaction().

7.15.3.19 const float SUPPORT_EDG_REMOVEE_OFSET = 0.02 [static]

Definition at line 40 of file supportsSegmentationServer.cpp.

Referenced by initializeParameters().

7.15.3.20 float supportEdgeRemoveOfset

Definition at line 43 of file supportsSegmentationServer.cpp.

Referenced by findSupports(), getPointOnPlane(), and initializeParameters().

7.15.3.21 PCLVisualizer visual

Definition at line 83 of file supportsSegmentationServer.cpp.