Group Assignment Report

Bharath Kumar Ramesh Babu
Akshay Kumar Harikrishnan

Problem Statement:

Given a robot simulator with integrated controllers and path planners, implement the following

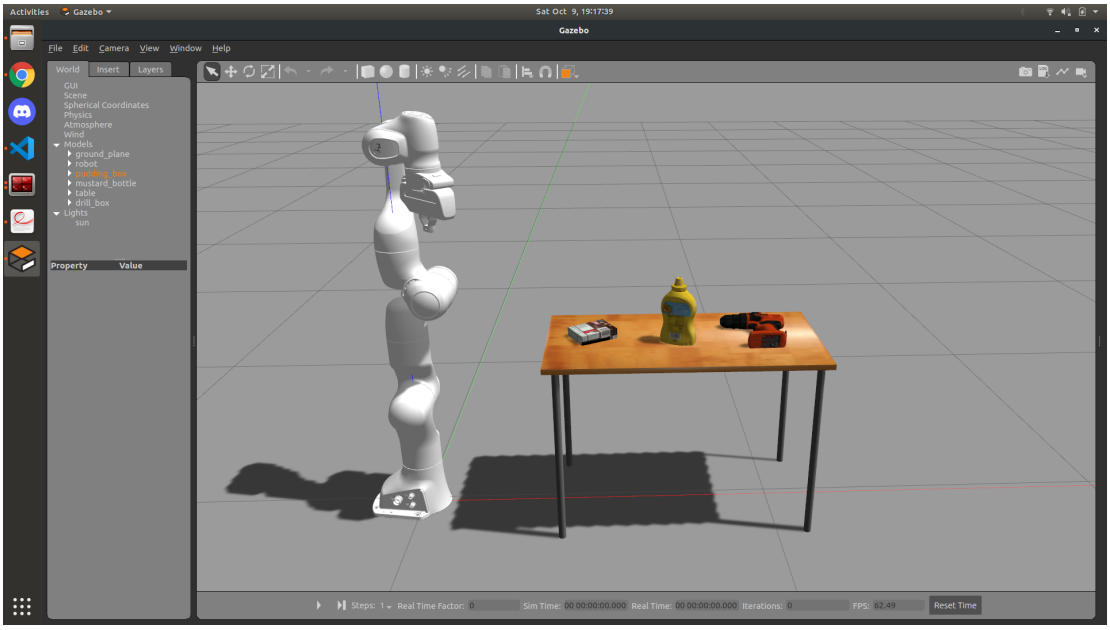
1. Spawn the robot. Spawn a table in front of the robot. Then spawn the three YCB objects, one at a time, in front of the robot.

2. Spawn only the table and the pudding box in front of the robot. Be sure that the pudding box is standing up. Since you spawned this object you know its pose. Make the robot end effector go to the object. Close the gripper to initiate the grasp.

