

# DocumentClassHierarchy

## 1.0

Generated by Doxygen 1.8.6

Wed Nov 2 2016 19:37:35



# Contents



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">pcm</a>	.....	??
<a href="#">pcp</a>	.....	??
<a href="#">srvm</a>	.....	??



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">pcm::PCManager</a>	.....	??
<a href="#">pcp::PCPrimitive</a>	.....	??
<a href="#">vector3d</a>	.....	??





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">obj_segmentation.cpp</a>	??
src/ <a href="#">ransac_segmentation.cpp</a>	??
src/point_cloud_library/ <a href="#">pc_manager.cpp</a>	??
src/point_cloud_library/ <a href="#">pc_manager.h</a>	??
src/point_cloud_library/ <a href="#">pc_primitive.cpp</a>	??
src/point_cloud_library/ <a href="#">pc_primitive.h</a>	??
src/point_cloud_library/ <a href="#">srv_manager.h</a>	??
src/segmentation_services/ <a href="#">arm_filter_srv.cpp</a>	??
src/segmentation_services/ <a href="#">cluster_segmentation_srv.cpp</a>	??
src/segmentation_services/ <a href="#">cone_segmentation_srv.cpp</a>	??
src/segmentation_services/ <a href="#">cylinder_segmentation_srv.cpp</a>	??
src/segmentation_services/ <a href="#">deep_filter_srv.cpp</a>	??
src/segmentation_services/ <a href="#">plane_segmentation_srv.cpp</a>	??
src/segmentation_services/ <a href="#">sphere_segmentation_srv.cpp</a>	??
src/segmentation_services/ <a href="#">supports_segmentation_srv.cpp</a>	??



## Chapter 4

# Namespace Documentation

### 4.1 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`

#### 4.1.1 Function Documentation

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

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

#### 4.1.2 Variable Documentation

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

Definition at line 24 of file `pc_manager.cpp`.

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

**4.1.2.2** `VoxelGrid<PointXYZ> pcm::sor [static]`

Definition at line 27 of file `pc_manager.cpp`.

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

## 4.2 pcp Namespace Reference

### Classes

- class [PCPrimitive](#)

## 4.3 srvm Namespace Reference

### Functions

- const vector< float > [DEFAULT\\_SERVICE\\_VEC\\_PARAMETER\\_REQUEST](#) ([DEFAULT\\_SERVICE\\_ARRAY\\_PARAMETER\\_REQUEST](#), [DEFAULT\\_SERVICE\\_ARRAY\\_PARAMETER\\_REQUEST](#)+sizeof([DEFAULT\\_SERVICE\\_ARRAY\\_PARAMETER\\_REQUEST](#))/sizeof(float))
- string [getStringParameter](#) (string input, const string defaultValue)
- string [getStringPtrParameter](#) (char \*input, const string defaultValue)
- bool [getBoolParameter](#) (string input, const bool defaultValue)
- bool [getBoolPtrParameter](#) (char \*input, const bool defaultValue)
- string [getPathParameter](#) (string input, const string defaultValue)
- string [getPathPtrParameter](#) (char \*input, const string defaultValue)
- float [getServiceFloatParameter](#) (float input, const float defaultValue)
- int [getServiceIntParameter](#) (int input, const int defaultValue)
- string [getServiceStringParameter](#) (string input, const string defaultValue)
- vector< float > [getService3DArrayParameter](#) (vector< float > input, const vector< float > defaultValue)
- vector< float > [get3DArray](#) (const float values[])
- vector< float > [getService3DArrayParameter](#) (vector< float > input, const float defaultValue[])
- string [getFlagValueToPrint](#) (bool flag)
- string [getArrayToPrint](#) (vector< float > arr)

### Variables

- const string [SRV\\_NAME\\_DEEP\\_FILTER](#) = "deep\_filter\_srv"
- const string [SRV\\_NAME\\_SUPPORT\\_FILTER](#) = "support\_segmentation\_srv"
- const string [SRV\\_NAME\\_CUSTER\\_FILTER](#) = "cluster\_Segmentation\_srv"
- const string [SRV\\_NAME\\_ARM\\_FILTER](#) = "arm\_filter\_srv"
- const string [SRV\\_NAME\\_RANSAC\\_SPHERE\\_FILTER](#) = "sphere\_segmentation\_srv"
- const string [SRV\\_NAME\\_RANSAC\\_CYLINDER\\_FILTER](#) = "cylinder\_segmentation\_srv"
- const string [SRV\\_NAME\\_RANSAC\\_CONE\\_FILTER](#) = "cone\_segmentation\_srv"
- const string [SRV\\_NAME\\_RANSAC\\_PLANE\\_FILTER](#) = "plane\_segmentation\_srv"
- const string [PARAM\\_NAME\\_INPUT\\_CLOUD\\_REFERENCE\\_FRAME](#) = "/pitt/ref\_frame/input\_cloud"
- const string [PARAM\\_NAME\\_OUTPUT\\_CLOUD\\_REFERENCE\\_FRAME](#) = "/pitt/ref\_frame/output\_cloud"
- const string [PARAM\\_NAME\\_DEEP\\_SRV\\_Z\\_THRESHOLD](#) = "/pitt/service/deep\_filter/z\_threshold"
- const string [PARAM\\_NAME\\_ARM\\_SRV\\_MIN\\_FOREARM\\_BOX](#) = "/pitt/srv/arm\_filter/min\_forearm\_box"
- const string [PARAM\\_NAME\\_ARM\\_SRV\\_MAX\\_FOREARM\\_BOX](#) = "/pitt/srv/arm\_filter/max\_forearm\_box"
- const string [PARAM\\_NAME\\_ARM\\_SRV\\_MIN\\_ELBOW\\_BOX](#) = "/pitt/srv/arm\_filter/min\_elbow\_box"
- const string [PARAM\\_NAME\\_ARM\\_SRV\\_MAX\\_ELBOW\\_BOX](#) = "/pitt/srv/arm\_filter/max\_elbow\_box"
- const string [PARAM\\_NAME\\_CLUSTER\\_TOLERANCE](#) = "/pitt/srv/cluster\_segmentation/tolerance"
- const string [PARAM\\_NAME\\_CLUSTER\\_MIN\\_RATE](#) = "/pitt/srv/cluster\_segmentation/min\_rate"
- const string [PARAM\\_NAME\\_CLUSTER\\_MAX\\_RATE](#) = "/pitt/srv/cluster\_segmentation/max\_rate"
- const string [PARAM\\_NAME\\_CLUSTER\\_MIN\\_INPUT\\_SIZE](#) = "/pitt/srv/cluster\_segmentation/min\_input\_size"
- const string [PARAM\\_NAME\\_SPHERE\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = "/pitt/srv/sphere\_segmentation/normal\_distance\_weight"
- const string [PARAM\\_NAME\\_SPHERE\\_DISTANCE\\_TH](#) = "/pitt/srv/sphere\_segmentation/distance\_th"

- const string [PARAM\\_NAME\\_SPHERE\\_MAX\\_ITERATION\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/max\_iter\_limit"
- const string [PARAM\\_NAME\\_SPHERE\\_MIN\\_RADIUS\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/min\_radius\_limit"
- const string [PARAM\\_NAME\\_SPHERE\\_MAX\\_RADIUS\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/max\_radius\_limit"
- const string [PARAM\\_NAME\\_SPHERE\\_EPS\\_ANGLE\\_TH](#) = "/pitt/srv/sphere\_segmentation/eps\_angle\_th"
- const string [PARAM\\_NAME\\_SPHERE\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/sphere\_segmentation/min\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_SPHERE\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/sphere\_segmentation/max\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_SPHERE\\_MIN\\_INLIERS](#) = "/pitt/srv/sphere\_segmentation/min\_inliers"
- const string [PARAM\\_NAME\\_CYLINDER\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = "/pitt/srv/cylinder\_segmentation/normal\_distance\_weight"
- const string [PARAM\\_NAME\\_CYLINDER\\_DISTANCE\\_TH](#) = "/pitt/srv/cylinder\_segmentation/distance\_th"
- const string [PARAM\\_NAME\\_CYLINDER\\_MAX\\_ITERATION\\_LIMIT](#) = "/pitt/srv/cylinder\_segmentation/max\_iter\_limit"
- const string [PARAM\\_NAME\\_CYLINDER\\_MIN\\_RADIUS\\_LIMIT](#) = "/pitt/srv/cylinder\_segmentation/min\_radius\_limit"
- const string [PARAM\\_NAME\\_CYLINDER\\_MAX\\_RADIUS\\_LIMIT](#) = "/pitt/srv/cylinder\_segmentation/max\_radius\_limit"
- const string [PARAM\\_NAME\\_CYLINDER\\_EPS\\_ANGLE\\_TH](#) = "/pitt/srv/cylinder\_segmentation/eps\_angle\_th"
- const string [PARAM\\_NAME\\_CYLINDER\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/cylinder\_segmentation/min\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_CYLINDER\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/cylinder\_segmentation/max\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_CYLINDER\\_MIN\\_INLIERS](#) = "/pitt/srv/cylinder\_segmentation/min\_inliers"
- const string [PARAM\\_NAME\\_CONE\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = "/pitt/srv/cone\_segmentation/normal\_distance\_weight"
- const string [PARAM\\_NAME\\_CONE\\_DISTANCE\\_TH](#) = "/pitt/srv/cone\_segmentation/distance\_th"
- const string [PARAM\\_NAME\\_CONE\\_MAX\\_ITERATION\\_LIMIT](#) = "/pitt/srv/cone\_segmentation/max\_iter\_limit"
- const string [PARAM\\_NAME\\_CONE\\_MIN\\_RADIUS\\_LIMIT](#) = "/pitt/srv/cone\_segmentation/min\_radius\_limit"
- const string [PARAM\\_NAME\\_CONE\\_MAX\\_RADIUS\\_LIMIT](#) = "/pitt/srv/cone\_segmentation/max\_radius\_limit"
- const string [PARAM\\_NAME\\_CONE\\_EPS\\_ANGLE\\_TH](#) = "/pitt/srv/cone\_segmentation/eps\_angle\_th"
- const string [PARAM\\_NAME\\_CONE\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/cone\_segmentation/min\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_CONE\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/cone\_segmentation/max\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_CONE\\_MIN\\_INLIERS](#) = "/pitt/srv/cone\_segmentation/min\_inliers"
- const string [PARAM\\_NAME\\_PLANE\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = "/pitt/srv/plane\_segmentation/normal\_distance\_weight"
- const string [PARAM\\_NAME\\_PLANE\\_DISTANCE\\_TH](#) = "/pitt/srv/plane\_segmentation/distance\_th"
- const string [PARAM\\_NAME\\_PLANE\\_MAX\\_ITERATION\\_LIMIT](#) = "/pitt/srv/plane\_segmentation/max\_iter\_limit"
- const string [PARAM\\_NAME\\_PLANE\\_EPS\\_ANGLE\\_TH](#) = "/pitt/srv/plane\_segmentation/eps\_angle\_th"
- const string [PARAM\\_NAME\\_PLANE\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/plane\_segmentation/min\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_PLANE\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = "/pitt/srv/plane\_segmentation/max\_opening\_angle\_deg"
- const string [PARAM\\_NAME\\_PLANE\\_MIN\\_INLIERS](#) = "/pitt/srv/plane\_segmentation/min\_inliers"
- const string [PARAM\\_NAME\\_MIN\\_ITERATIVE\\_CLOUD\\_PERCENTAGE](#) = "/pitt/srv/supports\_segmentation/min\_iter\_cloud\_percent"
- const string [PARAM\\_NAME\\_MIN\\_ITERATIVE\\_SUPPORT\\_PERCENTAGE](#) = "/pitt/srv/supports\_segmentation/min\_iter\_support\_percent"

- const string `PARAM_NAME_HORIZONTAL_VARIANCE_THRESHOLD` = "/pitt/srv/supports\_segmentation/horizontal\_variance\_th"
- const string `PARAM_NAME_RANSAC_IN_SHAPE_DISTANCE_POINT_THRESHOLD` = "/pitt/srv/supports\_segmentation/in\_shape\_distance\_th"
- const string `PARAM_NAME_RANSAC_MODEL_NORMAL_DISTANCE_WEIGHT` = "/pitt/srv/supports\_segmentation/normal\_distance\_weight"
- const string `PARAM_NAME_RANSAC_MAX_ITERATION_THRESHOLD` = "/pitt/srv/supports\_segmentation/max\_iter"
- const string `PARAM_NAME_HORIZONTAL_AXIS` = "/pitt/srv/supports\_segmentation/horizontal\_axis"
- const string `PARAM_NAME_SUPPORT_EDGE_REMOVE_OFFSET` = "/pitt/srv/supports\_segmentation/edge\_remove\_offset"
- const string `DEFAULT_PARAM_INPUT_CLOUD_REFERENCE_FRAME` = "/camera\_depth\_optical\_frame"
- const string `DEFAULT_PARAM_OUTPUT_CLOUD_REFERENCE_FRAME` = "/world"
- const string `DEFAULT_INPUT_PARAM_RAW_CLOUD_TOPIC` = "/camera/depth/points"
- const string `DEFAULT_INPUT_PARAM_CENTROID_LOG_FILE` = ""
- const bool `DEFAULT_INPUT_PARAM_SHOW_ORIGINAL_CLOUD` = false
- const bool `DEFAULT_INPUT_PARAM_SHOW_SUPPORTS` = false
- const bool `DEFAULT_INPUT_PARAM_SHOW_OBJECT_ON_SUPPORT` = false
- const bool `DEFAULT_INPUT_PARAM_SHOW_CLUSTERS` = false
- const string `DEFAULT_PARAM_ARM_SRV_CAMERA_FRAME` = "/camera\_depth\_optical\_frame"
- const string `DEFAULT_PARAM_ARM_SRV_RIGHT_FOREARM_FRAME` = "/right\_lower\_forearm"
- const string `DEFAULT_PARAM_ARM_SRV_LEFT_FOREARM_FRAME` = "/left\_lower\_forearm"
- const string `DEFAULT_PARAM_ARM_SRV_RIGHT_ELBOW_FRAME` = "/right\_lower\_elbow"
- const string `DEFAULT_PARAM_ARM_SRV_LEFT_ELBOW_FRAME` = "/left\_lower\_elbow"
- const bool `DEFAULT_PARAM_ARM_SRV_SHOW_CLOUDS` = false
- const string `TOPIC_OUT_NAME_OBJECT_PERCEPTION` = "obj\_segmentation/ClusterOutput"
- const int `DEFAULT_SERVICE_PARAMETER_REQUEST` = -1
- const float `DEFAULT_SERVICE_PARAMETER_REQUEST_F` = -1.0f
- const float `DEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST` [1] = {-1}
- const float `DEFAULT_TF_WAIT_SECONDS` = 2.0f
- const string `DEFAULT_SYMBOL` = "."

### 4.3.1 Function Documentation

4.3.1.1 `const vector<float> srvm::DEFAULT_SERVICE_VEC_PARAMETER_REQUEST ( DEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST , DEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST+ sizeofDEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST)/sizeof(float )`

Referenced by `callArmFilter()`, and `callSupportFilter()`.

4.3.1.2 `vector< float> srvm::get3DArray ( const float values[] )`

Definition at line 181 of file `srv_manager.h`.

Referenced by `getService3DArrayParameter()`.

4.3.1.3 `string srvm::getArrayToPrint ( vector< float > arr )`

Definition at line 196 of file `srv_manager.h`.

Referenced by `filter()`.

#### 4.3.1.4 bool srvm::getBoolParameter ( string *input*, const bool *defaultValue* )

Definition at line 139 of file `srv_manager.h`.

References `DEFAULT_SYMBOL`.

Referenced by `getBoolPtrParameter()`.

#### 4.3.1.5 bool srvm::getBoolPtrParameter ( char \* *input*, const bool *defaultValue* )

Definition at line 144 of file `srv_manager.h`.

References `getBoolParameter()`.

Referenced by `main()`.

Here is the call graph for this function:



#### 4.3.1.6 string srvm::getFlagValueToPrint ( bool *flag* )

Definition at line 191 of file `srv_manager.h`.

Referenced by `main()`.

#### 4.3.1.7 string srvm::getPathParameter ( string *input*, const string *defaultValue* )

Definition at line 149 of file `srv_manager.h`.

References `DEFAULT_SYMBOL`, and `pcm::PCManager::getFomrattedData()`.

Referenced by `getPathPtrParameter()`.

Here is the call graph for this function:



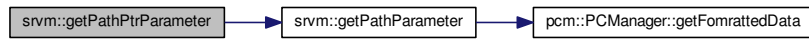
#### 4.3.1.8 string srvm::getPathPtrParameter ( char \* *input*, const string *defaultValue* )

Definition at line 158 of file `srv_manager.h`.

References `getPathParameter()`.

Referenced by `main()`.

Here is the call graph for this function:



**4.3.1.9** `vector<float> srvm::getService3DArrayParameter ( vector< float > input, const vector< float > defaultValue )`

Definition at line 176 of file `srv_manager.h`.

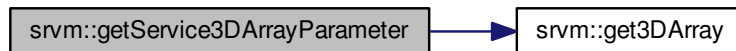
Referenced by `filter()`, and `initializeInputParameters()`.

**4.3.1.10** `vector<float> srvm::getService3DArrayParameter ( vector< float > input, const float defaultValue[] )`

Definition at line 185 of file `srv_manager.h`.

References `get3DArray()`.

Here is the call graph for this function:



**4.3.1.11** `float srvm::getServiceFloatParameter ( float input, const float defaultValue )`

Definition at line 163 of file `srv_manager.h`.

Referenced by `deepFiltering()`, and `initializeInputParameters()`.

**4.3.1.12** `int srvm::getServiceIntParameter ( int input, const int defaultValue )`

Definition at line 168 of file `srv_manager.h`.

Referenced by `initializeInputParameters()`.

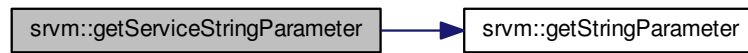
**4.3.1.13** `string srvm::getServiceStringParameter ( string input, const string defaultValue )`

Definition at line 173 of file `srv_manager.h`.

References `getStringParameter()`.



Here is the call graph for this function:



#### 4.3.1.14 string srvm::getStringParameter ( string *input*, const string *defaultValue* )

Definition at line 129 of file `srv_manager.h`.

References `DEFAULT_SYMBOL`.

Referenced by `getServiceStringParameter()`, `getStringPtrParameter()`, and `main()`.

#### 4.3.1.15 string srvm::getStringPtrParameter ( char \* *input*, const string *defaultValue* )

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

References `getStringParameter()`.

Referenced by `main()`.

Here is the call graph for this function:



### 4.3.2 Variable Documentation

#### 4.3.2.1 const string srvm::DEFAULT\_INPUT\_PARAM\_CENTROID\_LOG\_FILE = ""

Definition at line 101 of file `srv_manager.h`.

Referenced by `main()`.

#### 4.3.2.2 const string srvm::DEFAULT\_INPUT\_PARAM\_RAW\_CLOUD\_TOPIC = "/camera/depth/points"

Definition at line 100 of file `srv_manager.h`.

Referenced by `main()`.

#### 4.3.2.3 const bool srvm::DEFAULT\_INPUT\_PARAM\_SHOW\_CLUSTERS = false

Definition at line 105 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.4** `const bool srvm::DEFAULT_INPUT_PARAM_SHOW_OBJECT_ON_SUPPORT = false`

Definition at line 104 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.5** `const bool srvm::DEFAULT_INPUT_PARAM_SHOW_ORIGINAL_CLOUD = false`

Definition at line 102 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.6** `const bool srvm::DEFAULT_INPUT_PARAM_SHOW_SUPPORTS = false`

Definition at line 103 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.7** `const string srvm::DEFAULT_PARAM_ARM_SRV_CAMERA_FRAME = "/camera_depth_optical_frame"`

Definition at line 106 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.8** `const string srvm::DEFAULT_PARAM_ARM_SRV_LEFT_ELBOW_FRAME = "/left_lower_elbow"`

Definition at line 110 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.9** `const string srvm::DEFAULT_PARAM_ARM_SRV_LEFT_FOREARM_FRAME = "/left_lower_forearm"`

Definition at line 108 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.10** `const string srvm::DEFAULT_PARAM_ARM_SRV_RIGHT_ELBOW_FRAME = "/right_lower_elbow"`

Definition at line 109 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.11** `const string srvm::DEFAULT_PARAM_ARM_SRV_RIGHT_FOREARM_FRAME = "/right_lower_forearm"`

Definition at line 107 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.12** `const bool srvm::DEFAULT_PARAM_ARM_SRV_SHOW_CLOUDS = false`

Definition at line 111 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.13** `const string srvm::DEFAULT_PARAM_INPUT_CLOUD_REFERENCE_FRAME = "/camera_depth_optical_frame"`

Definition at line 98 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.14** `const string srvm::DEFAULT_PARAM_OUTPUT_CLOUD_REFERENCE_FRAME = "/world"`

Definition at line 99 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.15** `const float srvm::DEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST[1] = {-1}`

Definition at line 120 of file `srv_manager.h`.

**4.3.2.16** `const int srvm::DEFAULT_SERVICE_PARAMETER_REQUEST = -1`

Definition at line 118 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.17** `const float srvm::DEFAULT_SERVICE_PARAMETER_REQUEST_F = -1.0f`

Definition at line 119 of file `srv_manager.h`.

Referenced by `callDeepFilter()`, and `callSupportFilter()`.

**4.3.2.18** `const string srvm::DEFAULT_SYMBOL = "."`

Definition at line 127 of file `srv_manager.h`.

