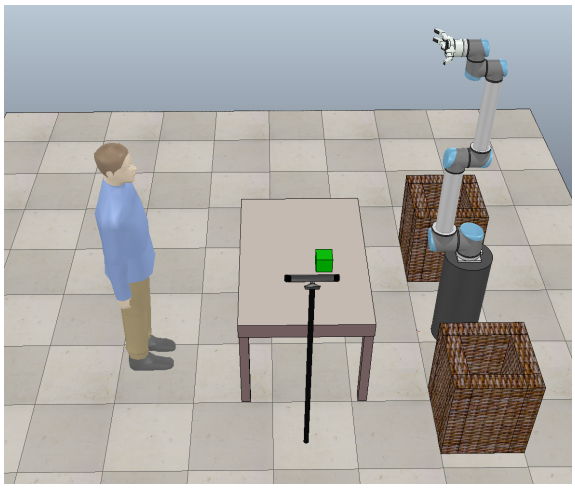
Robotics Assignment 2

1 Introduction

In this assignment you will control UR10, you will satisfy Bill's pick and place orders. You will use UR10.ttt for simulation.



Bill has powers to spawn cubes. He can also talk to robots via ROS topics.

In this task, he wants UR10 to grab cubes he has spawned and put them in the correct baskets, which he whispers to robot. He will also give points to UR10 whenever UR10 puts a cube in the correct basket.

2 Your Task

Your task is to help UR10 get all points he can get. You will have access to kinect point cloud data, and bills whispers. You can assign joint angles for robot to go.

You are expected to do following

1. Detect the cube.
2. Estimate the cube's position.
3. Pick the cube.
4. Drop the cube to correct basket.
5. Collect all the points.

For simplicity, kinect will not see the table that cubes are on

3 Steps

3.1 Subscribe to necessary ROS topics

You will need to subscribe to following topics.

- /kinect/depth: Use these to estimate cube position.

- /Cube_Target_Basket: Use these to drop the cube to the correct basket.
- ScoreTopics(/Accumulated_Points and /Total_Points): Use these to write your score and accuracy.(Your Score divided by possible score you could get).

You can look for basic tutorials related to writing subscribers in following page.
http://wiki.ros.org/ROS/Tutorials

3.2 Cube Position Estimation using Point cloud

You will first use TF to transform point cloud given in kinect depth frame('kinect_depth') to robot base frame('UR10_base'), you can then use PCL library to process cloud.

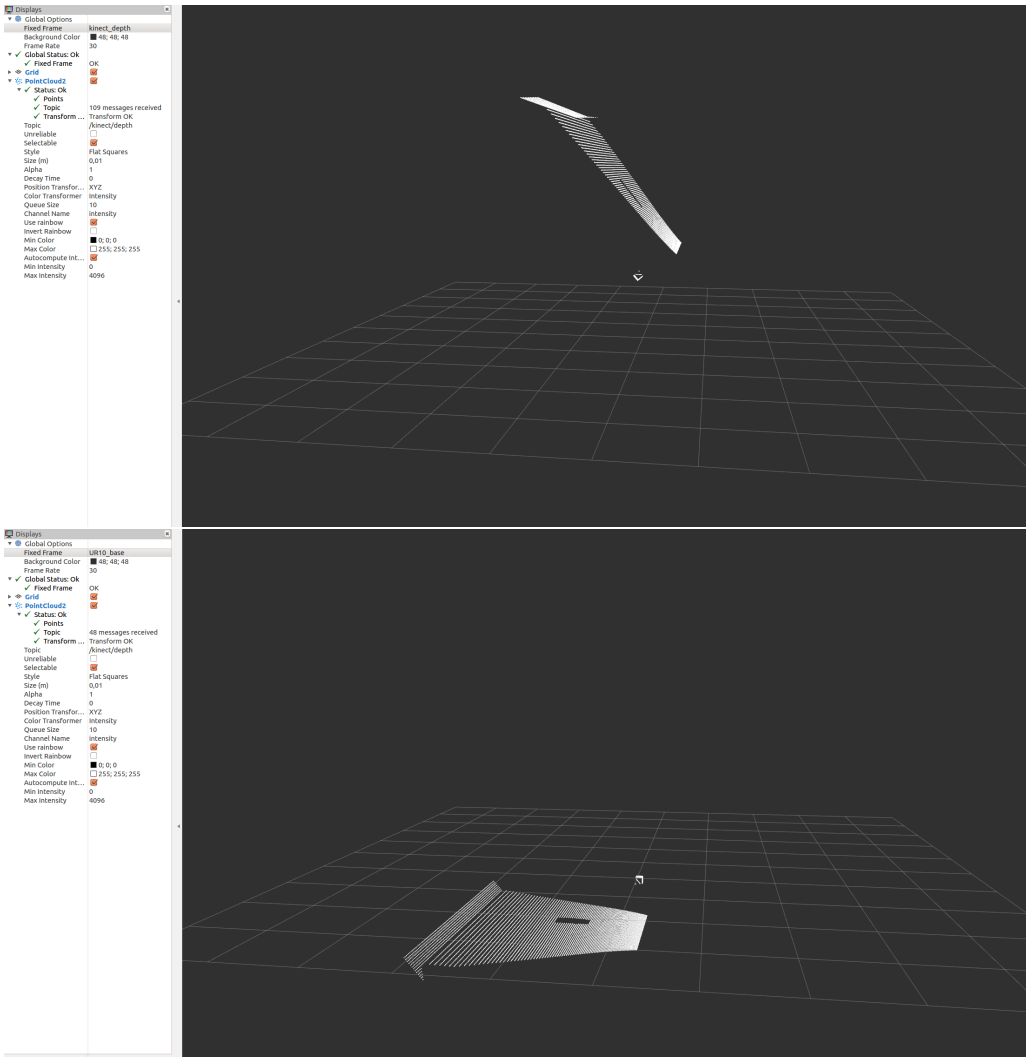


Figure 1: Up: Before transforming point cloud, Down: After transforming point cloud

http://wiki.ros.org/tf
http://docs.ros.org/kinetic/api/pcl_ros/html/transforms_8h.html
http://wiki.ros.org/pcl

You will calculate object position using cloud, simplest way to do this is to remove points out based on their x, y, z (Basically remove any point that is far from the table) and use remaining ones to estimate cube position.

3.3 Pick and Place actions

In here you are provided with forward kinematic model of the robot. You can experiment on how robot goes between given angles. Then you will code the inverse kinematic model preferably using

KDL library to find angles given end effector position. KDL library doesn't have to be used but forward kinematic model is provided in KDL library so it will be probably easier to do it with KDL library (You can use the same kinematic chain used in forward kinematic model in inverse kinematic model).

http://www.orocos.org/kdl

You will also send a number to robot gripper between 0 and 10 for robot to close its gripper and open its gripper.

Considering above you are expected to code pick and a drop action for robot.

3.4 Complete the task loop

Since cubes will continue to spawn, it is expected that your code should be able continue picking and dropping cubes. You should also be able to printout current score of robot and its accuracy.