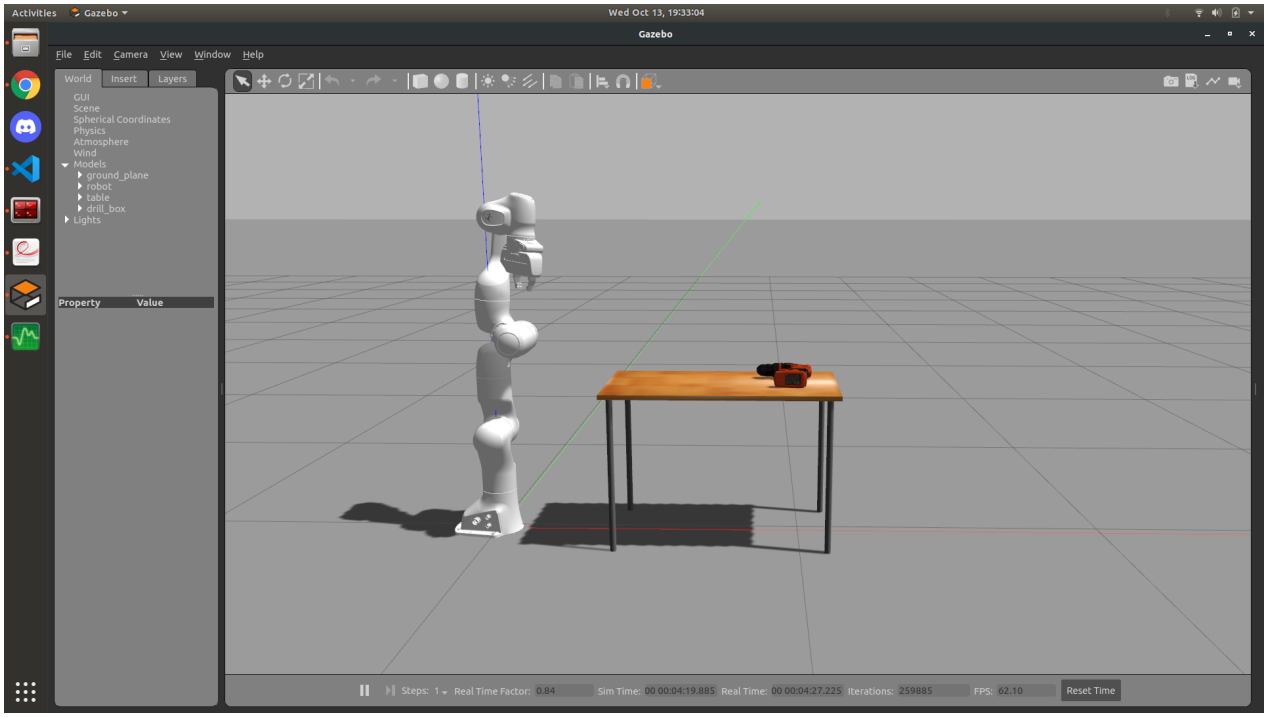
3. Start the robot far away from the object, where its camera is observing the table scene from a distance with a pudding box spawned on it. Capture a point cloud image of the scene. Run RANSAC algorithm of PCL. Detect the major plane. Remove it from the scene. In this way the object should be segmented from the scene.

Deliverables:

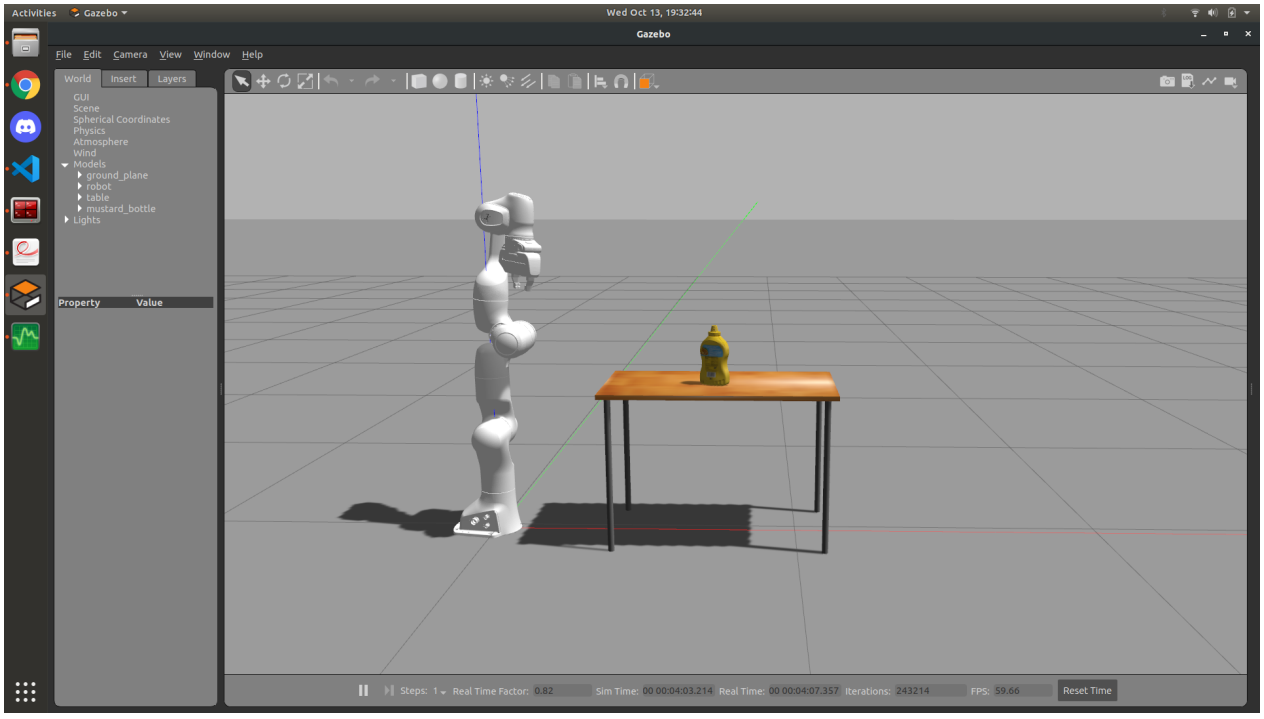
1. Snapshots of the gazebo environment for each spawned object