Referenced by `getBoolParameter()`, `getPathParameter()`, and `getStringParameter()`.

**4.3.2.19** `const float srvm::DEFAULT_TF_WAIT_SECONDS = 2.0f`

Definition at line 125 of file `srv_manager.h`.

Referenced by `main()`.

**4.3.2.20** `const string srvm::PARAM_NAME_ARM_SRV_MAX_ELBOW_BOX = "/pitt/srv/arm_filter/max_elbow_box"`

Definition at line 42 of file `srv_manager.h`.

Referenced by `callArmFilter()`.

**4.3.2.21** `const string srvm::PARAM_NAME_ARM_SRV_MAX_FOREARM_BOX = "/pitt/srv/arm_filter/max_forearm_box"`

Definition at line 40 of file `srv_manager.h`.

Referenced by `callArmFilter()`.

**4.3.2.22** `const string srvm::PARAM_NAME_ARM_SRV_MIN_ELBOW_BOX = "/pitt/srv/arm_filter/min_elbow_box"`

Definition at line 41 of file `srv_manager.h`.

Referenced by `callArmFilter()`.

**4.3.2.23** `const string srvm::PARAM_NAME_ARM_SRV_MIN_FOREARM_BOX = "/pitt/srv/arm_filter/min_forearm_box"`

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

Referenced by `callArmFilter()`.

**4.3.2.24** `const string srvm::PARAM_NAME_CLUSTER_MAX_RATE = "/pitt/srv/cluster_segmentation/max_rate"`

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

Referenced by `clusterize()`.

**4.3.2.25** `const string srvm::PARAM_NAME_CLUSTER_MIN_INPUT_SIZE = "/pitt/srv/cluster_segmentation/min_input_size"`

Definition at line 47 of file `srv_manager.h`.

**4.3.2.26** `const string srvm::PARAM_NAME_CLUSTER_MIN_RATE = "/pitt/srv/cluster_segmentation/min_rate"`

Definition at line 45 of file `srv_manager.h`.

Referenced by `clusterize()`.

**4.3.2.27** `const string srvm::PARAM_NAME_CLUSTER_TOLERANCE = "/pitt/srv/cluster_segmentation/tolerance"`

Definition at line 44 of file `srv_manager.h`.

Referenced by `clusterize()`.

**4.3.2.28** `const string srvm::PARAM_NAME_CONE_DISTANCE_TH = "/pitt/srv/cone_segmentation/distance_th"`

Definition at line 70 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.29** `const string srvm::PARAM_NAME_CONE_EPS_ANGLE_TH = "/pitt/srv/cone_segmentation/eps_angle_th"`

Definition at line 74 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.30** `const string srvm::PARAM_NAME_CONE_MAX_ITERATION_LIMIT = "/pitt/srv/cone_segmentation/max_iter_limit"`

Definition at line 71 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.31** `const string srvm::PARAM_NAME_CONE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/cone_segmentation/max_opening_angle_deg"`

Definition at line 76 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.32** `const string srvm::PARAM_NAME_CONE_MAX_RADIUS_LIMIT = "/pitt/srv/cone_segmentation/max_radius_limit"`

Definition at line 73 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.33** `const string srvm::PARAM_NAME_CONE_MIN_INLIERS = "/pitt/srv/cone_segmentation/min_inliers"`

Definition at line 77 of file `srv_manager.h`.

Referenced by `callRansacConeSegmentation()`.

**4.3.2.34** `const string srvm::PARAM_NAME_CONE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/cone_segmentation/min_opening_angle_deg"`

Definition at line 75 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.35** `const string srvm::PARAM_NAME_CONE_MIN_RADIUS_LIMIT = "/pitt/srv/cone_segmentation/min_radius_limit"`

Definition at line 72 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.36** `const string srvm::PARAM_NAME_CONE_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/cone_segmentation/normal_distance_weight"`

Definition at line 69 of file `srv_manager.h`.

Referenced by `ransacConeDetaction()`.

**4.3.2.37** `const string srvm::PARAM_NAME_CYLINDER_DISTANCE_TH = "/pitt/srv/cylinder_segmentation/distance_th"`

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

Referenced by `ransacCylinderDetaction()`.

**4.3.2.38** `const string srvm::PARAM_NAME_CYLINDER_EPS_ANGLE_TH = "/pitt/srv/cylinder_segmentation/eps_angle_th"`

Definition at line 64 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.39** `const string srvm::PARAM_NAME_CYLINDER_MAX_ITERATION_LIMIT = "/pitt/srv/cylinder_segmentation/max_iter_limit"`

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

Referenced by `ransacCylinderDetaction()`.

**4.3.2.40** `const string srvm::PARAM_NAME_CYLINDER_MAX_OPENING_ANGLE_DEGREE =  
"/pitt/srv/cylinder_segmentation/max_opening_angle_deg"`

Definition at line 66 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.41** `const string srvm::PARAM_NAME_CYLINDER_MAX_RADIUS_LIMIT = "/pitt/srv/cylinder_segmentation/max_radius -  
limit"`

Definition at line 63 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.42** `const string srvm::PARAM_NAME_CYLINDER_MIN_INLIERS = "/pitt/srv/cylinder_segmentation/min_inliers"`

Definition at line 67 of file `srv_manager.h`.

Referenced by `callRansacCylinderSegmentation()`.

**4.3.2.43** `const string srvm::PARAM_NAME_CYLINDER_MIN_OPENING_ANGLE_DEGREE =  
"/pitt/srv/cylinder_segmentation/min_opening_angle_deg"`

Definition at line 65 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.44** `const string srvm::PARAM_NAME_CYLINDER_MIN_RADIUS_LIMIT = "/pitt/srv/cylinder_segmentation/min_radius_limit"`

Definition at line 62 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.45** `const string srvm::PARAM_NAME_CYLINDER_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/cylinder -  
segmentation/normal_distance_weight"`

Definition at line 59 of file `srv_manager.h`.

Referenced by `ransacCylinderDetaction()`.

**4.3.2.46** `const string srvm::PARAM_NAME_DEEP_SRV_Z_THRESHOLD = "/pitt/service/deep_filter/z_threshold"`

Definition at line 37 of file `srv_manager.h`.

Referenced by `callDeepFilter()`.

**4.3.2.47** `const string srvm::PARAM_NAME_HORIZONTAL_AXIS = "/pitt/srv/supports_segmentation/horizontal_axis"`

Definition at line 94 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.48** `const string srvm::PARAM_NAME_HORIZONTAL_VARIANCE_THRESHOLD = "/pitt/srv/supports -  
segmentation/horizontal_variance_th"`

Definition at line 90 of file `srv_manager.h`.

Referenced by callSupportFilter().

**4.3.2.49** `const string srv::PARAM_NAME_INPUT_CLOUD_REFERENCE_FRAME = "/pitt/ref_frame/input_cloud"`

Definition at line 35 of file srv\_manager.h.

Referenced by main().

**4.3.2.50** `const string srv::PARAM_NAME_MIN_ITERATIVE_CLOUD_PERCENTAGE = "/pitt/srv/supports_segmentation/min_iter_cloud_percent"`

Definition at line 88 of file srv\_manager.h.

Referenced by callSupportFilter().

**4.3.2.51** `const string srv::PARAM_NAME_MIN_ITERATIVE_SUPPORT_PERCENTAGE = "/pitt/srv/supports_segmentation/min_iter_support_percent"`

Definition at line 89 of file srv\_manager.h.

Referenced by callSupportFilter().

**4.3.2.52** `const string srv::PARAM_NAME_OUTPUT_CLOUD_REFERENCE_FRAME = "/pitt/ref_frame/output_cloud"`

Definition at line 36 of file srv\_manager.h.

Referenced by main().

**4.3.2.53** `const string srv::PARAM_NAME_PLANE_DISTANCE_TH = "/pitt/srv/plane_segmentation/distance_th"`

Definition at line 80 of file srv\_manager.h.

Referenced by ransacPlaneDetaction().

**4.3.2.54** `const string srv::PARAM_NAME_PLANE_EPS_ANGLE_TH = "/pitt/srv/plane_segmentation/eps_angle_th"`

Definition at line 82 of file srv\_manager.h.

Referenced by ransacPlaneDetaction().

**4.3.2.55** `const string srv::PARAM_NAME_PLANE_MAX_ITERATION_LIMIT = "/pitt/srv/plane_segmentation/max_iter_limit"`

Definition at line 81 of file srv\_manager.h.

Referenced by ransacPlaneDetaction().

**4.3.2.56** `const string srv::PARAM_NAME_PLANE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/plane_segmentation/max_opening_angle_deg"`

Definition at line 84 of file srv\_manager.h.

Referenced by ransacPlaneDetaction().

**4.3.2.57** `const string srvm::PARAM_NAME_PLANE_MIN_INLIERS = "/pitt/srv/plane_segmentation/min_inliers"`

Definition at line 85 of file `srv_manager.h`.

Referenced by `callRansacPlaneSegmentation()`.

**4.3.2.58** `const string srvm::PARAM_NAME_PLANE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/plane_segmentation/min_opening_angle_deg"`

Definition at line 83 of file `srv_manager.h`.

Referenced by `ransacPlaneDetaction()`.

**4.3.2.59** `const string srvm::PARAM_NAME_PLANE_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/plane_segmentation/normal_distance_weight"`

Definition at line 79 of file `srv_manager.h`.

Referenced by `ransacPlaneDetaction()`.

**4.3.2.60** `const string srvm::PARAM_NAME_RANSAC_IN_SHAPE_DISTANCE_POINT_THRESHOLD = "/pitt/srv/supports_segmentation/in_shape_distance_th"`

Definition at line 91 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.61** `const string srvm::PARAM_NAME_RANSAC_MAX_ITERATION_THRESHOLD = "/pitt/srv/supports_segmentation/max_iter"`

Definition at line 93 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.62** `const string srvm::PARAM_NAME_RANSAC_MODEL_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/supports_segmentation/normal_distance_weight"`

Definition at line 92 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.63** `const string srvm::PARAM_NAME_SPHERE_DISTANCE_TH = "/pitt/srv/sphere_segmentation/distance_th"`

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

Referenced by `ransacSphereDetection()`.

**4.3.2.64** `const string srvm::PARAM_NAME_SPHERE_EPS_ANGLE_TH = "/pitt/srv/sphere_segmentation/eps_angle_th"`

Definition at line 54 of file `srv_manager.h`.

Referenced by `ransacSphereDetection()`.

**4.3.2.65** `const string srvm::PARAM_NAME_SPHERE_MAX_ITERATION_LIMIT = "/pitt/srv/sphere_segmentation/max_iter_limit"`

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



Referenced by `ransacSphereDetection()`.

**4.3.2.66** `const string srvm::PARAM_NAME_SPHERE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/sphere_segmentation/max_opening_angle_deg"`

Definition at line 56 of file `srv_manager.h`.

Referenced by `ransacSphereDetection()`.

**4.3.2.67** `const string srvm::PARAM_NAME_SPHERE_MAX_RADIUS_LIMIT = "/pitt/srv/sphere_segmentation/max_radius_limit"`

Definition at line 53 of file `srv_manager.h`.

Referenced by `ransacSphereDetection()`.

**4.3.2.68** `const string srvm::PARAM_NAME_SPHERE_MIN_INLIERS = "/pitt/srv/sphere_segmentation/min_inliers"`

Definition at line 57 of file `srv_manager.h`.

Referenced by `callRansacSphereSegmentation()`.

**4.3.2.69** `const string srvm::PARAM_NAME_SPHERE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/sphere_segmentation/min_opening_angle_deg"`

Definition at line 55 of file `srv_manager.h`.

Referenced by `ransacSphereDetection()`.

**4.3.2.70** `const string srvm::PARAM_NAME_SPHERE_MIN_RADIUS_LIMIT = "/pitt/srv/sphere_segmentation/min_radius_limit"`

Definition at line 52 of file `srv_manager.h`.

Referenced by `ransacSphereDetection()`.

**4.3.2.71** `const string srvm::PARAM_NAME_SPHERE_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/sphere_segmentation/normal_distance_weight"`

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

Referenced by `ransacSphereDetection()`.

**4.3.2.72** `const string srvm::PARAM_NAME_SUPPORT_EDGE_REMOVE_OFFSET = "/pitt/srv/supports_segmentation/edge_remove_offset"`

Definition at line 95 of file `srv_manager.h`.

Referenced by `callSupportFilter()`.

**4.3.2.73** `const string srvm::SRV_NAME_ARM_FILTER = "arm_filter_srv"`

Definition at line 28 of file `srv_manager.h`.

Referenced by `callArmFilter()`, and `main()`.

**4.3.2.74** `const string srvm::SRV_NAME_CLUSTER_FILTER = "cluster_Segmentation_srv"`

Definition at line 27 of file `srv_manager.h`.

Referenced by `callClusterSegmentation()`, and `main()`.

**4.3.2.75** `const string srvm::SRV_NAME_DEEP_FILTER = "deep_filter_srv"`

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

Referenced by `callDeepFilter()`, and `main()`.

**4.3.2.76** `const string srvm::SRV_NAME_RANSAC_CONE_FILTER = "cone_segmentation_srv"`

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

Referenced by `callRansacConeSegmentation()`, and `main()`.

**4.3.2.77** `const string srvm::SRV_NAME_RANSAC_CYLINDER_FILTER = "cylinder_segmentation_srv"`

Definition at line 30 of file `srv_manager.h`.

Referenced by `callRansacCylinderSegmentation()`, and `main()`.

**4.3.2.78** `const string srvm::SRV_NAME_RANSAC_PLANE_FILTER = "plane_segmentation_srv"`

Definition at line 32 of file `srv_manager.h`.

Referenced by `callRansacPlaneSegmentation()`, and `main()`.

**4.3.2.79** `const string srvm::SRV_NAME_RANSAC_SPHERE_FILTER = "sphere_segmentation_srv"`

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

Referenced by `callRansacSphereSegmentation()`, and `main()`.

**4.3.2.80** `const string srvm::SRV_NAME_SUPPORT_FILTER = "support_segmentation_srv"`

Definition at line 26 of file `srv_manager.h`.

Referenced by `callSupportFilter()`, and `main()`.

**4.3.2.81** `const string srvm::TOPIC_OUT_NAME_OBJECT_PERCEPTION = "obj_segmentation/ClusterOutput"`

Definition at line 115 of file `srv_manager.h`.

Referenced by `main()`.

## Chapter 5

# Class Documentation

### 5.1 pcm::PCManager Class Reference

```
#include <pc_manager.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 [clearPrimitiveShape](#) ()
- [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 string [getFomrattedData](#) ()
- static bool [writeToFile](#) (string txt, string filePath, bool append)

### 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

### Private Member Functions

- void [initialize](#) (bool [visualizationFlag](#))

### Private Attributes

- [PCLCloudPtr](#) [originalCloud](#)
- [PCLNormalPtr](#) [originalNorms](#)
- [PCLVisualizer](#) [visor](#)
- bool [visualizationFlag](#)
- vector< [PCPrimitivePtr](#) > [primitiveList](#)

#### 5.1.1 Detailed Description

Definition at line 27 of file [pc\\_manager.h](#).

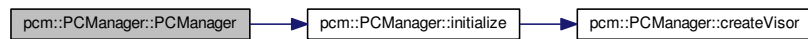
## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 pcm::PCManager::PCManager ( )

Definition at line 243 of file pc\_manager.cpp.

References `DEFAULT_VISUALIZATION_FLAG`, and `initialize()`.

Here is the call graph for this function:

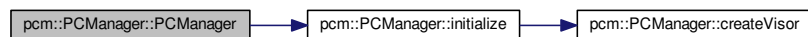


### 5.1.2.2 pcm::PCManager::PCManager ( bool *visualizationFlag* )

Definition at line 246 of file pc\_manager.cpp.

References `initialize()`.

Here is the call graph for this function:



### 5.1.2.3 pcm::PCManager::~~PCManager ( ) [virtual]

Definition at line 251 of file pc\_manager.cpp.

## 5.1.3 Member Function Documentation

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

### 5.1.3.2 int pcm::PCManager::clearPtimitiveShape ( )

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

Definition at line 175 of file pc\_manager.cpp.

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

Definition at line 90 of file pc\_manager.cpp.

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

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

Definition at line 85 of file pc\_manager.cpp.

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

Definition at line 80 of file pc\_manager.cpp.

Referenced by getOriginalCloudRosMsg().

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

Definition at line 112 of file pc\_manager.cpp.

Referenced by ransacConeDetaction(), ransacCylinderDetaction(), ransacPlaneDetaction(), and ransacSphere-Detection().

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

Definition at line 30 of file pc\_manager.cpp.

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

Definition at line 49 of file pc\_manager.cpp.

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

Definition at line 43 of file pc\_manager.cpp.

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

Definition at line 131 of file pc\_manager.cpp.

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

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

Definition at line 55 of file pc\_manager.cpp.

References DEFAULT\_DOWSEAMPLIG\_RATE.

Referenced by downSampling(), and setOriginalCloud().

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

Definition at line 58 of file pc\_manager.cpp.

References downSampling().

Here is the call graph for this function:



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

Definition at line 61 of file pc\_manager.cpp.

References pcm::sor.

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

Definition at line 68 of file pc\_manager.cpp.

References DEFAULT\_NORM\_SEARCH.

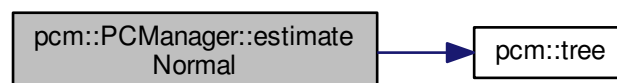
Referenced by setOriginalCloud().

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

Definition at line 71 of file pc\_manager.cpp.

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

Here is the call graph for this function:



#### 5.1.3.17 vector< PCLCloudPtr > pcm::PCManager::getCloudFromIdx ( PCLCloudPtr *originalCloud*, PrimitveIdxPtr *indices* ) [static]

Definition at line 181 of file pc\_manager.cpp.

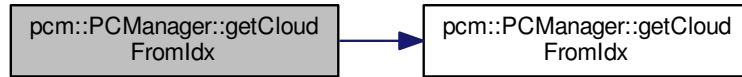
Referenced by getCloudFromIdx().

#### 5.1.3.18 vector< PCLCloudPtr > pcm::PCManager::getCloudFromIdx ( PrimitveIdxPtr *indices* )

Definition at line 235 of file pc\_manager.cpp.

References `getCloudFromIdx()`, and `originalCloud`.

Here is the call graph for this function:



#### 5.1.3.19 `string pcm::PCManager::getFomrattedData ( ) [static]`

Definition at line 121 of file `pc_manager.cpp`.

Referenced by `srvn::getPathParameter()`.

#### 5.1.3.20 `PCLCloudPtr pcm::PCManager::getOriginalCloud ( )`

Definition at line 289 of file `pc_manager.cpp`.

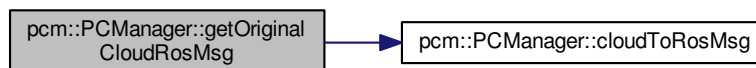
References `originalCloud`.

#### 5.1.3.21 `PointCloud2 pcm::PCManager::getOriginalCloudRosMsg ( )`

Definition at line 292 of file `pc_manager.cpp`.

References `cloudToRosMsg()`, and `originalCloud`.

Here is the call graph for this function:



#### 5.1.3.22 `PCLNormalPtr pcm::PCManager::getOriginalNormal ( )`

Definition at line 296 of file `pc_manager.cpp`.

References `originalNorms`.

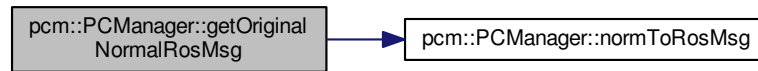
#### 5.1.3.23 `PointCloud2 pcm::PCManager::getOriginalNormalRosMsg ( )`

Definition at line 299 of file `pc_manager.cpp`.

References `normToRosMsg()`, and `originalNorms`.



Here is the call graph for this function:



#### 5.1.3.24 PCPrimitivePtr pcm::PCManager::getPrimitiveShape ( int *idx* )

#### 5.1.3.25 PCLVisualizer pcm::PCManager::getVisor ( )

Definition at line 306 of file pc\_manager.cpp.

References visor.

#### 5.1.3.26 bool pcm::PCManager::getVisualizationFlag ( )

Definition at line 303 of file pc\_manager.cpp.

References visualizationFlag.

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

Definition at line 347 of file pc\_manager.cpp.

References createVisor(), DEFAULT\_VISUALIZER\_TITLE, visor, and visualizationFlag.

Referenced by PCManager().

Here is the call graph for this function:



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

Definition at line 105 of file pc\_manager.cpp.

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

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

Definition at line 100 of file pc\_manager.cpp.

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

#### 5.1.3.30 `PointCloud2 pcm::PCManager::normToRosMsg ( PCLNormalPtr input ) [static]`

Definition at line 95 of file `pc_manager.cpp`.

Referenced by `getOriginalNormalRosMsg()`.

#### 5.1.3.31 `void pcm::PCManager::setOriginalCloud ( PCLCloudPtr cloud )`

Definition at line 312 of file `pc_manager.cpp`.

References `DEFAULT_DOWSEAMPLIG_RATE`, and `DEFAULT_NORM_SEARCH`.

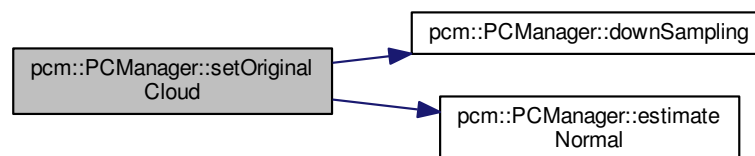
Referenced by `setOriginalCloud()`.

#### 5.1.3.32 `void pcm::PCManager::setOriginalCloud ( PCLCloudPtr cloud, int normSearch, float downSpanX, float downSpanY, float downSpanZ )`

Definition at line 315 of file `pc_manager.cpp`.

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

Here is the call graph for this function:

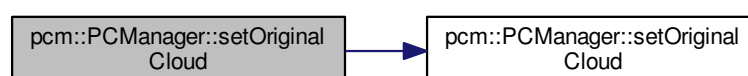


#### 5.1.3.33 `void pcm::PCManager::setOriginalCloud ( PointCloud2Ptr cloud )`

Definition at line 323 of file `pc_manager.cpp`.

References `DEFAULT_DOWSEAMPLIG_RATE`, `DEFAULT_NORM_SEARCH`, and `setOriginalCloud()`.

Here is the call graph for this function:

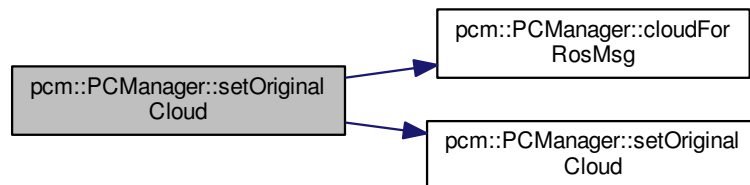


5.1.3.34 `void pcm::PCManager::setOriginalCloud ( PointCloud2Ptr cloud, int normSearch, float downSpanX, float downSpanY, float downSpanZ )`

Definition at line 326 of file `pc_manager.cpp`.

References `cloudForRosMsg()`, and `setOriginalCloud()`.

Here is the call graph for this function:



5.1.3.35 `void pcm::PCManager::setVisualizationFlag ( bool flag )`

Definition at line 330 of file `pc_manager.cpp`.

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

Here is the call graph for this function:



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

Definition at line 152 of file `pc_manager.cpp`.

References `VISUALIZER_POINT_SIZE`.

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

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

Definition at line 159 of file `pc_manager.cpp`.

References `updateVisor()`.

Here is the call graph for this function:



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

Definition at line 163 of file pc\_manager.cpp.

References DEFAULT\_NORM\_LEVEL, DEFAULT\_NORM\_NAME\_SUFFIX, DEFAULT\_NORM\_SCALE, and VISUALIZER\_POINT\_SIZE.

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

Definition at line 172 of file pc\_manager.cpp.

References updateVisor().

Here is the call graph for this function:



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

Definition at line 139 of file pc\_manager.cpp.

References VISUALIZER\_POINT\_SIZE\_BIG.

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

Definition at line 148 of file pc\_manager.cpp.

References updateVisor().

Here is the call graph for this function:



5.1.3.42 `void pcm::PCManager::visualize ( )`

5.1.3.43 `bool pcm::PCManager::writeToFile ( string txt, string filePath, bool append ) [static]`

Definition at line 215 of file `pc_manager.cpp`.

## 5.1.4 Member Data Documentation

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

Definition at line 112 of file `pc_manager.h`.

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

Definition at line 120 of file `pc_manager.h`.

Referenced by `downSampling()`, and `setOriginalCloud()`.

5.1.4.3 `const int pcm::PCManager::DEFAULT_NORM_LEVEL = 5 [static]`

Definition at line 115 of file `pc_manager.h`.

Referenced by `updateVisor()`.

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

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

Referenced by `updateVisor()`.

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

Definition at line 116 of file `pc_manager.h`.

Referenced by `updateVisor()`.

5.1.4.6 `const int pcm::PCManager::DEFAULT_NORM_SEARCH = 50 [static]`

Definition at line 119 of file `pc_manager.h`.

Referenced by `estimateNormal()`, and `setOriginalCloud()`.

**5.1.4.7** `const string pcm::PCManager::DEFAULT_ORIGINAL_CLOUD_VIEWER_NAME = "original" [static]`

Definition at line 114 of file pc\_manager.h.

**5.1.4.8** `const bool pcm::PCManager::DEFAULT_VISUALIZATION_FLAG = false [static]`

Definition at line 109 of file pc\_manager.h.

Referenced by PCManager().

**5.1.4.9** `const string pcm::PCManager::DEFAULT_VISUALIZER_TITLE = "PointCloud manager" [static]`

Definition at line 117 of file pc\_manager.h.

Referenced by initialize(), and setVisualizationFlag().

**5.1.4.10** `PCLCloudPtr pcm::PCManager::originalCloud [private]`

Definition at line 29 of file pc\_manager.h.

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

**5.1.4.11** `PCLNormalPtr pcm::PCManager::originalNorms [private]`

Definition at line 30 of file pc\_manager.h.

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

**5.1.4.12** `vector< PCPrimitivePtr> pcm::PCManager::primitiveList [private]`

Definition at line 35 of file pc\_manager.h.

**5.1.4.13** `PCLVisualizer pcm::PCManager::visor [private]`

Definition at line 32 of file pc\_manager.h.

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

**5.1.4.14** `bool pcm::PCManager::visualizationFlag [private]`

Definition at line 33 of file pc\_manager.h.

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

**5.1.4.15** `const int pcm::PCManager::VISUALIZER_POINT_SIZE = 3 [static]`

Definition at line 110 of file pc\_manager.h.

Referenced by updateVisor().

**5.1.4.16** `const int pcm::PCManager::VISUALIZER_POINT_SIZE_BIG = 10 [static]`

Definition at line 111 of file pc\_manager.h.

Referenced by updateVisor().

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

- [src/point\\_cloud\\_library/pc\\_manager.h](#)
- [src/point\\_cloud\\_library/pc\\_manager.cpp](#)

## 5.2 pcp::PCPrimitive Class Reference

