CW1 TEAM 8

Generated by Doxygen 1.8.17

Chapter 1

Class Documentation

1.1 cw1 Class Reference

Public Member Functions

- cw1 (ros::NodeHandle nh)

  constructor
- bool t1_callback (cw1_world_spawner::Task1Service::Request &request, cw1_world_spawner::Task1↩ Service::Response &response)

  function to solve tasks 1
- bool t2_callback (cw1_world_spawner::Task2Service::Request &request, cw1_world_spawner::Task2↩ Service::Response &response)

  function to solve tasks 2
- bool t3_callback (cw1_world_spawner::Task3Service::Request &request, cw1_world_spawner::Task3↩ Service::Response &response)

  function to solve tasks 2
- void pcCallBack (const sensor_msgs::PointCloud2ConstPtr &cloud_input_msg)

  the callback function for receving the point cloud function
- bool moveArm (geometry_msgs::Pose target_pose)

  move the robot arm to target pose
- bool moveGripper (float width)

  move the gripper to a certain width
- bool pick (geometry_msgs::Point position)

  pickup the cube in certain position
- bool place (geometry_msgs::Point position)

  place the cube into a basket in certain position
- bool armGo (geometry_msgs::Point position)

  move arm to a specicfic position
- int getNearestPoint (const PointC &cloud, const pcl::PointXYZRGBA &position)

  get the nearset point's index from the cloud and target position
- void pubFilteredPCMsg (ros::Publisher &pc_pub, PointC &pc)

  publish the point cloud message using specific publisher
- void applyPT (PointCPtr &in_cloud_ptr, PointCPtr ∗out_cloud_ptr)

  apply a PT filter to the input point cloud
- void findCenter (PointCPtr &in_cloud_ptr, geometry_msgs::PointStamped ∗pose_out)

  find the center of the input point cloud

- int findColor (const PointC &cloud, const geometry_msgs::PointStamped &loc, bool move_arm=true, int cloud_num_thresh=2000)

  find the color of the a position in the given point cloud
- int searchCubesTask3 ()

  search all the cubes and store their locs and colors
- int searchBasketsTask3 ()

  search all the baskets and store their locs and colors
- bool pickPlaceCubes (int n_cube, int n_basket)

  pick and place all the cubes accroding to the searching results

Public Attributes

- ros::NodeHandle **nh_**
- ros::ServiceServer t1_service_

  node handle
- ros::ServiceServer t2_service_

  service of task1
- ros::ServiceServer t3_service_

  service of task2
- moveit::planning_interface::MoveGroupInterface arm_group_ {"panda_arm"}

  service of task3
- moveit::planning_interface::MoveGroupInterface hand_group_ {"hand"}

  arm group ("panda_arm") in moveit
- float gripper_open_ = 80e-3

  arm group ("hand") in moveit
- float gripper_closed_ = 0.0

  safe value for the open size of gripper
- double angle_offset_ = 3.14159 / 4.0

  safe value for the closed size of gripper
- double z_offset_ = 0.125

  angle offset for grasping orentation
- double approach_distance_ = 0.1

  z-axis offset for the grasping pose
- std::string fram_id_

  the pre-grasping distance
- pcl::PCLPointCloud2 pcl_pc_

  frame id for the recevied cloud
- PointCPtr cloud_ptr_

  point cloud data in PCL
- tf::TransformListener listener_

  point cloud data pointer
- ros::Publisher pub_cloud_

  TF listener.
- ros::Publisher pub_seg_

  for publishing the filtered cloud
- pcl::PassThrough< PointT > pt_

  for publishing the seg of the first cube
- PointCPtr cloud_filtered_

  Pass Through filter.
- geometry_msgs::PointStamped pose_color

filtered cloud pointer

- geometry_msgs::Point basket_locs [4]

  the pose of the seg to be found the center
- geometry_msgs::Point cube_locs [100]

  store the locations of all the baskets
- int basket_colors [4]

  store the locations of all the cubes
- int cube_colors [100]

  store the colors of all the baskets

1.1.1 Member Function Documentation

1.1.1.1 applyPT()

```
void cw1::applyPT (
            PointCPtr & in_cloud_ptr,
            PointCPtr * out_cloud_ptr )
```

apply a PT filter to the input point cloud

Parameters

in_cloud_ptr	input point cloud to be filtered
out_cloud_ptr	output point cloud pointer

1.1.1.2 armGo()

```
bool cw1::armGo (
            geometry_msgs::Point position )
```

move arm to a specicfic position

Parameters

position	the target position of the arm

Returns

true if sucessful

1.1.1.3 findCenter()

```
void cw1::findCenter (
            PointCPtr & in_cloud_ptr,
            geometry_msgs::PointStamped * pose_out )
```

find the center of the input point cloud

Parameters

in_cloud_ptr	input cloud message from color frame
pose_out	ouput pose pointer

1.1.1.4 findColor()

```
int cw1::findColor (
            const PointC & cloud,
            const geometry_msgs::PointStamped & loc,
            bool move_arm = true,
            int cloud_num_thresh = 2000 )
```

find the color of the a position in the given point cloud

Parameters

cloud	input cloud
loc	position to be searched
move_arm	if ture, move arm above the loc to calculate the more accurate color
cloud_num_thresh	the threshold to judge a cloud is empty or not

Returns

the color (1 for red, 2 for blue, 3 for pink, 4 for empty and -1 for error)

1.1.1.5 getNearestPoint()

```
int cw1::getNearestPoint (
            const PointC & cloud,
            const pcl::PointXYZRGBA & position )
```

get the nearset point's index from the cloud and target position

Parameters

cloud	the point cloud data
position	the target position to be searched

Returns

the index of the nearest point

1.1.1.6 moveArm()

```
bool cw1::moveArm (
            geometry_msgs::Pose target_pose )
```

move the robot arm to target pose

Parameters

target_pose	the target pose

Returns

true if sucessful

1.1.1.7 moveGripper()

```
bool cw1::moveGripper (
            float width )
```

move the gripper to a certain width

Parameters

width	the width of the gripper

Returns

true if sucessful

1.1.1.8 pcCallBack()

```
void cw1::pcCallBack (
            const sensor_msgs::PointCloud2ConstPtr & cloud_input_msg )
```

the callback function for receving the point cloud function

Parameters

cloud_input_msg	input cloud message from color frame

1.1.1.9 pick()

```
bool cw1::pick (
            geometry_msgs::Point position )
```

pickup the cube in certain position

Parameters

position	the position of cube to be picked up

Returns

true if sucessful

1.1.1.10 pickPlaceCubes()

```
bool cw1::pickPlaceCubes (
            int n_cube,
            int n_basket )
```

pick and place all the cubes accroding to the searching results

Parameters

n_cube	the number of cubes
n_basket	the number of baskets

Returns

true if sucessful

1.1.1.11 place()

```
bool cw1::place (
            geometry_msgs::Point position )
```

place the cube into a basket in certain position

Parameters

position	the position of the basket

Returns

true if sucessful

1.1.1.12 pubFilteredPCMsg()

```
void cw1::pubFilteredPCMsg (
            ros::Publisher & pc_pub,
            PointC & pc )
```

publish the point cloud message using specific publisher

Parameters

pc_pub	the point cloud publisher
pc	the point cloud

1.1.1.13 searchBasketsTask3()

```
int cw1::searchBasketsTask3 ( )
```

search all the baskets and store their locs and colors

Returns

the number of baskets

1.1.1.14 searchCubesTask3()

```
int cw1::searchCubesTask3 ( )
```

search all the cubes and store their locs and colors

Returns

the number of cubes (boxes)

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

- include/cw1_class.h

Index