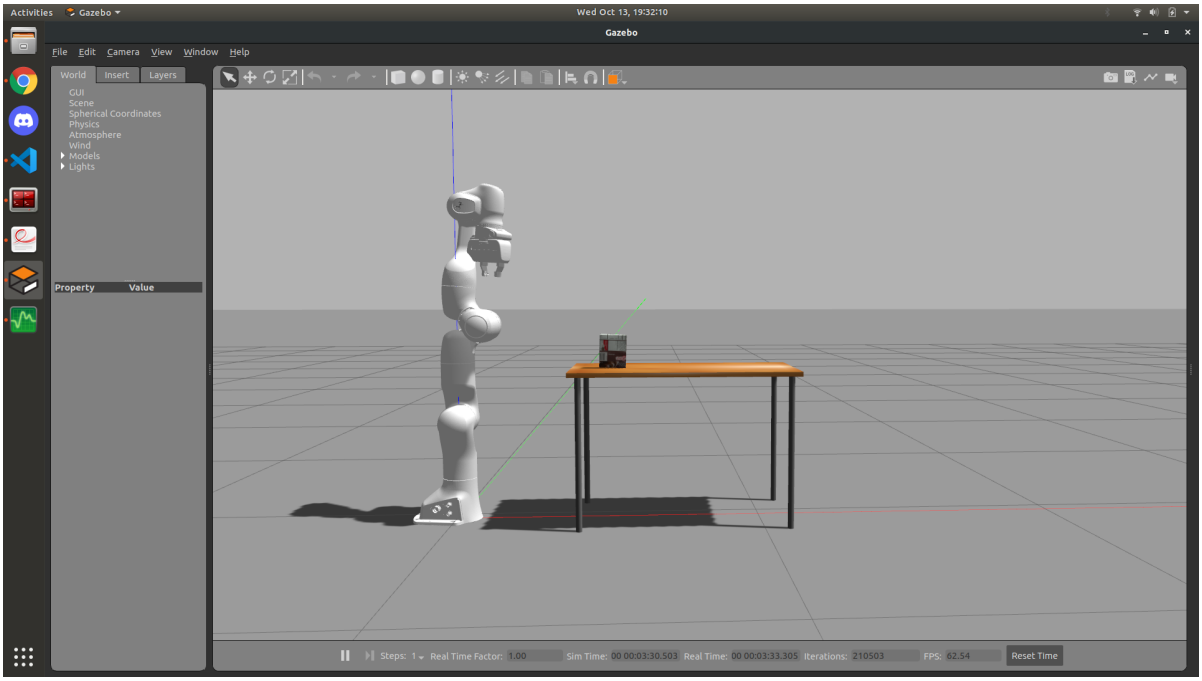
Spawning all three objects at a time



Spawning Power Drill



Spawning Mustard Bottle



Spawning Pudding Box

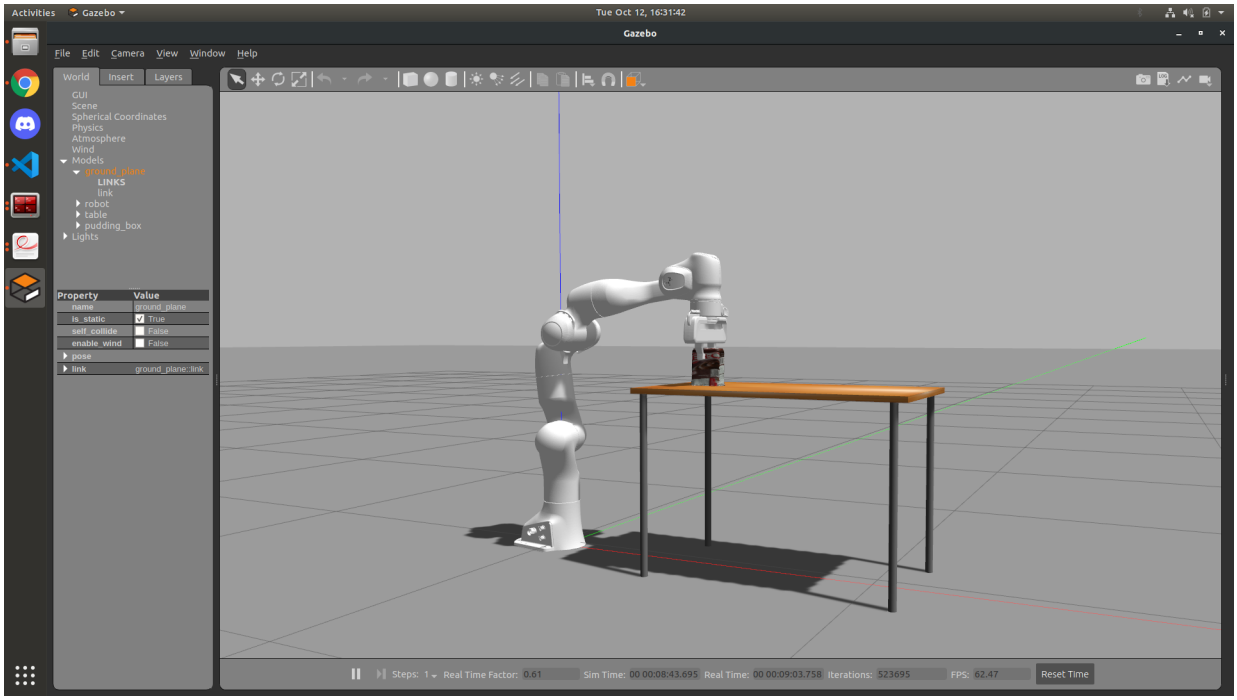
The Panda robot is spawned at the origin using the packages provided for the assignment. The files provided along with the assignment have all the control, gazebo simulation and moveit plugins/packages inbuilt. Running the launch file provides all the topics and services to control the arm seamlessly.

Table SDF file from the gazebo_worlds ros package is copied to this package. Spawn_urdf node from the gazebo_ros package takes the urdf file locations and the coordinate points as input and spawn the objects in gazebo.

The object files and material files for pudding box, mustard bottle and power drill are acquired from the Yale dataset. The URDF files for the respective objects are created. A roslaunch file is created that spawns the objects over the table for the Panda robot to sense and pick the objects.

2. Snapshot of the grasp in the gazebo simulator

The assignment files come along with a moveit package, control configuration files and robot description and configuration files required for the computation of inverse kinematics and trajectory control.



Panda robot grasping the pudding box.