```
#include <pc_primitive.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](#)

#### 5.2.1 Detailed Description

Definition at line 27 of file [pc\\_primitive.h](#).

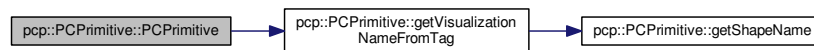
## 5.2.2 Constructor & Destructor Documentation

### 5.2.2.1 `pcp::PCPrimitive::PCPrimitive ( string shapename, int shapeMapIdx, bool visualFlag, PCLCloudPtr cloud, PCLNormalPtr norms )`

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

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

Here is the call graph for this function:



### 5.2.2.2 `pcp::PCPrimitive::~~PCPrimitive ( ) [virtual]`

Definition at line 26 of file `pc_primitive.cpp`.

## 5.2.3 Member Function Documentation

### 5.2.3.1 `ModelCoefficients pcp::PCPrimitive::copyCoefficients ( ModelCoefficients::Ptr input ) [private]`

Definition at line 64 of file `pc_primitive.cpp`.

### 5.2.3.2 `PCLCloud pcp::PCPrimitive::getPrimitiveCloud ( )`

Definition at line 88 of file `pc_primitive.cpp`.

References `primitiveCloud`.

### 5.2.3.3 `PCLNormal pcp::PCPrimitive::getPrimitiveNormal ( )`

Definition at line 91 of file `pc_primitive.cpp`.

References `primitiveNormals`.

### 5.2.3.4 `int pcp::PCPrimitive::getShapeMapIdx ( )`

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

References `shapeMapIdx`.

### 5.2.3.5 `string pcp::PCPrimitive::getShapeName ( )`

Definition at line 73 of file `pc_primitive.cpp`.

References `shapeName`.

Referenced by `getVisualizationNameFromTag()`.



### 5.2.3.6 bool pcp::PCPrimitive::getVisualizationFlag ( )

Definition at line 79 of file pc\_primitive.cpp.

References visualizationFlag.

### 5.2.3.7 string pcp::PCPrimitive::getVisualizationName ( )

Definition at line 76 of file pc\_primitive.cpp.

References visualizationName.

### 5.2.3.8 string pcp::PCPrimitive::getVisualizationNameFromTag ( int idx ) [private]

Definition at line 35 of file pc\_primitive.cpp.

References DEFAULT\_VISUALIZATION\_NAME\_SEPARATOR, and getShapeName().

Referenced by PCPrimitive().

Here is the call graph for this function:



## 5.2.4 Member Data Documentation

### 5.2.4.1 const string pcp::PCPrimitive::DEFAULT\_SHAPE\_NAME\_CLUSTER = "cluster" [static]

Definition at line 74 of file pc\_primitive.h.

### 5.2.4.2 const string pcp::PCPrimitive::DEFAULT\_SHAPE\_NAME\_PLANE = "plane" [static]

Definition at line 73 of file pc\_primitive.h.

### 5.2.4.3 const string pcp::PCPrimitive::DEFAULT\_VISUALIZATION\_NAME\_SEPARATOR = "-" [static]

Definition at line 76 of file pc\_primitive.h.

Referenced by getVisualizationNameFromTag().

### 5.2.4.4 PCLCloud pcp::PCPrimitive::primitiveCloud [private]

Definition at line 37 of file pc\_primitive.h.

Referenced by getPrimitiveCloud(), and PCPrimitive().

### 5.2.4.5 ModelCoefficients pcp::PCPrimitive::primitiveCoefficients [private]

Definition at line 39 of file pc\_primitive.h.

#### 5.2.4.6 PCLNormal pcp::PCPrimitive::primitiveNormals [private]

Definition at line 38 of file pc\_primitive.h.

Referenced by getPrimitiveNormal(), and PCPrimitive().

#### 5.2.4.7 int pcp::PCPrimitive::shapeMapIdx [private]

Definition at line 34 of file pc\_primitive.h.

Referenced by getShapeMapidx(), and PCPrimitive().

#### 5.2.4.8 string pcp::PCPrimitive::shapeName [private]

Definition at line 30 of file pc\_primitive.h.

Referenced by getShapeName(), and PCPrimitive().

#### 5.2.4.9 bool pcp::PCPrimitive::visualizationFlag [private]

Definition at line 32 of file pc\_primitive.h.

Referenced by getVisualizationFlag(), and PCPrimitive().

#### 5.2.4.10 string pcp::PCPrimitive::visualizationName [private]

Definition at line 31 of file pc\_primitive.h.

Referenced by getVisualizationName(), and PCPrimitive().

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

- [src/point\\_cloud\\_library/pc\\_primitive.h](#)
- [src/point\\_cloud\\_library/pc\\_primitive.cpp](#)

## 5.3 vector3d Struct Reference

### Public Attributes

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

#### 5.3.1 Detailed Description

Definition at line 34 of file cone\_segmentation\_srv.cpp.

#### 5.3.2 Member Data Documentation

##### 5.3.2.1 float vector3d::x

Definition at line 35 of file cone\_segmentation\_srv.cpp.

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

#### 5.3.2.2 float vector3d::y

Definition at line 36 of file cone\_segmentation\_srv.cpp.

Referenced by `getNormalizeAxesDirectionVector()`, `getPointOnAxes()`, `getVectorBetweenPoints()`, `ransacConeDetection()`, and `ransacCylinderDetection()`.

#### 5.3.2.3 float vector3d::z

Definition at line 37 of file cone\_segmentation\_srv.cpp.

Referenced by `getNormalizeAxesDirectionVector()`, `getPointOnAxes()`, `getVectorBetweenPoints()`, `ransacConeDetection()`, and `ransacCylinderDetection()`.

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

- [src/segmentation\\_services/cone\\_segmentation\\_srv.cpp](#)
- [src/segmentation\\_services/cylinder\\_segmentation\\_srv.cpp](#)



## Chapter 6

# File Documentation

### 6.1 src/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 "pitt_msgs/DeepFilter.h"
#include "pitt_msgs/SupportSegmentation.h"
#include "pitt_msgs/ClusterSegmentation.h"
#include "pitt_msgs/ArmFilter.h"
#include "pitt_msgs/Support.h"
#include "pitt_msgs/InliersCluster.h"
#include "pitt_msgs/ClustersOutput.h"
#include "point_cloud_library/pc_manager.h"
#include "point_cloud_library/srv_manager.h"
#include <boost/thread.hpp>
#include <boost/thread/mutex.hpp>
#include <boost/format.hpp>
```

Include dependency graph for obj\_segmentation.cpp:



### Typedefs

- typedef boost::shared\_ptr< vector< Support > > [InlierSupportsPtr](#)
- typedef vector< Support > [InlierSupports](#)
- typedef boost::shared\_ptr< vector< InliersCluster > > [InlierClusterPtr](#)
- typedef vector< InliersCluster > [InlierClusters](#)

## Functions

- void `visSpin` ()
- bool `callDeepFilter` (PCLCloudPtr &cloud)  
*BBBB.*
- bool `callArmFilter` (PCLCloudPtr &cloud)
- InlierSupportsPtr `callSupportFilter` (PCLCloudPtr inputCloud, PCLNormalPtr normal)
- InlierClusterPtr `callClusterSegmentation` (PCLCloudPtr cloud)
- void `depthAcquisition` (const PointCloud2Ptr &input)
- int `main` (int argc, char \*\*argv)

## Variables

- ros::NodeHandle \* `nh_ptr` = NULL
- bool `inputShowSupportClouds`
- bool `inputShowOriginalCloud`
- bool `inputShowClusterClouds`
- bool `inputShowObjectOnSupport`
- string `centroidLogFilePath`
- string `log_str_depth` = "Loading..."
- string `log_str_supp` = "Loading..."
- long `scanId` = 0
- static const int `MIN_POINT_IN_ORIGINAL_CLOUD` = 30
- pcm::PCManager \* `manager` = new pcm::PCManager( false)
- boost::shared\_ptr  
   < visualization::PCLVisualizer > `vis`
- Publisher `clusterPub`
- boost::thread `vis_thread`
- boost::mutex `vis_mutex`
- Eigen::Matrix4f `pclTransform`

### 6.1.1 Typedef Documentation

#### 6.1.1.1 typedef boost::shared\_ptr< vector< InliersCluster> > InlierClusterPtr

Definition at line 40 of file `obj_segmentation.cpp`.

#### 6.1.1.2 typedef vector< InliersCluster> InlierClusters

Definition at line 41 of file `obj_segmentation.cpp`.

#### 6.1.1.3 typedef vector< Support> InlierSupports

Definition at line 39 of file `obj_segmentation.cpp`.

#### 6.1.1.4 typedef boost::shared\_ptr< vector< Support> > InlierSupportsPtr

Definition at line 38 of file `obj_segmentation.cpp`.

## 6.1.2 Function Documentation

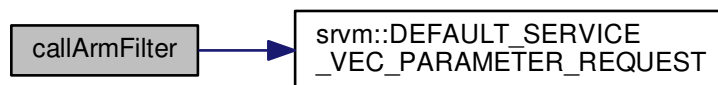
### 6.1.2.1 bool callArmFilter ( PCLCloudPtr & cloud )

Definition at line 112 of file obj\_segmentation.cpp.

References `srvm::DEFAULT_SERVICE_VEC_PARAMETER_REQUEST()`, `nh_ptr`, `srvm::PARAM_NAME_ARM_SRV_MAX_ELBOW_BOX`, `srvm::PARAM_NAME_ARM_SRV_MAX_FOREARM_BOX`, `srvm::PARAM_NAME_ARM_SRV_MIN_ELBOW_BOX`, `srvm::PARAM_NAME_ARM_SRV_MIN_FOREARM_BOX`, and `srvm::SRV_NAME_ARM_FILTER`.

Referenced by `depthAcquisition()`.

Here is the call graph for this function:



### 6.1.2.2 InlierClusterPtr callClusterSegmentation ( PCLCloudPtr cloud )

Definition at line 214 of file obj\_segmentation.cpp.

References `nh_ptr`, and `srvm::SRV_NAME_CUSTER_FILTER`.

Referenced by `depthAcquisition()`.

### 6.1.2.3 bool callDeepFilter ( PCLCloudPtr & cloud )

BBBB.

AAAAA

Parameters

<i>cloud</i>	
--------------	--

Returns

Definition at line 79 of file obj\_segmentation.cpp.

References `srvm::DEFAULT_SERVICE_PARAMETER_REQUEST_F`, `inputShowClusterClouds`, `inputShowObjectOnSupport`, `inputShowOriginalCloud`, `inputShowSupportClouds`, `log_str_depth`, `nh_ptr`, `srvm::PARAM_NAME_DEEP_SRV_Z_THRESHOLD`, `srvm::SRV_NAME_DEEP_FILTER`, `vis`, and `vis_mutex`.

Referenced by `depthAcquisition()`.

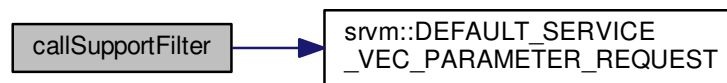
### 6.1.2.4 InlierSupportsPtr callSupportFilter ( PCLCloudPtr inputCloud, PCLNormalPtr normal )

Definition at line 147 of file obj\_segmentation.cpp.

References `srvm::DEFAULT_SERVICE_PARAMETER_REQUEST`, `srvm::DEFAULT_SERVICE_PARAMETER_REQUEST_F`, `srvm::DEFAULT_SERVICE_VEC_PARAMETER_REQUEST()`, `horizontalAxis`, `inputShowClusterClouds`, `inputShowObjectOnSupport`, `inputShowOriginalCloud`, `inputShowSupportClouds`, `log_str_supp`, `nh_ptr`, `srvm::PARAM_NAME_HORIZONTAL_AXIS`, `srvm::PARAM_NAME_HORIZONTAL_VARIANCE_THRESHOLD`, `srvm::PARAM_NAME_MIN_ITERATIVE_CLOUD_PERCENTAGE`, `srvm::PARAM_NAME_MIN_ITERATIVE_SUPPORT_PERCENTAGE`, `srvm::PARAM_NAME_RANSAC_IN_SHAPE_DISTANCE_POINT_THRESHOLD`, `srvm::PARAM_NAME_RANSAC_MAX_ITERATION_THRESHOLD`, `srvm::PARAM_NAME_RANSAC_MODEL_NORMAL_DISTANCE_WEIGHT`, `srvm::PARAM_NAME_SUPPORT_EDGE_REMOVE_OFFSET`, `srvm::SRV_NAME_SUPPORT_FILTER`, `vis`, and `vis_mutex`.

Referenced by `depthAcquisition()`.

Here is the call graph for this function:



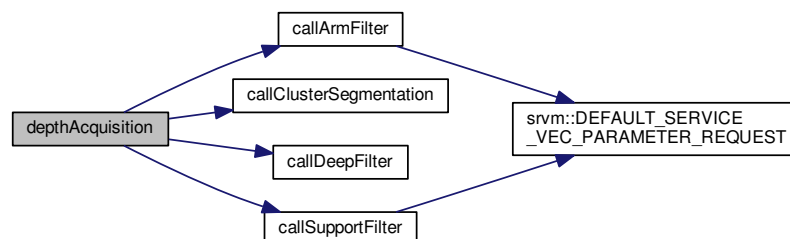
#### 6.1.2.5 void depthAcquisition ( const PointCloud2Ptr & input )

Definition at line 234 of file `obj_segmentation.cpp`.

References `callArmFilter()`, `callClusterSegmentation()`, `callDeepFilter()`, `callSupportFilter()`, `centroidFileLog`, `centroidLogFilePath`, `clusterPub`, `inputShowClusterClouds`, `inputShowObjectOnSupport`, `inputShowOriginalCloud`, `inputShowSupportClouds`, `MIN_POINT_IN_ORIGINAL_CLOUD`, `pclTransform`, `scanId`, `vis`, and `vis_mutex`.

Referenced by `main()`.

Here is the call graph for this function:



#### 6.1.2.6 int main ( int argc, char \*\* argv )

This method implements the main node loop and it spins as soon as a new data is available in the input topic. Particularly, the input topic can be specified through its name into the parameter .....



## Parameters

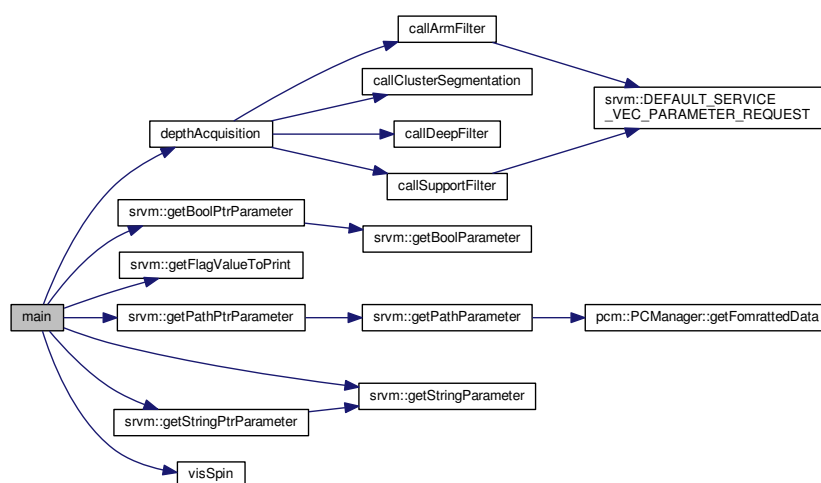
<i>argc</i>	
<i>argv</i>	

## Returns

Definition at line 337 of file obj\_segmentation.cpp.

References centroidLogFilePath, clusterPub, srvm::DEFAULT\_INPUT\_PARAM\_CENTROID\_LOG\_FILE, srvm::DEFAULT\_INPUT\_PARAM\_RAW\_CLOUD\_TOPIC, srvm::DEFAULT\_INPUT\_PARAM\_SHOW\_CLUSTERS, srvm::DEFAULT\_INPUT\_PARAM\_SHOW\_OBJECT\_ON\_SUPPORT, srvm::DEFAULT\_INPUT\_PARAM\_SHOW\_ORIGINAL\_CLOUD, srvm::DEFAULT\_INPUT\_PARAM\_SHOW\_SUPPORTS, srvm::DEFAULT\_PARAM\_INPUT\_CLOUD\_REFERENCE\_FRAME, srvm::DEFAULT\_PARAM\_OUTPUT\_CLOUD\_REFERENCE\_FRAME, srvm::DEFAULT\_TF\_WAIT\_SECONDS, depthAcquisition(), srvm::getBoolPtrParameter(), srvm::getFlagValueToPrint(), srvm::getPathPtrParameter(), srvm::getStringParameter(), srvm::getStringPtrParameter(), inputShowClusterClouds, inputShowObjectOnSupport, inputShowOriginalCloud, inputShowSupportClouds, log\_str\_depth, log\_str\_supp, nh\_ptr, srvm::PARAM\_NAME\_INPUT\_CLOUD\_REFERENCE\_FRAME, srvm::PARAM\_NAME\_OUTPUT\_CLOUD\_REFERENCE\_FRAME, pclTransform, srvm::TOPIC\_OUT\_NAME\_OBJECT\_PERCEPTION, vis, vis\_thread, and visSpin().

Here is the call graph for this function:



## 6.1.2.7 void visSpin ( )

Definition at line 64 of file obj\_segmentation.cpp.

References vis, and vis\_mutex.

Referenced by main().

## 6.1.3 Variable Documentation

## 6.1.3.1 string centroidLogFilePath

Definition at line 45 of file obj\_segmentation.cpp.

Referenced by `depthAcquisition()`, and `main()`.

#### 6.1.3.2 Publisher `clusterPub`

Definition at line 60 of file `obj_segmentation.cpp`.

Referenced by `depthAcquisition()`, and `main()`.

#### 6.1.3.3 `bool inputShowClusterClouds`

Definition at line 44 of file `obj_segmentation.cpp`.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, and `main()`.

#### 6.1.3.4 `bool inputShowObjectOnSupport`

Definition at line 44 of file `obj_segmentation.cpp`.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, and `main()`.

#### 6.1.3.5 `bool inputShowOriginalCloud`

Definition at line 44 of file `obj_segmentation.cpp`.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, and `main()`.

#### 6.1.3.6 `bool inputShowSupportClouds`

Definition at line 44 of file `obj_segmentation.cpp`.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, and `main()`.

#### 6.1.3.7 `string log_str_depth = "Loading..."`

Definition at line 48 of file `obj_segmentation.cpp`.

Referenced by `callDeepFilter()`, and `main()`.

#### 6.1.3.8 `string log_str_supp = "Loading..."`

Definition at line 49 of file `obj_segmentation.cpp`.

Referenced by `callSupportFilter()`, and `main()`.

#### 6.1.3.9 `pcm::PCManager* manager = new pcm::PCManager( false)`

Definition at line 58 of file `obj_segmentation.cpp`.

#### 6.1.3.10 `const int MIN_POINT_IN_ORIGINAL_CLOUD = 30` `[static]`

Definition at line 55 of file `obj_segmentation.cpp`.

Referenced by `depthAcquisition()`.

**6.1.3.11** `ros::NodeHandle* nh_ptr = NULL`

Definition at line 36 of file obj\_segmentation.cpp.

Referenced by `callArmFilter()`, `callClusterSegmentation()`, `callDeepFilter()`, `callSupportFilter()`, and `main()`.

**6.1.3.12** `Eigen::Matrix4f pclTransform`

Definition at line 232 of file obj\_segmentation.cpp.

Referenced by `depthAcquisition()`, and `main()`.

**6.1.3.13** `long scanId = 0`

Definition at line 52 of file obj\_segmentation.cpp.

Referenced by `depthAcquisition()`.

**6.1.3.14** `boost::shared_ptr< visualization::PCLVisualizer> vis`

Definition at line 59 of file obj\_segmentation.cpp.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, `main()`, and `visSpin()`.

**6.1.3.15** `boost::mutex vis_mutex`

Definition at line 62 of file obj\_segmentation.cpp.

Referenced by `callDeepFilter()`, `callSupportFilter()`, `depthAcquisition()`, and `visSpin()`.

**6.1.3.16** `boost::thread vis_thread`

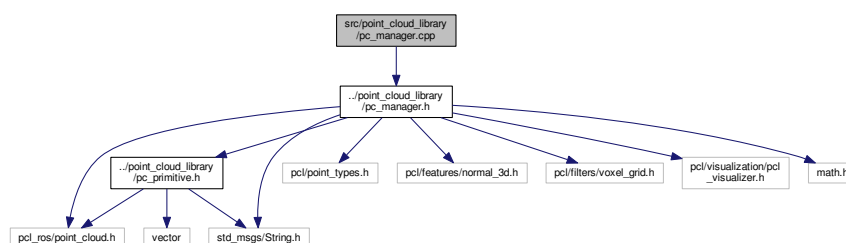
Definition at line 61 of file obj\_segmentation.cpp.

Referenced by `main()`.

**6.2** `src/point_cloud_library/pc_manager.cpp` File Reference

```
#include "../point_cloud_library/pc_manager.h"
```

Include dependency graph for `pc_manager.cpp`:





## Typedefs

- typedef boost::shared\_ptr  
   < [pcp::PCPrimitive](#) > [PCPrimitivePtr](#)
- typedef boost::shared\_ptr  
   < visualization::PCLVisualizer > [PCLVisualizer](#)

### 6.3.1 Typedef Documentation

#### 6.3.1.1 typedef boost::shared\_ptr< visualization::PCLVisualizer> PCLVisualizer

Definition at line 23 of file pc\_manager.h.

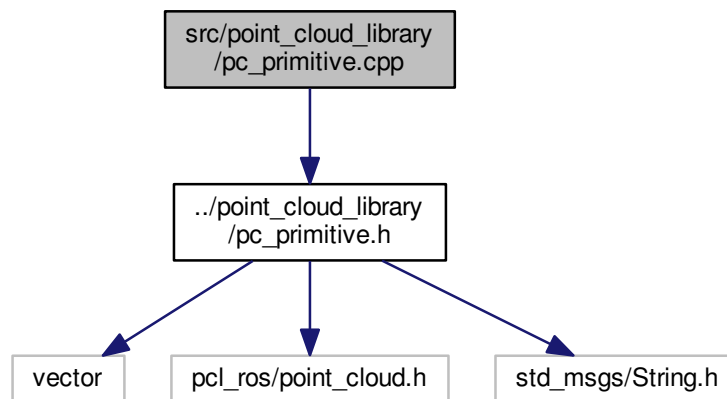
#### 6.3.1.2 typedef boost::shared\_ptr< [pcp::PCPrimitive](#)> PCPrimitivePtr

Definition at line 22 of file pc\_manager.h.

## 6.4 src/point\_cloud\_library/pc\_primitive.cpp File Reference

```
#include "../point_cloud_library/pc_primitive.h"
```

Include dependency graph for pc\_primitive.cpp:



## Namespaces

- [pcp](#)

## 6.5 src/point\_cloud\_library/pc\_primitive.h File Reference

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



6.5.1.2 `typedef pcl::PointCloud< pcl::PointXYZ>::Ptr PCLCloudPtr`

Definition at line 21 of file `pc_primitive.h`.

6.5.1.3 `typedef pcl::PointCloud< pcl::Normal> PCLNormal`

Definition at line 22 of file `pc_primitive.h`.

6.5.1.4 `typedef pcl::PointCloud< pcl::Normal>::Ptr PCLNormalPtr`

Definition at line 23 of file `pc_primitive.h`.

6.5.1.5 `typedef std::vector< int> Primitiveldx`

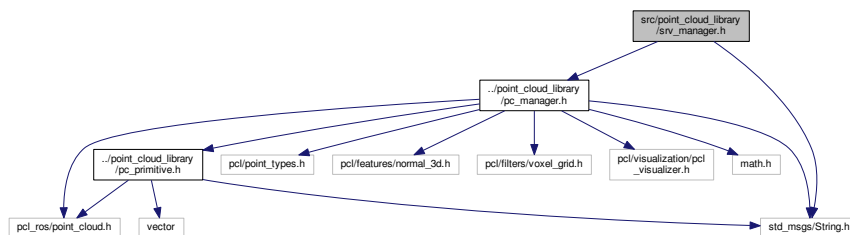
Definition at line 18 of file `pc_primitive.h`.

6.5.1.6 `typedef boost::shared_ptr< std::vector< int> > PrimitiveldxPtr`

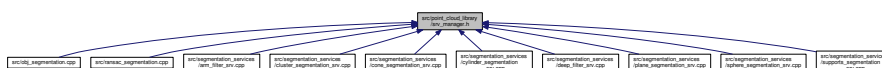
Definition at line 19 of file `pc_primitive.h`.

## 6.6 src/point\_cloud\_library/srv\_manager.h File Reference

```
#include <std_msgs/String.h>
#include "../point_cloud_library/pc_manager.h"
Include dependency graph for srv_manager.h:
```



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



## Namespaces

- [srvm](#)

## Functions

- const vector< float > [srvm::DEFAULT\\_SERVICE\\_VEC\\_PARAMETER\\_REQUEST](#) (DEFAULT\_SERVICE\_ARRAY\_PARAMETER\_REQUEST, DEFAULT\_SERVICE\_ARRAY\_PARAMETER\_REQUEST+sizeof(DEFAULT\_SERVICE\_ARRAY\_PARAMETER\_REQUEST)/sizeof(float))
- string [srvm::getStringParameter](#) (string input, const string defaultValue)
- string [srvm::getStringPtrParameter](#) (char \*input, const string defaultValue)
- bool [srvm::getBoolParameter](#) (string input, const bool defaultValue)
- bool [srvm::getBoolPtrParameter](#) (char \*input, const bool defaultValue)
- string [srvm::getPathParameter](#) (string input, const string defaultValue)
- string [srvm::getPathPtrParameter](#) (char \*input, const string defaultValue)
- float [srvm::getServiceFloatParameter](#) (float input, const float defaultValue)
- int [srvm::getServiceIntParameter](#) (int input, const int defaultValue)
- string [srvm::getServiceStringParameter](#) (string input, const string defaultValue)
- vector< float > [srvm::getService3DArrayParameter](#) (vector< float > input, const vector< float > defaultValue)
- vector< float > [srvm::get3DArray](#) (const float values[])
- vector< float > [srvm::getService3DArrayParameter](#) (vector< float > input, const float defaultValue[])
- string [srvm::getFlagValueToPrint](#) (bool flag)
- string [srvm::getArrayToPrint](#) (vector< float > arr)

## Variables

- const string [srvm::SRV\\_NAME\\_DEEP\\_FILTER](#) = "deep\_filter\_srv"
- const string [srvm::SRV\\_NAME\\_SUPPORT\\_FILTER](#) = "support\_segmentation\_srv"
- const string [srvm::SRV\\_NAME\\_CUSTER\\_FILTER](#) = "cluster\_Segmentation\_srv"
- const string [srvm::SRV\\_NAME\\_ARM\\_FILTER](#) = "arm\_filter\_srv"
- const string [srvm::SRV\\_NAME\\_RANSAC\\_SPHERE\\_FILTER](#) = "sphere\_segmentation\_srv"
- const string [srvm::SRV\\_NAME\\_RANSAC\\_CYLINDER\\_FILTER](#) = "cylinder\_segmentation\_srv"
- const string [srvm::SRV\\_NAME\\_RANSAC\\_CONE\\_FILTER](#) = "cone\_segmentation\_srv"
- const string [srvm::SRV\\_NAME\\_RANSAC\\_PLANE\\_FILTER](#) = "plane\_segmentation\_srv"
- const string [srvm::PARAM\\_NAME\\_INPUT\\_CLOUD\\_REFERENCE\\_FRAME](#) = "/pitt/ref\_frame/input\_cloud"
- const string [srvm::PARAM\\_NAME\\_OUTPUT\\_CLOUD\\_REFERENCE\\_FRAME](#) = "/pitt/ref\_frame/output\_cloud"
- const string [srvm::PARAM\\_NAME\\_DEEP\\_SRV\\_Z\\_THRESHOLD](#) = "/pitt/service/deep\_filter/z\_threshold"
- const string [srvm::PARAM\\_NAME\\_ARM\\_SRV\\_MIN\\_FOREARM\\_BOX](#) = "/pitt/srv/arm\_filter/min\_forearm\_box"
- const string [srvm::PARAM\\_NAME\\_ARM\\_SRV\\_MAX\\_FOREARM\\_BOX](#) = "/pitt/srv/arm\_filter/max\_forearm\_box"
- const string [srvm::PARAM\\_NAME\\_ARM\\_SRV\\_MIN\\_ELBOW\\_BOX](#) = "/pitt/srv/arm\_filter/min\_elbow\_box"
- const string [srvm::PARAM\\_NAME\\_ARM\\_SRV\\_MAX\\_ELBOW\\_BOX](#) = "/pitt/srv/arm\_filter/max\_elbow\_box"
- const string [srvm::PARAM\\_NAME\\_CLUSTER\\_TOLERANCE](#) = "/pitt/srv/cluster\_segmentation/tolerance"
- const string [srvm::PARAM\\_NAME\\_CLUSTER\\_MIN\\_RATE](#) = "/pitt/srv/cluster\_segmentation/min\_rate"
- const string [srvm::PARAM\\_NAME\\_CLUSTER\\_MAX\\_RATE](#) = "/pitt/srv/cluster\_segmentation/max\_rate"
- const string [srvm::PARAM\\_NAME\\_CLUSTER\\_MIN\\_INPUT\\_SIZE](#) = "/pitt/srv/cluster\_segmentation/min\_input\_size"
- const string [srvm::PARAM\\_NAME\\_SPHERE\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = "/pitt/srv/sphere\_segmentation/normal\_distance\_weight"
- const string [srvm::PARAM\\_NAME\\_SPHERE\\_DISTANCE\\_TH](#) = "/pitt/srv/sphere\_segmentation/distance\_th"
- const string [srvm::PARAM\\_NAME\\_SPHERE\\_MAX\\_ITERATION\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/max\_iter\_limit"
- const string [srvm::PARAM\\_NAME\\_SPHERE\\_MIN\\_RADIUS\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/min\_radius\_limit"
- const string [srvm::PARAM\\_NAME\\_SPHERE\\_MAX\\_RADIUS\\_LIMIT](#) = "/pitt/srv/sphere\_segmentation/max\_radius\_limit"



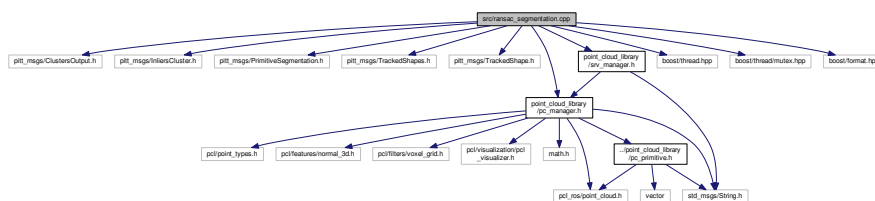
- `const string srvvm::PARAM_NAME_SPHERE_EPS_ANGLE_TH = "/pitt/srv/sphere_segmentation/eps_angle_th"`
- `const string srvvm::PARAM_NAME_SPHERE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/sphere_segmentation/min_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_SPHERE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/sphere_segmentation/max_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_SPHERE_MIN_INLIERS = "/pitt/srv/sphere_segmentation/min_inliers"`
- `const string srvvm::PARAM_NAME_CYLINDER_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/cylinder_segmentation/normal_distance_weight"`
- `const string srvvm::PARAM_NAME_CYLINDER_DISTANCE_TH = "/pitt/srv/cylinder_segmentation/distance_th"`
- `const string srvvm::PARAM_NAME_CYLINDER_MAX_ITERATION_LIMIT = "/pitt/srv/cylinder_segmentation/max_iter_limit"`
- `const string srvvm::PARAM_NAME_CYLINDER_MIN_RADIUS_LIMIT = "/pitt/srv/cylinder_segmentation/min_radius_limit"`
- `const string srvvm::PARAM_NAME_CYLINDER_MAX_RADIUS_LIMIT = "/pitt/srv/cylinder_segmentation/max_radius_limit"`
- `const string srvvm::PARAM_NAME_CYLINDER_EPS_ANGLE_TH = "/pitt/srv/cylinder_segmentation/eps_angle_th"`
- `const string srvvm::PARAM_NAME_CYLINDER_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/cylinder_segmentation/min_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_CYLINDER_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/cylinder_segmentation/max_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_CYLINDER_MIN_INLIERS = "/pitt/srv/cylinder_segmentation/min_inliers"`
- `const string srvvm::PARAM_NAME_CONE_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/cone_segmentation/normal_distance_weight"`
- `const string srvvm::PARAM_NAME_CONE_DISTANCE_TH = "/pitt/srv/cone_segmentation/distance_th"`
- `const string srvvm::PARAM_NAME_CONE_MAX_ITERATION_LIMIT = "/pitt/srv/cone_segmentation/max_iter_limit"`
- `const string srvvm::PARAM_NAME_CONE_MIN_RADIUS_LIMIT = "/pitt/srv/cone_segmentation/min_radius_limit"`
- `const string srvvm::PARAM_NAME_CONE_MAX_RADIUS_LIMIT = "/pitt/srv/cone_segmentation/max_radius_limit"`
- `const string srvvm::PARAM_NAME_CONE_EPS_ANGLE_TH = "/pitt/srv/cone_segmentation/eps_angle_th"`
- `const string srvvm::PARAM_NAME_CONE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/cone_segmentation/min_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_CONE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/cone_segmentation/max_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_CONE_MIN_INLIERS = "/pitt/srv/cone_segmentation/min_inliers"`
- `const string srvvm::PARAM_NAME_PLANE_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/plane_segmentation/normal_distance_weight"`
- `const string srvvm::PARAM_NAME_PLANE_DISTANCE_TH = "/pitt/srv/plane_segmentation/distance_th"`
- `const string srvvm::PARAM_NAME_PLANE_MAX_ITERATION_LIMIT = "/pitt/srv/plane_segmentation/max_iter_limit"`
- `const string srvvm::PARAM_NAME_PLANE_EPS_ANGLE_TH = "/pitt/srv/plane_segmentation/eps_angle_th"`
- `const string srvvm::PARAM_NAME_PLANE_MIN_OPENING_ANGLE_DEGREE = "/pitt/srv/plane_segmentation/min_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_PLANE_MAX_OPENING_ANGLE_DEGREE = "/pitt/srv/plane_segmentation/max_opening_angle_deg"`
- `const string srvvm::PARAM_NAME_PLANE_MIN_INLIERS = "/pitt/srv/plane_segmentation/min_inliers"`
- `const string srvvm::PARAM_NAME_MIN_ITERATIVE_CLOUD_PERCENTAGE = "/pitt/srv/supports_segmentation/min_iter_cloud_percent"`
- `const string srvvm::PARAM_NAME_MIN_ITERATIVE_SUPPORT_PERCENTAGE = "/pitt/srv/supports_segmentation/min_iter_support_percent"`
- `const string srvvm::PARAM_NAME_HORIZONTAL_VARIANCE_THRESHOLD = "/pitt/srv/supports_segmentation/horizontal_variance_th"`

- `const string srv::PARAM_NAME_RANSAC_IN_SHAPE_DISTANCE_POINT_THRESHOLD = "/pitt/srv/supports-_segmentation/in_shape_distance_th"`
- `const string srv::PARAM_NAME_RANSAC_MODEL_NORMAL_DISTANCE_WEIGHT = "/pitt/srv/supports-_segmentation/normal_distance_weight"`
- `const string srv::PARAM_NAME_RANSAC_MAX_ITERATION_THRESHOLD = "/pitt/srv/supports-_segmentation/max_iter"`
- `const string srv::PARAM_NAME_HORIZONTAL_AXIS = "/pitt/srv/supports_segmentation/horizontal_axis"`
- `const string srv::PARAM_NAME_SUPPORT_EDGE_REMOVE_OFFSET = "/pitt/srv/supports_segmentation/edge-_remove_offset"`
- `const string srv::DEFAULT_PARAM_INPUT_CLOUD_REFERENCE_FRAME = "/camera_depth_optical _frame"`
- `const string srv::DEFAULT_PARAM_OUTPUT_CLOUD_REFERENCE_FRAME = "/world"`
- `const string srv::DEFAULT_INPUT_PARAM_RAW_CLOUD_TOPIC = "/camera/depth/points"`
- `const string srv::DEFAULT_INPUT_PARAM_CENTROID_LOG_FILE = ""`
- `const bool srv::DEFAULT_INPUT_PARAM_SHOW_ORIGINAL_CLOUD = false`
- `const bool srv::DEFAULT_INPUT_PARAM_SHOW_SUPPORTS = false`
- `const bool srv::DEFAULT_INPUT_PARAM_SHOW_OBJECT_ON_SUPPORT = false`
- `const bool srv::DEFAULT_INPUT_PARAM_SHOW_CLUSTERS = false`
- `const string srv::DEFAULT_PARAM_ARM_SRV_CAMERA_FRAME = "/camera_depth_optical_frame"`
- `const string srv::DEFAULT_PARAM_ARM_SRV_RIGHT_FOREARM_FRAME = "/right_lower_forearm"`
- `const string srv::DEFAULT_PARAM_ARM_SRV_LEFT_FOREARM_FRAME = "/left_lower_forearm"`
- `const string srv::DEFAULT_PARAM_ARM_SRV_RIGHT_ELBOW_FRAME = "/right_lower_elbow"`
- `const string srv::DEFAULT_PARAM_ARM_SRV_LEFT_ELBOW_FRAME = "/left_lower_elbow"`
- `const bool srv::DEFAULT_PARAM_ARM_SRV_SHOW_CLOUDS = false`
- `const string srv::TOPIC_OUT_NAME_OBJECT_PERCEPTION = "obj_segmentation/ClusterOutput"`
- `const int srv::DEFAULT_SERVICE_PARAMETER_REQUEST = -1`
- `const float srv::DEFAULT_SERVICE_PARAMETER_REQUEST_F = -1.0f`
- `const float srv::DEFAULT_SERVICE_ARRAY_PARAMETER_REQUEST [1] = {-1}`
- `const float srv::DEFAULT_TF_WAIT_SECONDS = 2.0f`
- `const string srv::DEFAULT_SYMBOL = "."`

## 6.7 src/ransac\_segmentation.cpp File Reference