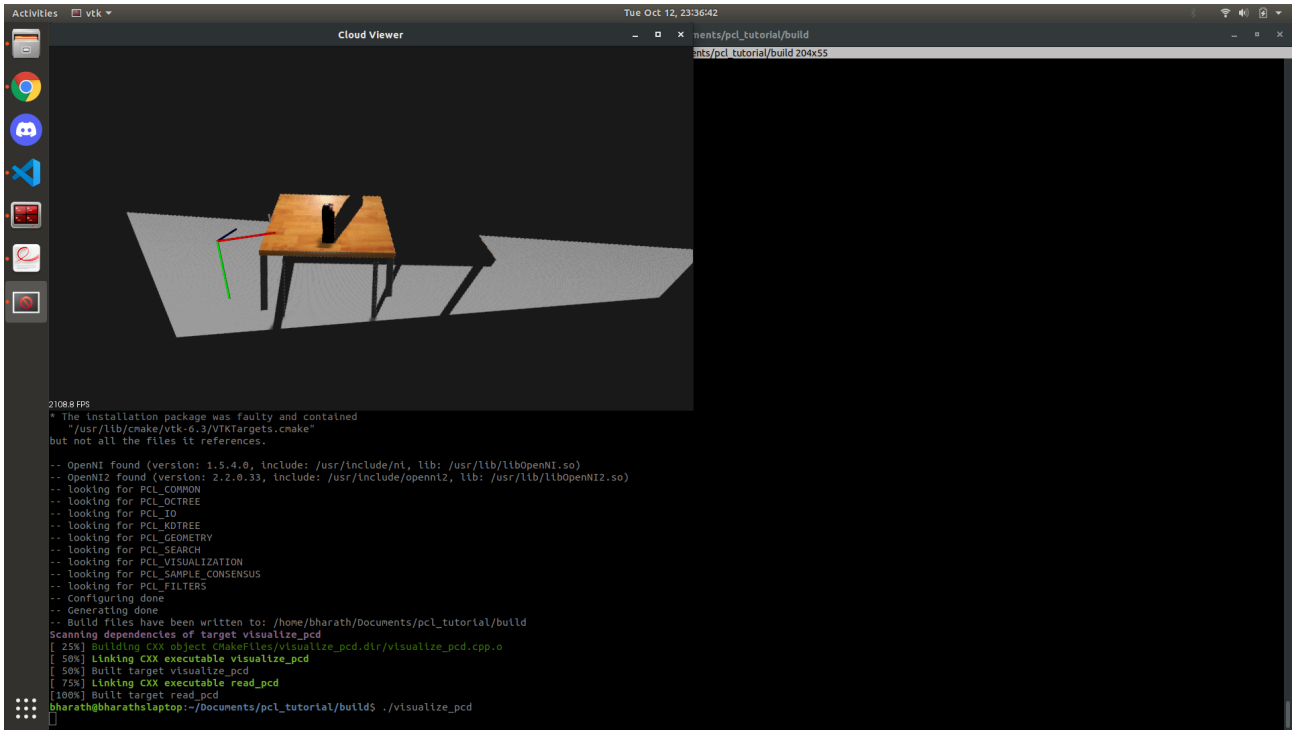
The moveit package provides a set of classes and methods that allow us to control the joint states and the end effector pose state. A table is spawned near the robot, a pudding box is spawned over the table. Since we know the pose of the pudding box from the launch file.