```
#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 "point_cloud_library/pc_manager.h"
#include "point_cloud_library/srv_manager.h"
#include <boost/thread.hpp>
#include <boost/thread/mutex.hpp>
#include <boost/format.hpp>
```

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

- void [visSpin](#) ()
- 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)
- void [clustersAcquisition](#) (const [ClustersOutputConstPtr](#) &clusterObj)
- int [main](#) (int argc, char \*\*argv)

## Variables

- [ros::NodeHandle](#) \* [nh\\_ptr](#) = NULL
- boost::shared\_ptr  
    < [visualization::PCLVisualizer](#) > [vis](#)
- boost::thread [vis\\_thread](#)
- boost::mutex [vis\\_mutex](#)
- Publisher [pub](#)
- static const int [DEFAULT\\_SPHERE\\_MIN\\_INLIERS](#) = 40
- static const int [DEFAULT\\_CYLINDER\\_MIN\\_INLIERS](#) = 40
- static const int [DEFAULT\\_CONE\\_MIN\\_INLIERS](#) = 40
- static const int [DEFAULT\\_PLANE\\_MIN\\_INLIERS](#) = 40
- static const float [DEFAULT\\_CONE\\_OVER\\_CYLINDER\\_PRIORITY](#) = 0.9f
- static const bool [DEFAULT\\_SHOW\\_PRIMITIVE](#) = false
- static bool [SHOW\\_PRIMITIVE](#)
- 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
- int [sphereMinInliers](#)
- int [cylinderMinInliers](#)
- int [coneMinInliers](#)
- int [planeMinInliers](#)
- int [coneOverCylinderPriority](#)
- string [centroidFileLog](#)

## 6.7.1 Typedef Documentation

### 6.7.1.1 `typedef vector< InliersCluster> InliersClusters`

Definition at line 20 of file `ransac_segmentation.cpp`.

### 6.7.1.2 `typedef boost::shared_ptr< InliersClusters> InliersClustersPtr`

Definition at line 21 of file `ransac_segmentation.cpp`.

### 6.7.1.3 `typedef boost::shared_ptr< PrimitiveSegmentation> PrimitiveSegmentationPtr`

Definition at line 22 of file `ransac_segmentation.cpp`.

## 6.7.2 Function Documentation

### 6.7.2.1 `bool callRansacConeSegmentation ( PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr & out )`

Definition at line 135 of file `ransac_segmentation.cpp`.

References `coneMinInliers`, `DEFAULT_CONE_MIN_INLIERS`, `nh_ptr`, `srvm::PARAM_NAME_CONE_MIN_INLIERS`, and `srvm::SRV_NAME_RANSAC_CONE_FILTER`.

Referenced by `clustersAcquisition()`.

### 6.7.2.2 `bool callRansacCylinderSegmentation ( PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr & out )`

Definition at line 96 of file `ransac_segmentation.cpp`.

References `cylinderMinInliers`, `DEFAULT_CYLINDER_MIN_INLIERS`, `nh_ptr`, `srvm::PARAM_NAME_CYLINDER_MIN_INLIERS`, and `srvm::SRV_NAME_RANSAC_CYLINDER_FILTER`.

Referenced by `clustersAcquisition()`.

### 6.7.2.3 `bool callRansacPlaneSegmentation ( PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr & out )`

Definition at line 175 of file `ransac_segmentation.cpp`.

References `DEFAULT_PLANE_MIN_INLIERS`, `nh_ptr`, `srvm::PARAM_NAME_PLANE_MIN_INLIERS`, `planeMinInliers`, and `srvm::SRV_NAME_RANSAC_PLANE_FILTER`.

Referenced by `clustersAcquisition()`.

### 6.7.2.4 `bool callRansacSphereSegmentation ( PCLCloudPtr cloud, PCLNormalPtr norm, PrimitiveSegmentationPtr & out )`

Definition at line 58 of file `ransac_segmentation.cpp`.

References `DEFAULT_SPHERE_MIN_INLIERS`, `nh_ptr`, `srvm::PARAM_NAME_SPHERE_MIN_INLIERS`, `sphereMinInliers`, and `srvm::SRV_NAME_RANSAC_SPHERE_FILTER`.

Referenced by `clustersAcquisition()`.

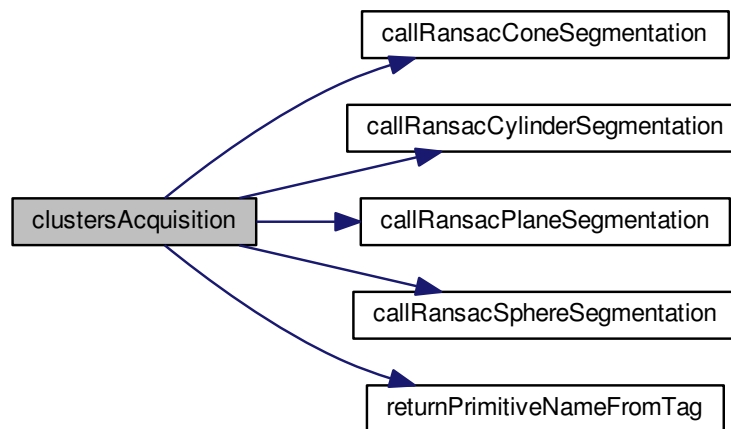
#### 6.7.2.5 void clustersAcquisition ( const ClustersOutputConstPtr & clusterObj )

Definition at line 223 of file ransac\_segmentation.cpp.

References callRansacConeSegmentation(), callRansacCylinderSegmentation(), callRansacPlaneSegmentation(), callRansacSphereSegmentation(), coneMinInliers, cylinderMinInliers, DEFAULT\_CONE\_OVER\_CYLINDER\_PRIORITY, planeMinInliers, pub, returnPrimitiveNameFromTag(), SHOW\_PRIMITIVE, sphereMinInliers, TXT\_CONE\_SHAPE\_TAG, TXT\_CYLINDER\_SHAPE\_TAG, TXT\_PLANE\_SHAPE\_TAG, TXT\_SPHERE\_SHAPE\_TAG, TXT\_UNKNOWN\_SHAPE\_TAG, vis, and vis\_mutex.

Referenced by main().

Here is the call graph for this function:

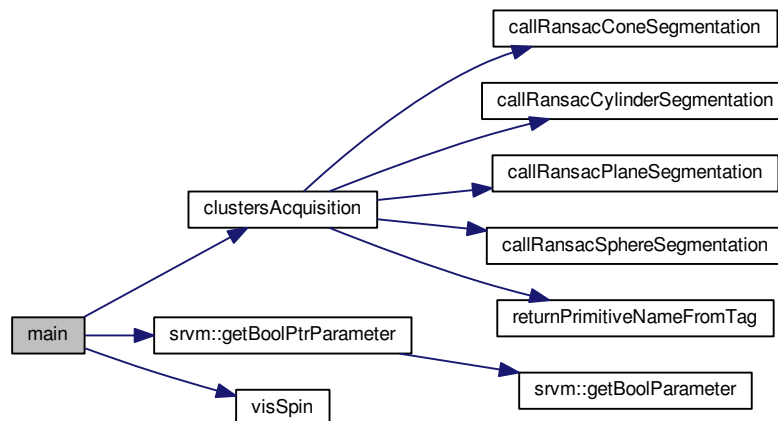


#### 6.7.2.6 int main ( int argc, char \*\* argv )

Definition at line 347 of file ransac\_segmentation.cpp.

References centroidFileLog, clustersAcquisition(), DEFAULT\_CONE\_MIN\_INLIERS, DEFAULT\_CONE\_OVER\_CYLINDER\_PRIORITY, DEFAULT\_CYLINDER\_MIN\_INLIERS, DEFAULT\_PLANE\_MIN\_INLIERS, DEFAULT\_SHOW\_PRIMITIVE, DEFAULT\_SPHERE\_MIN\_INLIERS, srvn::getBoolPtrParameter(), nh\_ptr, pub, SHOW\_PRIMITIVE, vis, vis\_thread, and visSpin().

Here is the call graph for this function:



#### 6.7.2.7 void printConeInfo ( PrimitiveSegmentationPtr info, int idx )

Definition at line 161 of file ransac\_segmentation.cpp.

#### 6.7.2.8 void printCylinderInfo ( PrimitiveSegmentationPtr info, int idx )

Definition at line 122 of file ransac\_segmentation.cpp.

#### 6.7.2.9 void printPlaneInfo ( PrimitiveSegmentationPtr info, int idx )

Definition at line 201 of file ransac\_segmentation.cpp.

#### 6.7.2.10 void printSphereInfo ( PrimitiveSegmentationPtr info, int idx )

Definition at line 86 of file ransac\_segmentation.cpp.

#### 6.7.2.11 string returnPrimitiveNameFromTag ( int primitiveTag )

Definition at line 210 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().

#### 6.7.2.12 void visSpin ( )

Definition at line 50 of file ransac\_segmentation.cpp.

References vis, and vis\_mutex.

Referenced by main().

### 6.7.3 Variable Documentation

#### 6.7.3.1 string centroidFileLog

Definition at line 222 of file ransac\_segmentation.cpp.

Referenced by depthAcquisition(), and main().

#### 6.7.3.2 int coneMinInliers

Definition at line 48 of file ransac\_segmentation.cpp.

Referenced by callRansacConeSegmentation(), and clustersAcquisition().

#### 6.7.3.3 int coneOverCylinderPriority

Definition at line 48 of file ransac\_segmentation.cpp.

#### 6.7.3.4 int cylinderMinInliers

Definition at line 48 of file ransac\_segmentation.cpp.

Referenced by callRansacCylinderSegmentation(), and clustersAcquisition().

#### 6.7.3.5 const int DEFAULT\_CONE\_MIN\_INLIERS = 40 [static]

Definition at line 34 of file ransac\_segmentation.cpp.

Referenced by callRansacConeSegmentation(), and main().

#### 6.7.3.6 const float DEFAULT\_CONE\_OVER\_CYLINDER\_PRIORITY = 0.9f [static]

Definition at line 37 of file ransac\_segmentation.cpp.

Referenced by clustersAcquisition(), and main().

#### 6.7.3.7 const int DEFAULT\_CYLINDER\_MIN\_INLIERS = 40 [static]

Definition at line 33 of file ransac\_segmentation.cpp.

Referenced by callRansacCylinderSegmentation(), and main().

#### 6.7.3.8 const int DEFAULT\_PLANE\_MIN\_INLIERS = 40 [static]

Definition at line 35 of file ransac\_segmentation.cpp.

Referenced by callRansacPlaneSegmentation(), and main().

#### 6.7.3.9 const bool DEFAULT\_SHOW\_PRIMITIVE = false [static]

Definition at line 39 of file ransac\_segmentation.cpp.

Referenced by main().

**6.7.3.10** `const int DEFAULT_SPHERE_MIN_INLIERS = 40` `[static]`

Definition at line 32 of file `ransac_segmentation.cpp`.

Referenced by `callRansacSphereSegmentation()`, and `main()`.

**6.7.3.11** `ros::NodeHandle* nh_ptr = NULL`

Definition at line 24 of file `ransac_segmentation.cpp`.

Referenced by `callRansacConeSegmentation()`, `callRansacCylinderSegmentation()`, `callRansacPlaneSegmentation()`, `callRansacSphereSegmentation()`, and `main()`.

**6.7.3.12** `int planeMinInliers`

Definition at line 48 of file `ransac_segmentation.cpp`.

Referenced by `callRansacPlaneSegmentation()`, and `clustersAcquisition()`.

**6.7.3.13** `Publisher pub`

Definition at line 30 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `main()`.

**6.7.3.14** `bool SHOW_PRIMITIVE` `[static]`

Definition at line 40 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `main()`.

**6.7.3.15** `int sphereMinInliers`

Definition at line 48 of file `ransac_segmentation.cpp`.

Referenced by `callRansacSphereSegmentation()`, and `clustersAcquisition()`.

**6.7.3.16** `const int TXT_CONE_SHAPE_TAG = 3` `[static]`

Definition at line 45 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `returnPrimitiveNameFromTag()`.

**6.7.3.17** `const int TXT_CYLINDER_SHAPE_TAG = 4` `[static]`

Definition at line 46 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `returnPrimitiveNameFromTag()`.

**6.7.3.18** `const int TXT_PLANE_SHAPE_TAG = 1` `[static]`

Definition at line 43 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `returnPrimitiveNameFromTag()`.



6.7.3.19 `const int TXT_SPHERE_SHAPE_TAG = 2` `[static]`

Definition at line 44 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `returnPrimitiveNameFromTag()`.

6.7.3.20 `const int TXT_UNKNOWN_SHAPE_TAG = 0` `[static]`

Definition at line 42 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `returnPrimitiveNameFromTag()`.

6.7.3.21 `boost::shared_ptr< visualization::PCLVisualizer> vis`

Definition at line 26 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, `main()`, and `visSpin()`.

6.7.3.22 `boost::mutex vis_mutex`

Definition at line 28 of file `ransac_segmentation.cpp`.

Referenced by `clustersAcquisition()`, and `visSpin()`.

6.7.3.23 `boost::thread vis_thread`

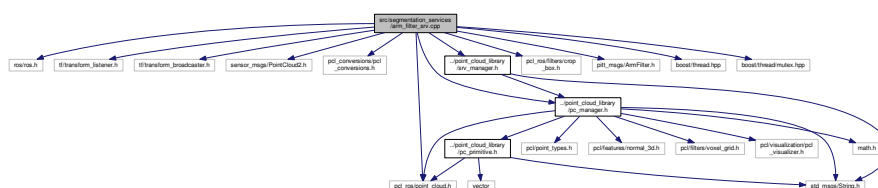
Definition at line 27 of file `ransac_segmentation.cpp`.

Referenced by `main()`.

## 6.8 src/segmentation\_services/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 "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
#include <pitt_msgs/ArmFilter.h>
#include <boost/thread.hpp>
#include <boost/thread/mutex.hpp>
```

Include dependency graph for `arm_filter_srv.cpp`:



## Functions

- [PCLCloudPtr inputCloud](#) (new [PCLCloud](#))
- [PCLCloudPtr outputCloud1](#) (new [PCLCloud](#))
- [PCLCloudPtr outputCloud2](#) (new [PCLCloud](#))
- [PCLCloudPtr outputCloud3](#) (new [PCLCloud](#))
- [PCLCloudPtr outputCloud4](#) (new [PCLCloud](#))
- void [visSpin](#) ()
- [PCLCloudPtr armFiltering](#) ([PCLCloudPtr](#) original, [Vector4f](#) minValues, [Vector4f](#) maxValues, [StampedTransform](#) frame)
- [Vector4f generateBoxVector](#) ([vector< float >](#) vec)
- bool [filter](#) ([ArmFilterRequest](#) &input, [ArmFilterResponse](#) &output)
- int [main](#) (int argc, char \*\*argv)

## Variables

- const float [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MIN\\_FOREARM\\_BOX](#) [] = { -0.040f, -0.120f, -0.190f }
- const float [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MAX\\_FOREARM\\_BOX](#) [] = { 0.340f, 0.120f, 0.105f }
- const float [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MIN\\_ELBOW\\_BOX](#) [] = { -0.090f, -0.135f, -0.160f }
- const float [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MAX\\_ELBOW\\_BOX](#) [] = { 0.440f, 0.135f, 0.110f }
- const [Duration](#) [WAIT\\_FOR\\_TF\\_TIME\\_OUT](#) = [Duration](#)( [srvm::DEFAULT\\_TF\\_WAIT\\_SECONDS](#) )
- [StampedTransform](#) [leftForearmCameraTansf](#)
- [StampedTransform](#) [rightForearmCameraTansf](#)
- [StampedTransform](#) [leftElbowCameraTansf](#)
- [StampedTransform](#) [rightElbowCameraTansf](#)
- bool [tfError](#) = false
- bool [showClouds](#)
- boost::shared\_ptr  
    < [visualization::PCLVisualizer](#) > [vis](#)
- boost::thread [vis\\_thread](#)
- boost::mutex [vis\\_mutex](#)
- string [log\\_str\\_depth](#) = "Loading..."

### 6.8.1 Function Documentation

#### 6.8.1.1 [PCLCloudPtr armFiltering](#) ( [PCLCloudPtr](#) original, [Vector4f](#) minValues, [Vector4f](#) maxValues, [StampedTransform](#) frame )

Definition at line 66 of file [arm\\_filter\\_srv.cpp](#).

Referenced by [filter\(\)](#).

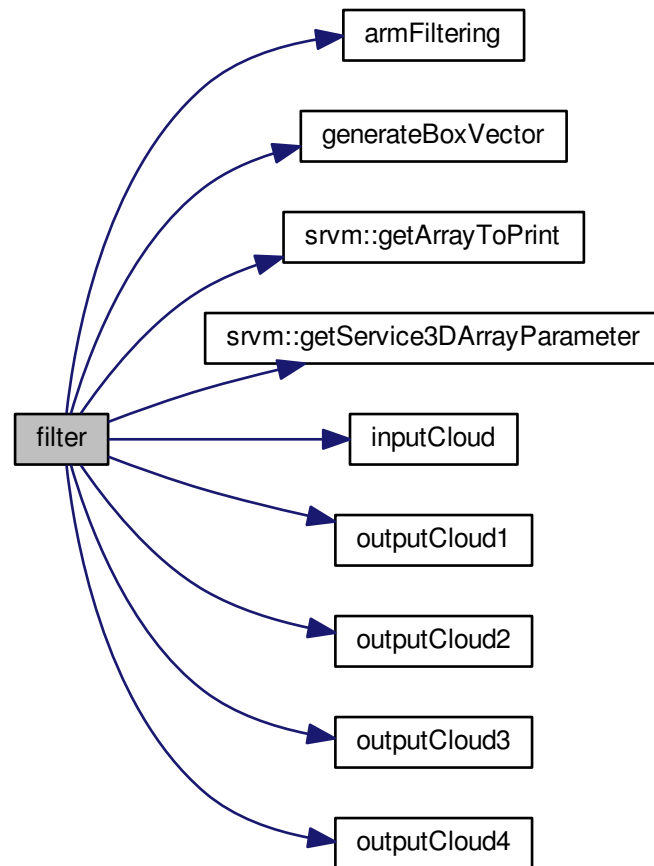
#### 6.8.1.2 bool [filter](#) ( [ArmFilterRequest](#) & input, [ArmFilterResponse](#) & output )

Definition at line 112 of file [arm\\_filter\\_srv.cpp](#).

References [armFiltering\(\)](#), [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MAX\\_ELBOW\\_BOX](#), [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MAX\\_FOREARM\\_BOX](#), [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MIN\\_ELBOW\\_BOX](#), [DEFAULT\\_PARAM\\_ARM\\_SRV\\_MIN\\_FOREARM\\_BOX](#), [generateBoxVector\(\)](#), [srvm::getArrayToPrint\(\)](#), [srvm::getService3DArrayParameter\(\)](#), [inputCloud\(\)](#), [leftElbowCameraTansf](#), [leftForearmCameraTansf](#), [log\\_str\\_depth](#), [outputCloud1\(\)](#), [outputCloud2\(\)](#), [outputCloud3\(\)](#), [outputCloud4\(\)](#), [rightElbowCameraTansf](#), [rightForearmCameraTansf](#), [showClouds](#), [tfError](#), [vis](#), and [vis\\_mutex](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



#### 6.8.1.3 Vector4f generateBoxVector ( vector< float > vec )

Definition at line 105 of file arm\_filter\_srv.cpp.

Referenced by filter().

#### 6.8.1.4 PCLCloudPtr inputCloud ( new PCLCloud )

Referenced by filter().

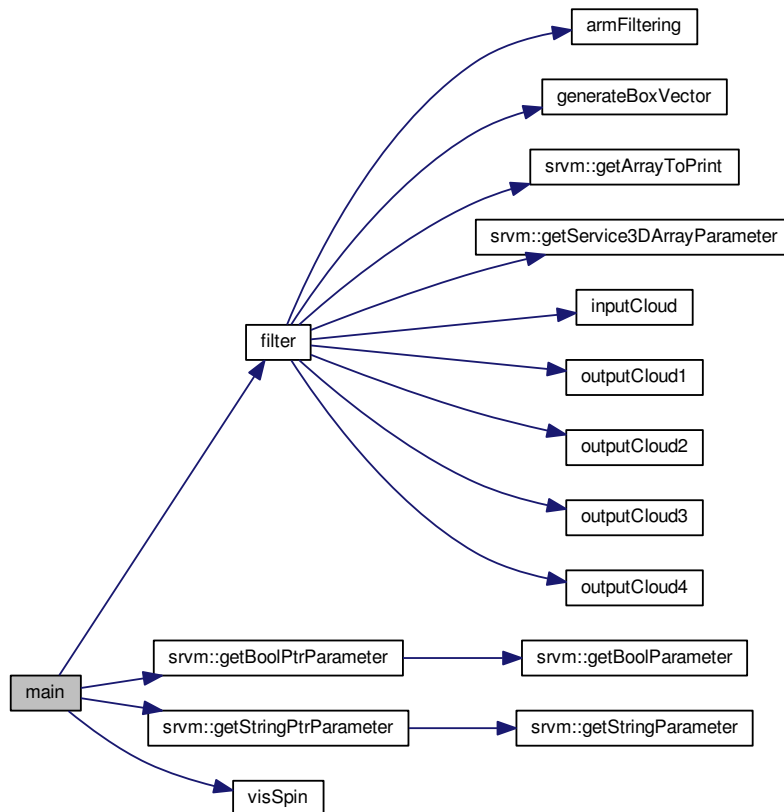
#### 6.8.1.5 int main ( int argc, char \*\* argv )

Definition at line 183 of file arm\_filter\_srv.cpp.

References srvm::DEFAULT\_PARAM\_ARM\_SRV\_CAMERA\_FRAME, srvm::DEFAULT\_PARAM\_ARM\_SRV\_LEFT\_ELBOW\_FRAME, srvm::DEFAULT\_PARAM\_ARM\_SRV\_LEFT\_FOREARM\_FRAME, srvm::DEFAULT\_PARAM\_ARM\_SRV\_RIGHT\_ELBOW\_FRAME, srvm::DEFAULT\_PARAM\_ARM\_SRV\_RIGHT\_FOREARM\_FRAME,

srvm::DEFAULT\_PARAM\_ARM\_SRV\_SHOW\_CLOUDS, filter(), srvm::getBoolPtrParameter(), srvm::getStringPtrParameter(), leftElbowCameraTansf, leftForearmCameraTansf, log\_str\_depth, rightElbowCameraTansf, rightForearmCameraTansf, showClouds, srvm::SRV\_NAME\_ARM\_FILTER, tfError, vis, vis\_thread, visSpin(), and WAIT\_FOR\_TF\_TIME\_OUT.

Here is the call graph for this function:



#### 6.8.1.6 PCLCloudPtr outputCloud1 ( new PCLCloud )

Referenced by filter().

#### 6.8.1.7 PCLCloudPtr outputCloud2 ( new PCLCloud )

Referenced by filter().

#### 6.8.1.8 PCLCloudPtr outputCloud3 ( new PCLCloud )

Referenced by filter().

#### 6.8.1.9 PCLCloudPtr outputCloud4 ( new PCLCloud )

Referenced by filter().

#### 6.8.1.10 void visSpin ( )

Definition at line 57 of file arm\_filter\_srv.cpp.

References vis, and vis\_mutex.

Referenced by main().

### 6.8.2 Variable Documentation

#### 6.8.2.1 const float DEFAULT\_PARAM\_ARM\_SRV\_MAX\_ELBOW\_BOX[] = { 0.440f, 0.135f, 0.110f}

Definition at line 35 of file arm\_filter\_srv.cpp.

Referenced by filter().

#### 6.8.2.2 const float DEFAULT\_PARAM\_ARM\_SRV\_MAX\_FOREARM\_BOX[] = { 0.340f, 0.120f, 0.105f}

Definition at line 33 of file arm\_filter\_srv.cpp.

Referenced by filter().

#### 6.8.2.3 const float DEFAULT\_PARAM\_ARM\_SRV\_MIN\_ELBOW\_BOX[] = { -0.090f, -0.135f, -0.160f}

Definition at line 34 of file arm\_filter\_srv.cpp.

Referenced by filter().

#### 6.8.2.4 const float DEFAULT\_PARAM\_ARM\_SRV\_MIN\_FOREARM\_BOX[] = { -0.040f, -0.120f, -0.190f}

Definition at line 32 of file arm\_filter\_srv.cpp.

Referenced by filter().

#### 6.8.2.5 StampedTransform leftElbowCameraTansf

Definition at line 40 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.6 StampedTransform leftForearmCameraTansf

Definition at line 40 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.7 string log\_str\_depth = "Loading..."

Definition at line 49 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.8 StampedTransform rightElbowCameraTansf

Definition at line 40 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.9 StampedTransform rightForearmCameraTansf

Definition at line 40 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.10 bool showClouds

Definition at line 44 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.11 bool tfError = false

Definition at line 43 of file arm\_filter\_srv.cpp.

Referenced by filter(), and main().

#### 6.8.2.12 boost::shared\_ptr< visualization::PCLVisualizer> vis

Definition at line 46 of file arm\_filter\_srv.cpp.

Referenced by filter(), main(), and visSpin().

#### 6.8.2.13 boost::mutex vis\_mutex

Definition at line 48 of file arm\_filter\_srv.cpp.

Referenced by filter(), and visSpin().

#### 6.8.2.14 boost::thread vis\_thread

Definition at line 47 of file arm\_filter\_srv.cpp.

Referenced by main().

#### 6.8.2.15 const Duration WAIT\_FOR\_TF\_TIME\_OUT = Duration( srvm::DEFAULT\_TF\_WAIT\_SECONDS)

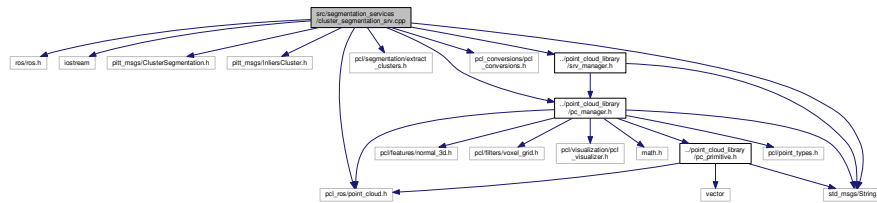
Definition at line 37 of file arm\_filter\_srv.cpp.

Referenced by main().

## 6.9 src/segmentation\_services/cluster\_segmentation\_srv.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 "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
```

Include dependency graph for cluster\_segmentation\_srv.cpp:



## Functions

- bool [clusterize](#) (ClusterSegmentation::Request &req, ClusterSegmentation::Response &res)
- int [main](#) (int argc, char \*\*argv)

## Variables

- ros::NodeHandle \* [nh\\_ptr](#) = NULL
- const double [CLUSTER\\_TOLERANCE\\_DEFAULT](#) = 0.03
- const double [CLUSTER\\_MIN\\_RATE\\_DEFAULT](#) = 0.01
- const double [CLUSTER\\_MAX\\_RATE\\_DEFAULT](#) = 0.99
- const int [CLUSTER\\_MIN\\_INPUT\\_SIZE\\_DEFAULT](#) = 30

### 6.9.1 Function Documentation

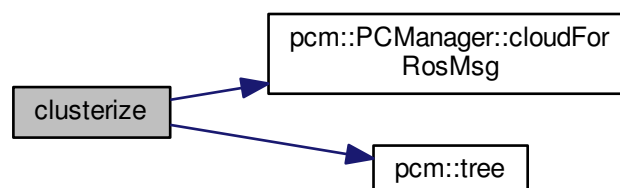
#### 6.9.1.1 bool clusterize ( ClusterSegmentation::Request & req, ClusterSegmentation::Response & res )

Definition at line 38 of file cluster\_segmentation\_srv.cpp.

References `pcm::PCManager::cloudForRosMsg()`, `CLUSTER_MAX_RATE_DEFAULT`, `CLUSTER_MIN_INPUT_SIZE_DEFAULT`, `CLUSTER_MIN_RATE_DEFAULT`, `CLUSTER_TOLERANCE_DEFAULT`, `nh_ptr`, `srvm::PARAM_NAME_CLUSTER_MAX_RATE`, `srvm::PARAM_NAME_CLUSTER_MIN_RATE`, `srvm::PARAM_NAME_CLUSTER_TOLERANCE`, and `pcm::tree()`.

Referenced by `main()`.

Here is the call graph for this function:

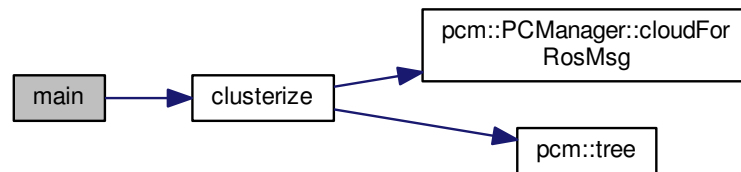


### 6.9.1.2 `int main ( int argc, char ** argv )`

Definition at line 114 of file `cluster_segmentation_srv.cpp`.

References `clusterize()`, `nh_ptr`, and `srvm::SRV_NAME_CUSTER_FILTER`.

Here is the call graph for this function:



## 6.9.2 Variable Documentation

### 6.9.2.1 `const double CLUSTER_MAX_RATE_DEFAULT = 0.99`

Definition at line 34 of file `cluster_segmentation_srv.cpp`.

Referenced by `clusterize()`.

### 6.9.2.2 `const int CLUSTER_MIN_INPUT_SIZE_DEFAULT = 30`

Definition at line 35 of file `cluster_segmentation_srv.cpp`.

Referenced by `clusterize()`.

### 6.9.2.3 `const double CLUSTER_MIN_RATE_DEFAULT = 0.01`

Definition at line 33 of file `cluster_segmentation_srv.cpp`.

Referenced by `clusterize()`.

### 6.9.2.4 `const double CLUSTER_TOLERANCE_DEFAULT = 0.03`

Definition at line 32 of file `cluster_segmentation_srv.cpp`.

Referenced by `clusterize()`.

### 6.9.2.5 `ros::NodeHandle* nh_ptr = NULL`

Definition at line 29 of file `cluster_segmentation_srv.cpp`.

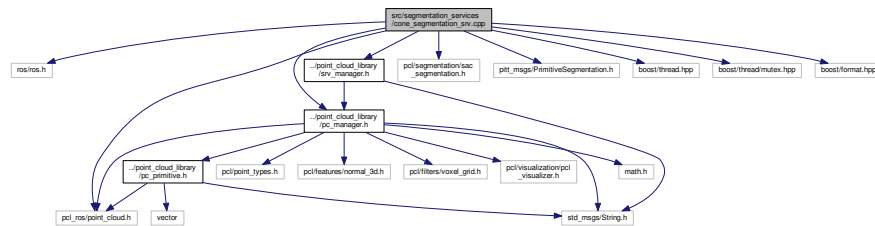
Referenced by `clusterize()`, and `main()`.

## 6.10 `src/segmentation_services/cone_segmentation_srv.cpp` File Reference

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



Include dependency graph for cone\_segmentation\_srv.cpp:



- struct **vector3d**

- void **visSpin** ()
- **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)

- `ros::NodeHandle * nh_ptr = NULL`
- `static const double CONE_NORMAL_DISTANCE_WEIGHTH = 0.0006`
- `static const double CONE_DISTANCE_TH = 0.0055`
- `static const double CONE_MIN_RADIUS_LIMIT = 0.001`
- `static const double CONE_MAX_RADIUS_LIMIT = 0.500`
- `static const int CONE_MAX_ITERATION_LIMIT = 1000`
- `static const double CONE_EPS_ANGLE_TH = 0.4`
- `static const double CONE_MIN_OPENING_ANGLE_DEGREE = 10.0`
- `static const double CONE_MAX_OPENING_ANGLE_DEGREE = 170.0`
- `const bool VISUALIZE_RESULT = false`
- `boost::shared_ptr  
< visualization::PCLVisualizer > vis`
- `boost::thread vis_thread`
- `boost::mutex vis_mutex`

## 6.10.1 Function Documentation

### 6.10.1.1 `vector3d getNormalizeAxesDirectionVector ( ModelCoefficients::Ptr coefficients )`

Definition at line 54 of file cone\_segmentation\_srv.cpp.

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

Referenced by `ransacConeDetaction()`.

### 6.10.1.2 `vector3d getPointOnAxes ( ModelCoefficients::Ptr coefficients, vector3d direction, float t )`

Definition at line 65 of file cone\_segmentation\_srv.cpp.

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

Referenced by `ransacConeDetaction()`.

### 6.10.1.3 `vector3d getVectorBetweenPoints ( vector3d p1, vector3d p2 )`

Definition at line 74 of file cone\_segmentation\_srv.cpp.

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

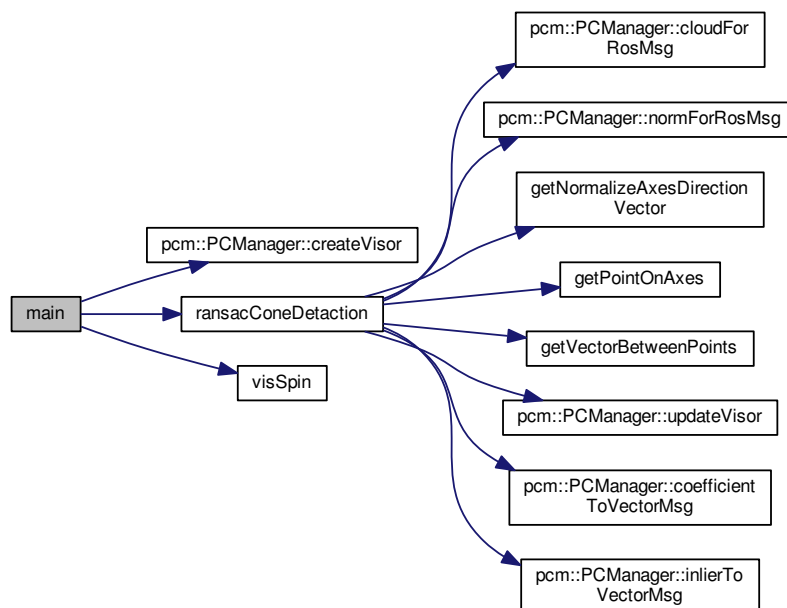
Referenced by `ransacConeDetaction()`.

### 6.10.1.4 `int main ( int argc, char ** argv )`

Definition at line 220 of file cone\_segmentation\_srv.cpp.

References `pcm::PCManager::createVisor()`, `nh_ptr`, `ransacConeDetaction()`, `srvm::SRV_NAME_RANSAC_CONE_FILTER`, `vis`, `vis_thread`, `visSpin()`, and `VISUALIZE_RESULT`.

Here is the call graph for this function:



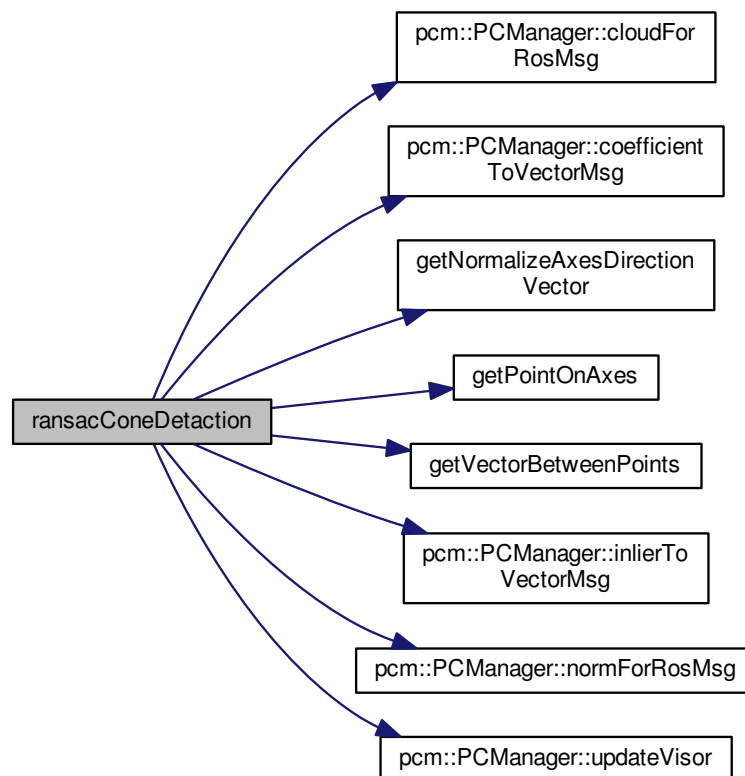
## 6.10.1.5 bool ransacConeDetaction ( PrimitiveSegmentation::Request &amp; req, PrimitiveSegmentation::Response &amp; res )

Definition at line 83 of file cone\_segmentation\_srv.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), CONE\_DISTANCE\_TH, CONE\_EPS\_ANGLE\_TH, CONE\_MAX\_ITERATION\_LIMIT, CONE\_MAX\_OPENING\_ANGLE\_DEGREE, CONE\_MAX\_RADIUS\_LIMIT, CONE\_MIN\_OPENING\_ANGLE\_DEGREE, CONE\_MIN\_RADIUS\_LIMIT, CONE\_NORMAL\_DISTANCE\_WEIGHTH, getNormalizeAxesDirectionVector(), getPointOnAxes(), getVectorBetweenPoints(), pcm::PCManager::inlierToVectorMsg(), nh\_ptr, pcm::PCManager::normForRosMsg(), srv::PARAM\_NAME\_CONE\_DISTANCE\_TH, srv::PARAM\_NAME\_CONE\_EPS\_ANGLE\_TH, srv::PARAM\_NAME\_CONE\_MAX\_ITERATION\_LIMIT, srv::PARAM\_NAME\_CONE\_MAX\_OPENING\_ANGLE\_DEGREE, srv::PARAM\_NAME\_CONE\_MAX\_RADIUS\_LIMIT, srv::PARAM\_NAME\_CONE\_MIN\_OPENING\_ANGLE\_DEGREE, srv::PARAM\_NAME\_CONE\_MIN\_RADIUS\_LIMIT, srv::PARAM\_NAME\_CONE\_NORMAL\_DISTANCE\_WEIGHT, seg, pcm::PCManager::updateVisor(), vis, vis\_mutex, VISUALIZE\_RESULT, vector3d::x, vector3d::y, and vector3d::z.

Referenced by main().

Here is the call graph for this function:



## 6.10.1.6 void visSpin ( )

Definition at line 46 of file cone\_segmentation\_srv.cpp.

References vis, and vis\_mutex.

Referenced by main().

## 6.10.2 Variable Documentation

### 6.10.2.1 `const double CONE_DISTANCE_TH = 0.0055` [static]

Definition at line 25 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.2 `const double CONE_EPS_ANGLE_TH = 0.4` [static]

Definition at line 29 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.3 `const int CONE_MAX_ITERATION_LIMIT = 1000` [static]

Definition at line 28 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.4 `const double CONE_MAX_OPENING_ANGLE_DEGREE = 170.0` [static]

Definition at line 31 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.5 `const double CONE_MAX_RADIUS_LIMIT = 0.500` [static]

Definition at line 27 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.6 `const double CONE_MIN_OPENING_ANGLE_DEGREE = 10.0` [static]

Definition at line 30 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.7 `const double CONE_MIN_RADIUS_LIMIT = 0.001` [static]

Definition at line 26 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.8 `const double CONE_NORMAL_DISTANCE_WEIGTH = 0.0006` [static]

Definition at line 24 of file cone\_segmentation\_srv.cpp.

Referenced by `ransacConeDetaction()`.

### 6.10.2.9 `ros::NodeHandle* nh_ptr = NULL`

Definition at line 21 of file cone\_segmentation\_srv.cpp.

Referenced by `main()`, and `ransacConeDetaction()`.

6.10.2.10 `boost::shared_ptr< visualization::PCLVisualizer > vis`

Definition at line 42 of file `cone_segmentation_srv.cpp`.

Referenced by `main()`, `ransacConeDetaction()`, and `visSpin()`.

6.10.2.11 `boost::mutex vis_mutex`

Definition at line 44 of file `cone_segmentation_srv.cpp`.

Referenced by `ransacConeDetaction()`, and `visSpin()`.

6.10.2.12 `boost::thread vis_thread`

Definition at line 43 of file `cone_segmentation_srv.cpp`.

Referenced by `main()`.

6.10.2.13 `const bool VISUALIZE_RESULT = false`

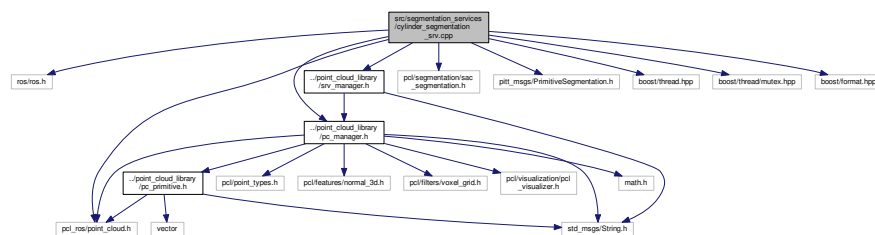
Definition at line 41 of file `cone_segmentation_srv.cpp`.

Referenced by `main()`, and `ransacConeDetaction()`.

## 6.11 src/segmentation\_services/cylinder\_segmentation\_srv.cpp File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "pitt_msgs/PrimitiveSegmentation.h"
#include "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
#include <boost/thread.hpp>
#include <boost/thread/mutex.hpp>
#include <boost/format.hpp>
```

Include dependency graph for `cylinder_segmentation_srv.cpp`:



## Classes

- struct `vector3d`

## Functions

- void `visSpin ()`

- [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

- ros::NodeHandle \* [nh\\_ptr](#) = NULL
- static const double [CYLINDER\\_NORMAL\\_DISTANCE\\_WEIGTH](#) = 0.001
- static const double [CYLINDER\\_DISTANCE\\_TH](#) = 0.008
- static const double [CYLINDER\\_MIN\\_RADIUS\\_LIMIT](#) = 0.005
- static const double [CYLINDER\\_MAX\\_RADIUS\\_LIMIT](#) = 0.500
- static const int [CYLINDER\\_MAX\\_ITERATION\\_LIMIT](#) = 1000
- static const double [CYLINDER\\_EPS\\_ANGLE\\_TH](#) = 0.0001
- static const double [CYLINDER\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = 50.0
- static const double [CYLINDER\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = 180.0
- const bool [VISUALIZE\\_RESULT](#) = false
- boost::shared\_ptr  
    < [visualization::PCLVisualizer](#) > [vis](#)
- boost::thread [vis\\_thread](#)
- boost::mutex [vis\\_mutex](#)

## 6.11.1 Function Documentation

### 6.11.1.1 [vector3d getNormalizeAxesDirectionVector](#) ( [ModelCoefficients::Ptr](#) *coefficients* )

Definition at line 53 of file [cylinder\\_segmentation\\_srv.cpp](#).