We can directly use the go-to pose method from the moveit package to place the gripper of the robot right above the pudding box. The end effector is then lowered and the gripper is closed over the sides of the pudding box.

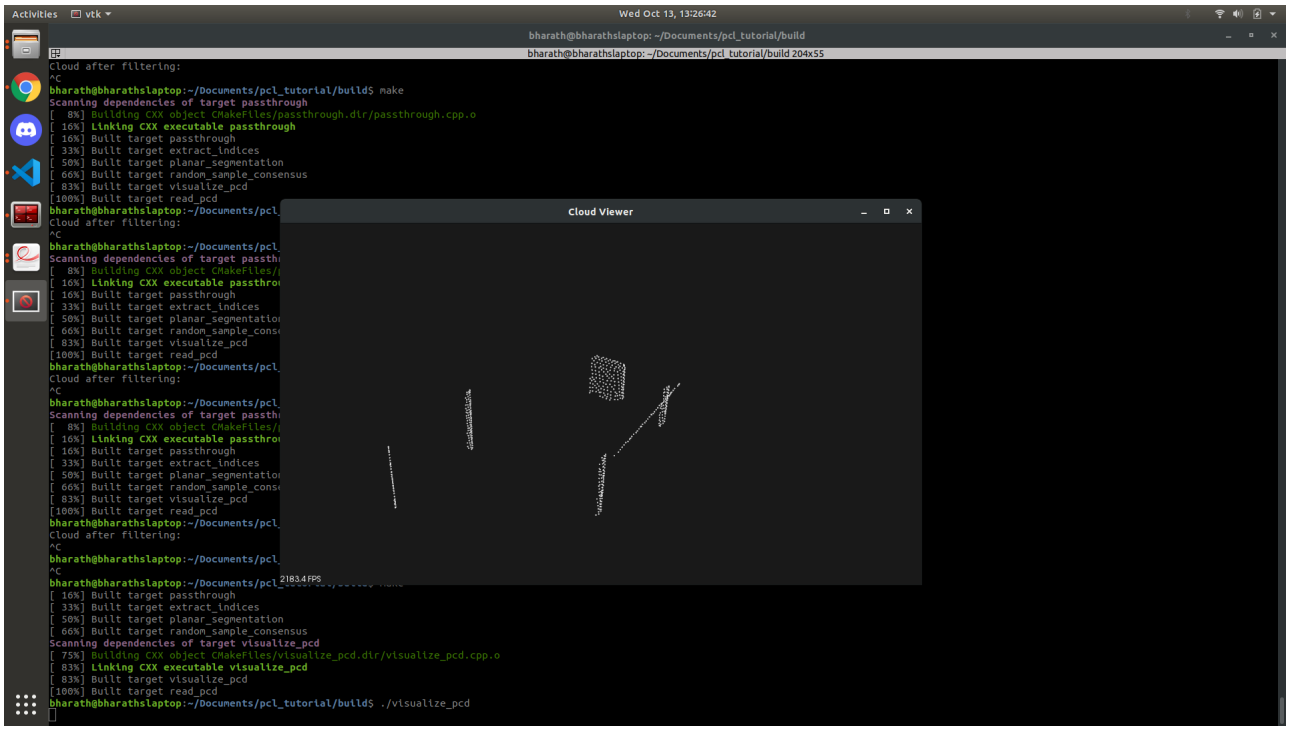
3. Visualize in the PCL visualizer, the scene without the segmentation, the segmented table plane, the segmented object plane