References [vector3d::x](#), [vector3d::y](#), and [vector3d::z](#).

Referenced by [ransacCylinderDetaction\(\)](#).

### 6.11.1.2 [vector3d getPointOnAxes](#) ( [ModelCoefficients::Ptr](#) *coefficients*, [vector3d](#) *direction*, float *t* )

Definition at line 64 of file [cylinder\\_segmentation\\_srv.cpp](#).

References [vector3d::x](#), [vector3d::y](#), and [vector3d::z](#).

Referenced by [ransacCylinderDetaction\(\)](#).

### 6.11.1.3 [vector3d getVectorBetweenPoints](#) ( [vector3d](#) *p1*, [vector3d](#) *p2* )

Definition at line 73 of file [cylinder\\_segmentation\\_srv.cpp](#).

References [vector3d::x](#), [vector3d::y](#), and [vector3d::z](#).

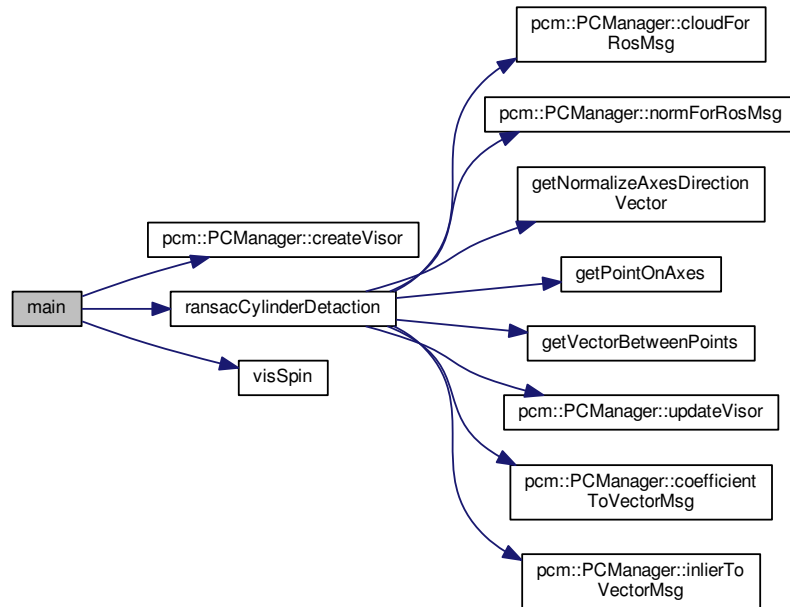
Referenced by [ransacCylinderDetaction\(\)](#).

### 6.11.1.4 [int main](#) ( [int](#) *argc*, [char](#) \*\* *argv* )

Definition at line 220 of file [cylinder\\_segmentation\\_srv.cpp](#).

References [pcm::PCManager::createVisor\(\)](#), [nh\\_ptr](#), [ransacCylinderDetaction\(\)](#), [srvm::SRV\\_NAME\\_RANSAC\\_CYLINDER\\_FILTER](#), [vis](#), [vis\\_thread](#), [visSpin\(\)](#), and [VISUALIZE\\_RESULT](#).

Here is the call graph for this function:



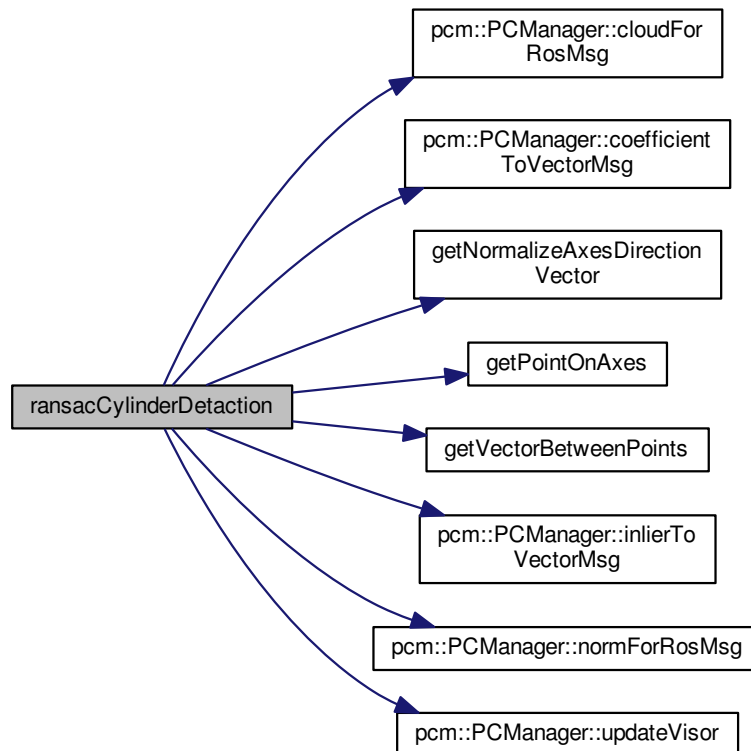
#### 6.11.1.5 bool ransacCylinderDetaction ( PrimitiveSegmentation::Request & req, PrimitiveSegmentation::Response & res )

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

References `pcm::PCManager::cloudForRosMsg()`, `pcm::PCManager::coefficientToVectorMsg()`, `CYLINDER_DISTANCE_TH`, `CYLINDER_EPS_ANGLE_TH`, `CYLINDER_MAX_ITERATION_LIMIT`, `CYLINDER_MAX_OPENING_ANGLE_DEGREE`, `CYLINDER_MAX_RADIUS_LIMIT`, `CYLINDER_MIN_OPENING_ANGLE_DEGREE`, `CYLINDER_MIN_RADIUS_LIMIT`, `CYLINDER_NORMAL_DISTANCE_WEIGHT`, `getNormalizeAxesDirectionVector()`, `getPointOnAxes()`, `getVectorBetweenPoints()`, `pcm::PCManager::inlierToVectorMsg()`, `nh_ptr`, `pcm::PCManager::normForRosMsg()`, `srvm::PARAM_NAME_CYLINDER_DISTANCE_TH`, `srvm::PARAM_NAME_CYLINDER_EPS_ANGLE_TH`, `srvm::PARAM_NAME_CYLINDER_MAX_ITERATION_LIMIT`, `srvm::PARAM_NAME_CYLINDER_MAX_OPENING_ANGLE_DEGREE`, `srvm::PARAM_NAME_CYLINDER_MAX_RADIUS_LIMIT`, `srvm::PARAM_NAME_CYLINDER_MIN_OPENING_ANGLE_DEGREE`, `srvm::PARAM_NAME_CYLINDER_MIN_RADIUS_LIMIT`, `srvm::PARAM_NAME_CYLINDER_NORMAL_DISTANCE_WEIGHT`, `seg`, `pcm::PCManager::updateVisor()`, `vis`, `vis_mutex`, `VISUALIZE_RESULT`, `vector3d::x`, `vector3d::y`, and `vector3d::z`.

Referenced by `main()`.

Here is the call graph for this function:



#### 6.11.1.6 void visSpin ( )

Definition at line 45 of file `cylinder_segmentation_srv.cpp`.

References `vis`, and `vis_mutex`.

Referenced by `main()`.

### 6.11.2 Variable Documentation

#### 6.11.2.1 const double CYLINDER\_DISTANCE\_TH = 0.008 [static]

Definition at line 24 of file `cylinder_segmentation_srv.cpp`.

Referenced by `ransacCylinderDetaction()`.

#### 6.11.2.2 const double CYLINDER\_EPS\_ANGLE\_TH = 0.0001 [static]

Definition at line 28 of file `cylinder_segmentation_srv.cpp`.

Referenced by `ransacCylinderDetaction()`.



**6.11.2.3** `const int CYLINDER_MAX_ITERATION_LIMIT = 1000` `[static]`

Definition at line 27 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.4** `const double CYLINDER_MAX_OPENING_ANGLE_DEGREE = 180.0` `[static]`

Definition at line 30 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.5** `const double CYLINDER_MAX_RADIUS_LIMIT = 0.500` `[static]`

Definition at line 26 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.6** `const double CYLINDER_MIN_OPENING_ANGLE_DEGREE = 50.0` `[static]`

Definition at line 29 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.7** `const double CYLINDER_MIN_RADIUS_LIMIT = 0.005` `[static]`

Definition at line 25 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.8** `const double CYLINDER_NORMAL_DISTANCE_WEIGHT = 0.001` `[static]`

Definition at line 23 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction().

**6.11.2.9** `ros::NodeHandle* nh_ptr = NULL`

Definition at line 20 of file cylinder\_segmentation\_srv.cpp.

Referenced by main(), and ransacCylinderDetaction().

**6.11.2.10** `boost::shared_ptr< visualization::PCLVisualizer> vis`

Definition at line 41 of file cylinder\_segmentation\_srv.cpp.

Referenced by main(), ransacCylinderDetaction(), and visSpin().

**6.11.2.11** `boost::mutex vis_mutex`

Definition at line 43 of file cylinder\_segmentation\_srv.cpp.

Referenced by ransacCylinderDetaction(), and visSpin().

#### 6.11.2.12 boost::thread vis\_thread

Definition at line 42 of file cylinder\_segmentation\_srv.cpp.

Referenced by main().

#### 6.11.2.13 const bool VISUALIZE\_RESULT = false

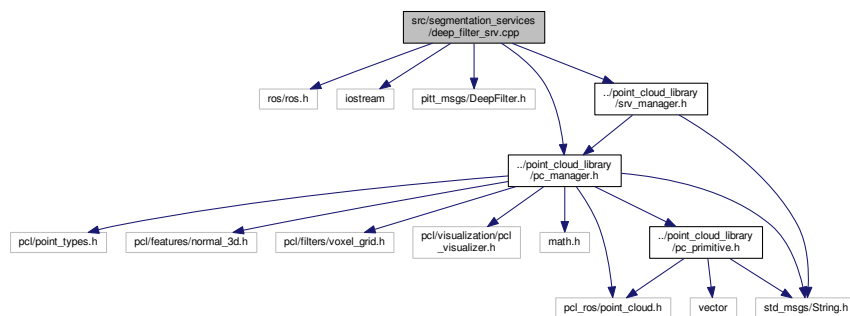
Definition at line 40 of file cylinder\_segmentation\_srv.cpp.

Referenced by main(), and ransacCylinderDetaction().

## 6.12 src/segmentation\_services/deep\_filter\_srv.cpp File Reference

```
#include "ros/ros.h"
#include <iostream>
#include "pitt_msgs/DeepFilter.h"
#include "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
```

Include dependency graph for deep\_filter\_srv.cpp:



## Functions

- bool [deepFiltering](#) (DeepFilter::Request &req, DeepFilter::Response &res)
- int [main](#) (int argc, char \*\*argv)

## Variables

- const float [DEFAULT\\_PARAM\\_DEEP\\_SRV\\_Z\\_THRESHOLD](#) = 3.000f

### 6.12.1 Function Documentation

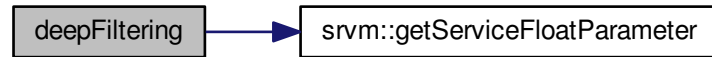
#### 6.12.1.1 bool deepFiltering ( DeepFilter::Request & req, DeepFilter::Response & res )

Definition at line 27 of file deep\_filter\_srv.cpp.

References [DEFAULT\\_PARAM\\_DEEP\\_SRV\\_Z\\_THRESHOLD](#), and [srvm::getServiceFloatParameter\(\)](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



#### 6.12.1.2 int main ( int argc, char \*\* argv )

Definition at line 59 of file `deep_filter_srv.cpp`.

References `deepFiltering()`, and `srvm::SRV_NAME_DEEP_FILTER`.

Here is the call graph for this function:



### 6.12.2 Variable Documentation

#### 6.12.2.1 const float DEFAULT\_PARAM\_DEEP\_SRV\_Z\_THRESHOLD = 3.000f

Definition at line 20 of file `deep_filter_srv.cpp`.

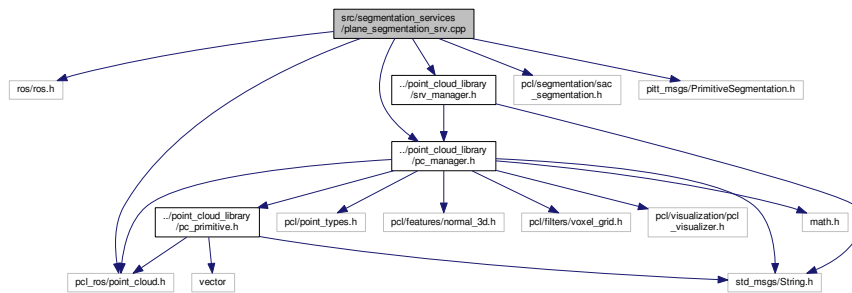
Referenced by `deepFiltering()`.

## 6.13 src/segmentation\_services/plane\_segmentation\_srv.cpp File Reference

```

#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "pitt_msgs/PrimitiveSegmentation.h"
#include "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
  
```

Include dependency graph for plane\_segmentation\_srv.cpp:



## Functions

- bool [ransacPlaneDetaction](#) (PrimitiveSegmentation::Request &req, PrimitiveSegmentation::Response &res)
- int [main](#) (int argc, char \*\*argv)

## Variables

- ros::NodeHandle \* [nh\\_ptr](#) = NULL
- static const double [PLANE\\_NORMAL\\_DISTANCE\\_WEIGTH](#) = 0.001
- static const double [PLANE\\_DISTANCE\\_TH](#) = 0.007
- static const int [PLANE\\_MAX\\_ITERATION\\_LIMIT](#) = 1000
- static const double [PLANE\\_EPS\\_ANGLE\\_TH](#) = 0.0
- static const double [PLANE\\_MIN\\_OPENING\\_ANGLE\\_DEGREE](#) = 0.0
- static const double [PLANE\\_MAX\\_OPENING\\_ANGLE\\_DEGREE](#) = 10.0

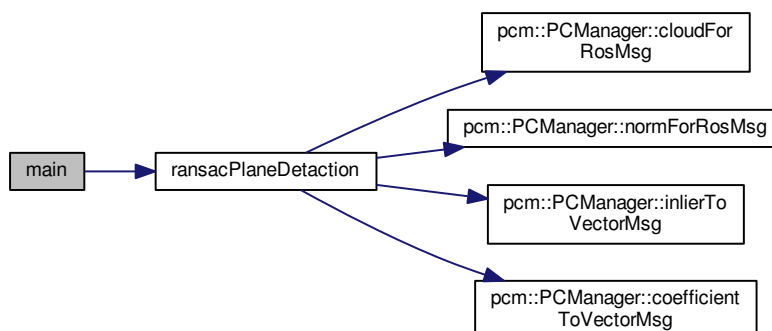
### 6.13.1 Function Documentation

#### 6.13.1.1 int main ( int argc, char \*\* argv )

Definition at line 78 of file plane\_segmentation\_srv.cpp.

References [nh\\_ptr](#), [ransacPlaneDetaction\(\)](#), and [srvm::SRV\\_NAME\\_RANSAC\\_PLANE\\_FILTER](#).

Here is the call graph for this function:



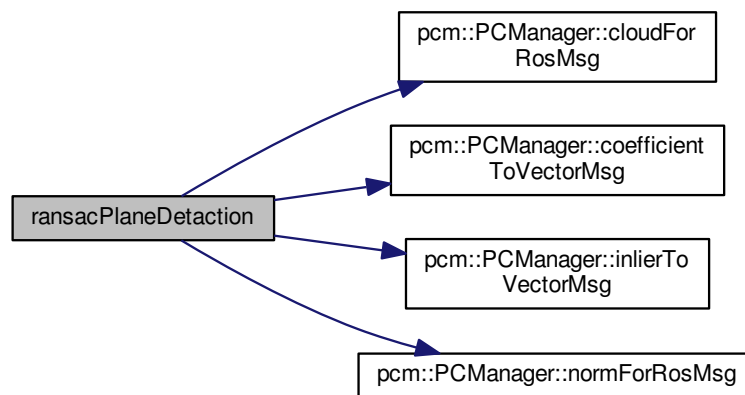
### 6.13.1.2 bool ransacPlaneDetaction ( PrimitiveSegmentation::Request & *req*, PrimitiveSegmentation::Response & *res* )

Definition at line 27 of file plane\_segmentation\_srv.cpp.

References pcm::PCManager::cloudForRosMsg(), pcm::PCManager::coefficientToVectorMsg(), pcm::PCManager::inlierToVectorMsg(), nh\_ptr, pcm::PCManager::normForRosMsg(), srvm::PARAM\_NAME\_PLANE\_DISTANCE\_TH, srvm::PARAM\_NAME\_PLANE\_EPS\_ANGLE\_TH, srvm::PARAM\_NAME\_PLANE\_MAX\_ITERATION\_LIMIT, srvm::PARAM\_NAME\_PLANE\_MAX\_OPENING\_ANGLE\_DEGREE, srvm::PARAM\_NAME\_PLANE\_MIN\_OPENING\_ANGLE\_DEGREE, srvm::PARAM\_NAME\_PLANE\_NORMAL\_DISTANCE\_WEIGHT, PLANE\_DISTANCE\_TH, PLANE\_EPS\_ANGLE\_TH, PLANE\_MAX\_ITERATION\_LIMIT, PLANE\_MAX\_OPENING\_ANGLE\_DEGREE, PLANE\_MIN\_OPENING\_ANGLE\_DEGREE, PLANE\_NORMAL\_DISTANCE\_WEIGHT, and seg.

Referenced by main().

Here is the call graph for this function:



## 6.13.2 Variable Documentation

### 6.13.2.1 ros::NodeHandle\* nh\_ptr = NULL

Definition at line 16 of file plane\_segmentation\_srv.cpp.

Referenced by main(), and ransacPlaneDetaction().

### 6.13.2.2 const double PLANE\_DISTANCE\_TH = 0.007 [static]

Definition at line 20 of file plane\_segmentation\_srv.cpp.

Referenced by ransacPlaneDetaction().

### 6.13.2.3 const double PLANE\_EPS\_ANGLE\_TH = 0.0 [static]

Definition at line 22 of file plane\_segmentation\_srv.cpp.

Referenced by ransacPlaneDetaction().

#### 6.13.2.4 `const int PLANE_MAX_ITERATION_LIMIT = 1000` [static]

Definition at line 21 of file `plane_segmentation_srv.cpp`.

Referenced by `ransacPlaneDetaction()`.

#### 6.13.2.5 `const double PLANE_MAX_OPENING_ANGLE_DEGREE = 10.0` [static]

Definition at line 24 of file `plane_segmentation_srv.cpp`.

Referenced by `ransacPlaneDetaction()`.

#### 6.13.2.6 `const double PLANE_MIN_OPENING_ANGLE_DEGREE = 0.0` [static]

Definition at line 23 of file `plane_segmentation_srv.cpp`.

Referenced by `ransacPlaneDetaction()`.

#### 6.13.2.7 `const double PLANE_NORMAL_DISTANCE_WEIGHT = 0.001` [static]

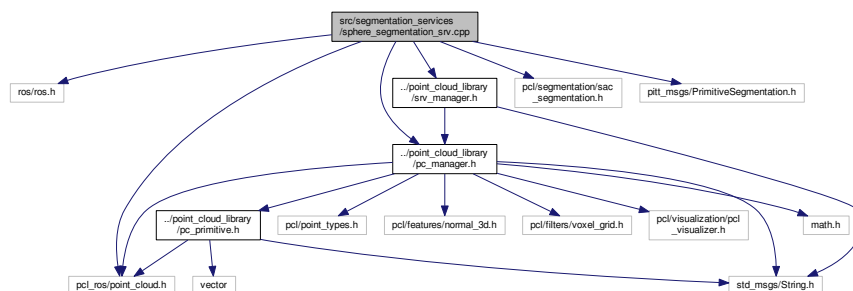
Definition at line 19 of file `plane_segmentation_srv.cpp`.

Referenced by `ransacPlaneDetaction()`.

## 6.14 `src/segmentation_services/sphere_segmentation_srv.cpp` File Reference

```
#include "ros/ros.h"
#include <pcl_ros/point_cloud.h>
#include <pcl/segmentation/sac_segmentation.h>
#include "pitt_msgs/PrimitiveSegmentation.h"
#include "../point_cloud_library/pc_manager.h"
#include "../point_cloud_library/srv_manager.h"
```

Include dependency graph for `sphere_segmentation_srv.cpp`:



## Functions

- bool [ransacSphereDetection](#) (`PrimitiveSegmentation::Request &req`, `PrimitiveSegmentation::Response &res`)
- int [main](#) (`int argc`, `char **argv`)

## Variables

- `ros::NodeHandle * nh_ptr = NULL`
- static const double `SPHERE_NORMAL_DISTANCE_WEIGHT = 0.001`
- static const double `SPHERE_DISTANCE_TH = 0.007`
- static const double `SPHERE_MIN_RADIUS_LIMIT = 0.005`
- static const double `SPHERE_MAX_RADIUS_LIMIT = 0.500`
- static const int `SPHERE_MAX_ITERATION_LIMIT = 1000`
- static const double `SPHERE_EPS_ANGLE_TH = 0.0`
- static const double `SPHERE_MIN_OPENING_ANGLE_DEGREE = 100.0`
- static const double `SPHERE_MAX_OPENING_ANGLE_DEGREE = 180.0`

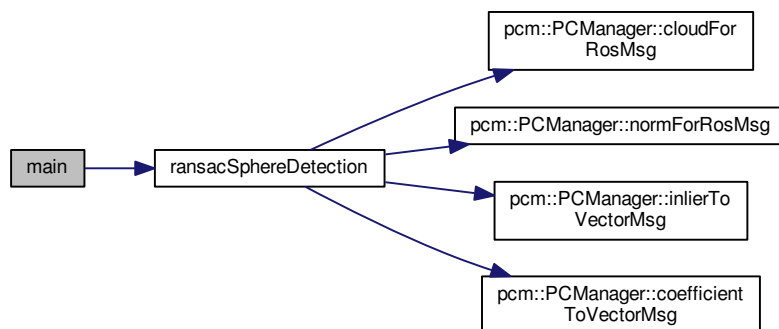
### 6.14.1 Function Documentation

#### 6.14.1.1 `int main ( int argc, char ** argv )`

Definition at line 100 of file `sphere_segmentation_srv.cpp`.

References `nh_ptr`, `ransacSphereDetection()`, and `srvm::SRV_NAME_RANSAC_SPHERE_FILTER`.

Here is the call graph for this function:



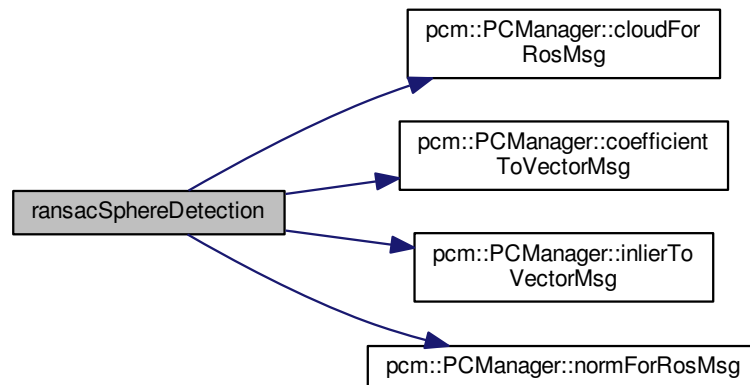
#### 6.14.1.2 `bool ransacSphereDetection ( PrimitiveSegmentation::Request & req, PrimitiveSegmentation::Response & res )`

Definition at line 29 of file `sphere_segmentation_srv.cpp`.

References `pcm::PCManager::cloudForRosMsg()`, `pcm::PCManager::coefficientToVectorMsg()`, `pcm::PCManager::inlierToVectorMsg()`, `nh_ptr`, `pcm::PCManager::normForRosMsg()`, `srvm::PARAM_NAME_SPHERE_DISTANCE_TH`, `srvm::PARAM_NAME_SPHERE_EPS_ANGLE_TH`, `srvm::PARAM_NAME_SPHERE_MAX_ITERATION_LIMIT`, `srvm::PARAM_NAME_SPHERE_MAX_OPENING_ANGLE_DEGREE`, `srvm::PARAM_NAME_SPHERE_MAX_RADIUS_LIMIT`, `srvm::PARAM_NAME_SPHERE_MIN_OPENING_ANGLE_DEGREE`, `srvm::PARAM_NAME_SPHERE_MIN_RADIUS_LIMIT`, `srvm::PARAM_NAME_SPHERE_NORMAL_DISTANCE_WEIGHT`, `seg`, `SPHERE_DISTANCE_TH`, `SPHERE_EPS_ANGLE_TH`, `SPHERE_MAX_ITERATION_LIMIT`, `SPHERE_MAX_OPENING_ANGLE_DEGREE`, `SPHERE_MAX_RADIUS_LIMIT`, `SPHERE_MIN_OPENING_ANGLE_DEGREE`, `SPHERE_MIN_RADIUS_LIMIT`, and `SPHERE_NORMAL_DISTANCE_WEIGHT`.

Referenced by `main()`.

Here is the call graph for this function:



## 6.14.2 Variable Documentation

### 6.14.2.1 `ros::NodeHandle* nh_ptr = NULL`

Definition at line 16 of file `sphere_segmentation_srv.cpp`.

Referenced by `main()`, and `ransacSphereDetection()`.

### 6.14.2.2 `const double SPHERE_DISTANCE_TH = 0.007` `[static]`

Definition at line 20 of file `sphere_segmentation_srv.cpp`.

Referenced by `ransacSphereDetection()`.

### 6.14.2.3 `const double SPHERE_EPS_ANGLE_TH = 0.0` `[static]`

Definition at line 24 of file `sphere_segmentation_srv.cpp`.

Referenced by `ransacSphereDetection()`.

### 6.14.2.4 `const int SPHERE_MAX_ITERATION_LIMIT = 1000` `[static]`

Definition at line 23 of file `sphere_segmentation_srv.cpp`.

Referenced by `ransacSphereDetection()`.

### 6.14.2.5 `const double SPHERE_MAX_OPENING_ANGLE_DEGREE = 180.0` `[static]`

Definition at line 26 of file `sphere_segmentation_srv.cpp`.

Referenced by `ransacSphereDetection()`.

### 6.14.2.6 `const double SPHERE_MAX_RADIUS_LIMIT = 0.500` `[static]`

Definition at line 22 of file `sphere_segmentation_srv.cpp`.

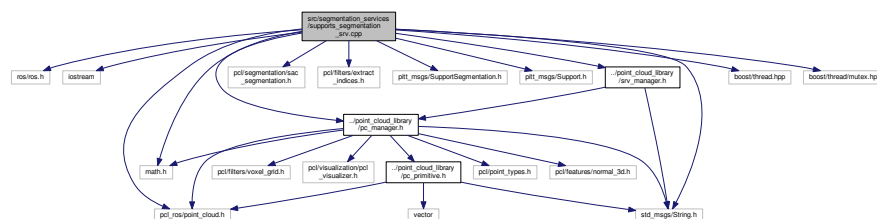


6.14.2.7 `const double SPHERE_MIN_OPENING_ANGLE_DEGREE = 100.0` [static]

Referenced by `ransacSphereDetection()`.

Referenced by `ransacSphereDetection()`.

Referenced by `ransacSphereDetection()`.



- void `initializeInputParameters` (SupportSegmentation::Request &req)
- void `ransacPlaneSegmentator` (PCLCloudPtr inputCloud, PCLNormalPtr normals, PointIndices::Ptr &inlier-Output, ModelCoefficients::Ptr &coefficientOutput)
- ExtractIndices< PointXYZ > `extract` (true)
- void `removePlaneInliner` (PCLCloudPtr inputCloud, PointIndices::Ptr &removeIndex, PCLCloudPtr output)

- bool [valueBelongsToArray](#) (int value, PointIndices::Ptr inliers)
- [PrimitiveldxPtr createNewldxMap](#) ([PrimitiveldxPtr](#) previousInliersMap, PointIndices::Ptr inliers, int level)
- bool [isHorizontalPlane](#) ([PCLNormalPtr](#) normal, ModelCoefficients::Ptr coefficients, vector< float > referimentAxis)
- [PCLCloudPtr getPointOnPlane](#) ([PCLCloudPtr](#) plane, [PrimitiveldxPtr](#) inlierIdx, int mapLevel)
- bool [findSupports](#) (SupportSegmentation::Request &req, SupportSegmentation::Response &res)
- int [main](#) (int argc, char \*\*argv)

## Variables

- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_MIN\\_ITERATIVE\\_CLOUD\\_PERCENTAGE](#) = 0.030f
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_MIN\\_ITERATIVE\\_SUPPORT\\_PERCENTAGE](#) = 0.030f
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_VARIANCE\\_THRESHOLD\\_FOR\\_HORIZONTAL](#) = 0.09f
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_RANSAC\\_DISTANCE\\_POINT\\_IN\\_SHAPE\\_THRESHOLD](#) = 0.02f
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_RANSAC\\_MODEL\\_NORMAL\\_DISTANCE\\_WEIGHT](#) = 0.9f
- const int [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_RANSAC\\_MAX\\_ITERATION\\_THRESHOLD](#) = 10
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_HORIZONTAL\\_AXIS](#) [3] = { 0.0f, 0.0f, -1.0f}
- const float [DEFAULT\\_PARAM\\_SUPPORT\\_SRV\\_SUPPORT\\_EDGE\\_REMOVE\\_OFFSET](#) [] = { 0.02, 0.02, 0.005}
- float [minIterativeCloudPercentage](#)
- float [minPlanePercentageSize](#)
- float [minVarianceThForHorizontal](#)
- float [maxVarianceThForHorizontal](#)
- float [ransacThDistancePointShape](#)
- float [ransacNormalDistanceWeigth](#)
- vector< float > [horizontalAxis](#)
- vector< float > [supportEdgeRemoveOffset](#)
- int [ransacMaxIteration](#)
- [PCLVisualizer](#) vis
- [PCLCloudPtr](#) originalCloud
- [PCLNormalPtr](#) originalNormals
- boost::thread [vis\\_thread](#)
- boost::mutex [vis\\_mutex](#)
- SACSegmentationFromNormals  
< PointXYZ, Normal > [seg](#)
- const double [inf](#) = std::numeric\_limits<double>::infinity()
- const double [neg\\_inf](#) = -std::numeric\_limits<double>::infinity()

## 6.15.1 Function Documentation

### 6.15.1.1 [PrimitiveldxPtr createNewldxMap](#) ( [PrimitiveldxPtr](#) previousInliersMap, PointIndices::Ptr inliers, int level )

Definition at line 139 of file supports\_segmentation\_srv.cpp.

References [valueBelongsToArray](#)().

Referenced by [findSupports](#)().

Here is the call graph for this function:



#### 6.15.1.2 ExtractIndices<PointXYZ> extract ( true )

Referenced by removePlaneInliner().

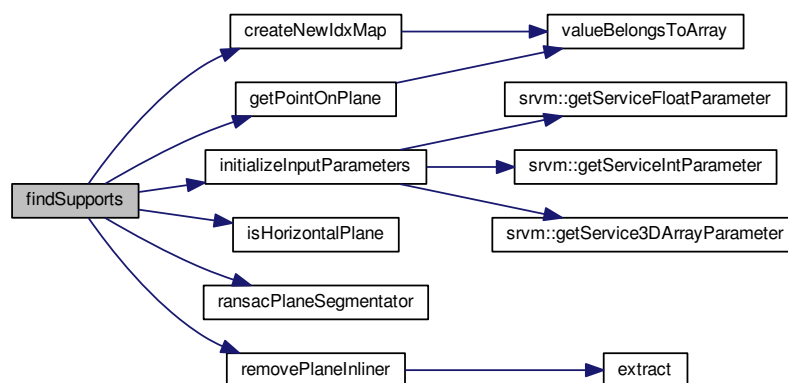
#### 6.15.1.3 bool findSupports ( SupportSegmentation::Request & req, SupportSegmentation::Response & res )

Definition at line 241 of file supports\_segmentation\_srv.cpp.

References createNewIdxMap(), getPointOnPlane(), horizontalAxis, initializeInputParameters(), isHorizontalPlane(), maxVarianceThForHorizontal, minIterativeCloudPercentage, minPlanePercentageSize, minVarianceThForHorizontal, originalCloud, originalNormals, ransacMaxIteration, ransacNormalDistanceWeigth, ransacPlaneSegmentator(), ransacThDistancePointShape, removePlaneInliner(), and supportEdgeRemoveOffset.

Referenced by main().

Here is the call graph for this function:



#### 6.15.1.4 PCLCloudPtr getPointOnPlane ( PCLCloudPtr plane, PrimitivIdxPtr inlierIdx, int mapLevel )

Definition at line 187 of file supports\_segmentation\_srv.cpp.

References inf, neg\_inf, originalCloud, supportEdgeRemoveOffset, and valueBelongsToArray().

Referenced by findSupports().

Here is the call graph for this function:



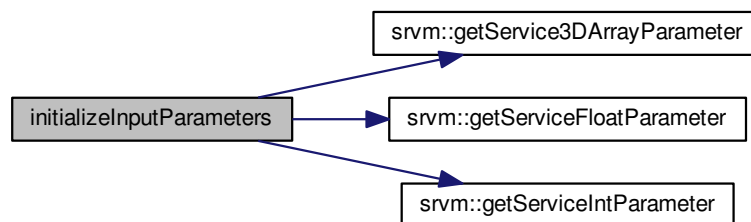
#### 6.15.1.5 void initializeInputParameters ( SupportSegmentation::Request & req )

Definition at line 70 of file supports\_segmentation\_srv.cpp.

References DEFAULT\_PARAM\_SUPPORT\_SRV\_HORIZONTAL\_AXIS, DEFAULT\_PARAM\_SUPPORT\_SRV\_MIN\_ITERATIVE\_CLOUD\_PERCENTAGE, DEFAULT\_PARAM\_SUPPORT\_SRV\_MIN\_ITERATIVE\_SUPPORT\_PERCENTAGE, DEFAULT\_PARAM\_SUPPORT\_SRV\_RANSAC\_DISTANCE\_POINT\_IN\_SHAPE\_THRESHOLD, DEFAULT\_PARAM\_SUPPORT\_SRV\_RANSAC\_MAX\_ITERATION\_THRESHOLD, DEFAULT\_PARAM\_SUPPORT\_SRV\_RANSAC\_MODEL\_NORMAL\_DISTANCE\_WEIGHT, DEFAULT\_PARAM\_SUPPORT\_SRV\_SUPPORT\_EDGE\_REMOVE\_OFFSET, DEFAULT\_PARAM\_SUPPORT\_SRV\_VARIANCE\_THRESHOLD\_FOR\_HORIZONTAL, srvm::getService3DArrayParameter(), srvm::getServiceFloatParameter(), srvm::getServiceIntParameter(), horizontalAxis, maxVarianceThForHorizontal, minIterativeCloudPercentage, minPlanePercentageSize, minVarianceThForHorizontal, ransacMaxIteration, ransacNormalDistanceWeigth, ransacThDistancePointShape, and supportEdgeRemoveOffset.

Referenced by findSupports().

Here is the call graph for this function:



#### 6.15.1.6 bool isHorizontalPlane ( PCLNormalIPtr normal, ModelCoefficients::Ptr coefficients, vector< float > referimentAxis )

Definition at line 161 of file supports\_segmentation\_srv.cpp.

References maxVarianceThForHorizontal, and minVarianceThForHorizontal.

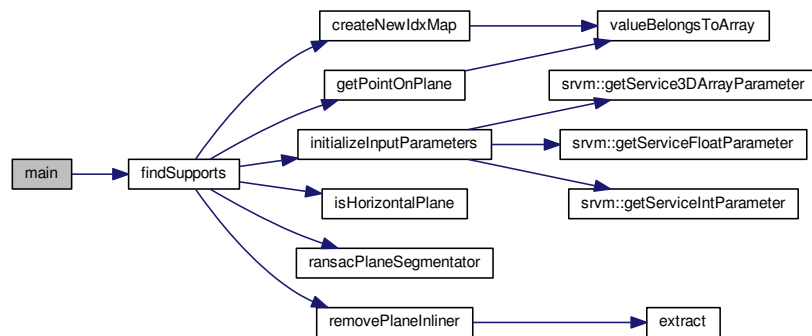
Referenced by findSupports().

#### 6.15.1.7 `int main ( int argc, char ** argv )`

Definition at line 363 of file `supports_segmentation_srv.cpp`.

References `findSupports()`, and `srvm::SRV_NAME_SUPPORT_FILTER`.

Here is the call graph for this function:



#### 6.15.1.8 `void ransacPlaneSegmentator ( PCLCloudPtr inputCloud, PCLNormalPtr normals, PointIndices::Ptr & inlierOutput, ModelCoefficients::Ptr & coefficientOutput )`

Definition at line 90 of file `supports_segmentation_srv.cpp`.

References `ransacMaxIteration`, `ransacNormalDistanceWeigth`, `ransacThDistancePointShape`, and `seg`.

Referenced by `findSupports()`.

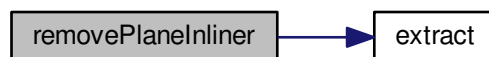
#### 6.15.1.9 `void removePlaneInliner ( PCLCloudPtr inputCloud, PointIndices::Ptr & removeIndex, PCLCloudPtr output )`

Definition at line 115 of file `supports_segmentation_srv.cpp`.

References `extract()`.

Referenced by `findSupports()`.

Here is the call graph for this function:



#### 6.15.1.10 `bool valueBelongsToArray ( int value, PointIndices::Ptr inliers )`

Definition at line 131 of file `supports_segmentation_srv.cpp`.

Referenced by `createNewIdxMap()`, and `getPointOnPlane()`.

## 6.15.2 Variable Documentation

### 6.15.2.1 `const float DEFAULT_PARAM_SUPPORT_SRV_HORIZONTAL_AXIS[3] = { 0.0f, 0.0f, -1.0f}`

Definition at line 38 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.2 `const float DEFAULT_PARAM_SUPPORT_SRV_MIN_ITERATIVE_CLOUD_PERCENTAGE = 0.030f`

Definition at line 30 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.3 `const float DEFAULT_PARAM_SUPPORT_SRV_MIN_ITERATIVE_SUPPORT_PERCENTAGE = 0.030f`

Definition at line 31 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.4 `const float DEFAULT_PARAM_SUPPORT_SRV_RANSAC_DISTANCE_POINT_IN_SHAPE_THRESHOLD = 0.02f`

Definition at line 35 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.5 `const int DEFAULT_PARAM_SUPPORT_SRV_RANSAC_MAX_ITERATION_THRESHOLD = 10`

Definition at line 37 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.6 `const float DEFAULT_PARAM_SUPPORT_SRV_RANSAC_MODEL_NORMAL_DISTANCE_WEIGHT = 0.9f`

Definition at line 36 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.7 `const float DEFAULT_PARAM_SUPPORT_SRV_SUPPORT_EDGE_REMOVE_OFFSET[] = { 0.02, 0.02, 0.005}`

Definition at line 39 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.8 `const float DEFAULT_PARAM_SUPPORT_SRV_VARIANCE_THRESHOLD_FOR_HORIZONTAL = 0.09f`

Definition at line 33 of file `supports_segmentation_srv.cpp`.

Referenced by `initializeInputParameters()`.

### 6.15.2.9 `vector<float> horizontalAxis`

Definition at line 46 of file `supports_segmentation_srv.cpp`.

Referenced by `callSupportFilter()`, `findSupports()`, and `initializeInputParameters()`.

**6.15.2.10** `const double inf = std::numeric_limits<double>::infinity()`

Definition at line 184 of file `supports_segmentation_srv.cpp`.

Referenced by `getPointOnPlane()`.

**6.15.2.11** `float maxVarianceThForHorizontal`

Definition at line 44 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`, `initializeInputParameters()`, and `isHorizontalPlane()`.

**6.15.2.12** `float minIterativeCloudPercentage`

Definition at line 44 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`, and `initializeInputParameters()`.

**6.15.2.13** `float minPlanePercentageSize`

Definition at line 44 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`, and `initializeInputParameters()`.

**6.15.2.14** `float minVarianceThForHorizontal`

Definition at line 44 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`, `initializeInputParameters()`, and `isHorizontalPlane()`.

**6.15.2.15** `const double neg_inf = -std::numeric_limits<double>::infinity()`

Definition at line 185 of file `supports_segmentation_srv.cpp`.

Referenced by `getPointOnPlane()`.

**6.15.2.16** `PCLCloudPtr originalCloud`

Definition at line 51 of file `supports_segmentation_srv.cpp`.

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

**6.15.2.17** `PCLNormalPtr originalNorms`

Definition at line 52 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`.

**6.15.2.18** `int ransacMaxIteration`

Definition at line 47 of file `supports_segmentation_srv.cpp`.

Referenced by `findSupports()`, `initializeInputParameters()`, and `ransacPlaneSegmentator()`.

**6.15.2.19 float ransacNormalDistanceWeigth**

Definition at line 44 of file supports\_segmentation\_srv.cpp.

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

**6.15.2.20 float ransacThDistancePointShape**

Definition at line 44 of file supports\_segmentation\_srv.cpp.

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

**6.15.2.21 SACSegmentationFromNormals< PointXYZ, Normal> seg**

Definition at line 89 of file supports\_segmentation\_srv.cpp.

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

**6.15.2.22 vector<float> supportEdgeRemoveOffset**

Definition at line 46 of file supports\_segmentation\_srv.cpp.

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

**6.15.2.23 PCLVisualizer vis**

Definition at line 50 of file supports\_segmentation\_srv.cpp.

**6.15.2.24 boost::mutex vis\_mutex**

Definition at line 54 of file supports\_segmentation\_srv.cpp.

**6.15.2.25 boost::thread vis\_thread**

Definition at line 53 of file supports\_segmentation\_srv.cpp.