The point cloud data from the RGBD sensor is available from the topic /panda_camera/depth/points. pointcloud_to_pcd node from the pcl_ros wrapper package is used to convert the point cloud topic to pcd files.

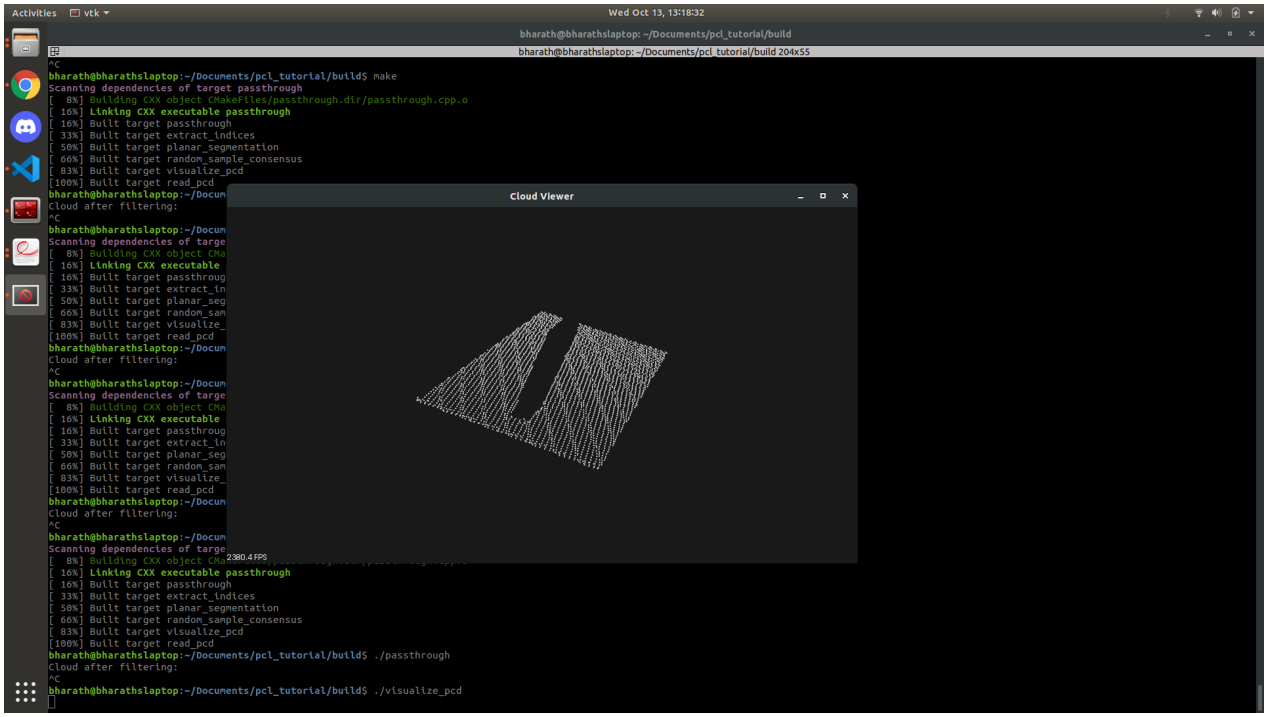
The point information along with the width and height of the point cloud is written into the pcd file. The pcd file is read using the point cloud library. The point cloud is downsampled for faster computation. The downsampled point cloud is visualized using pcl visualizer.



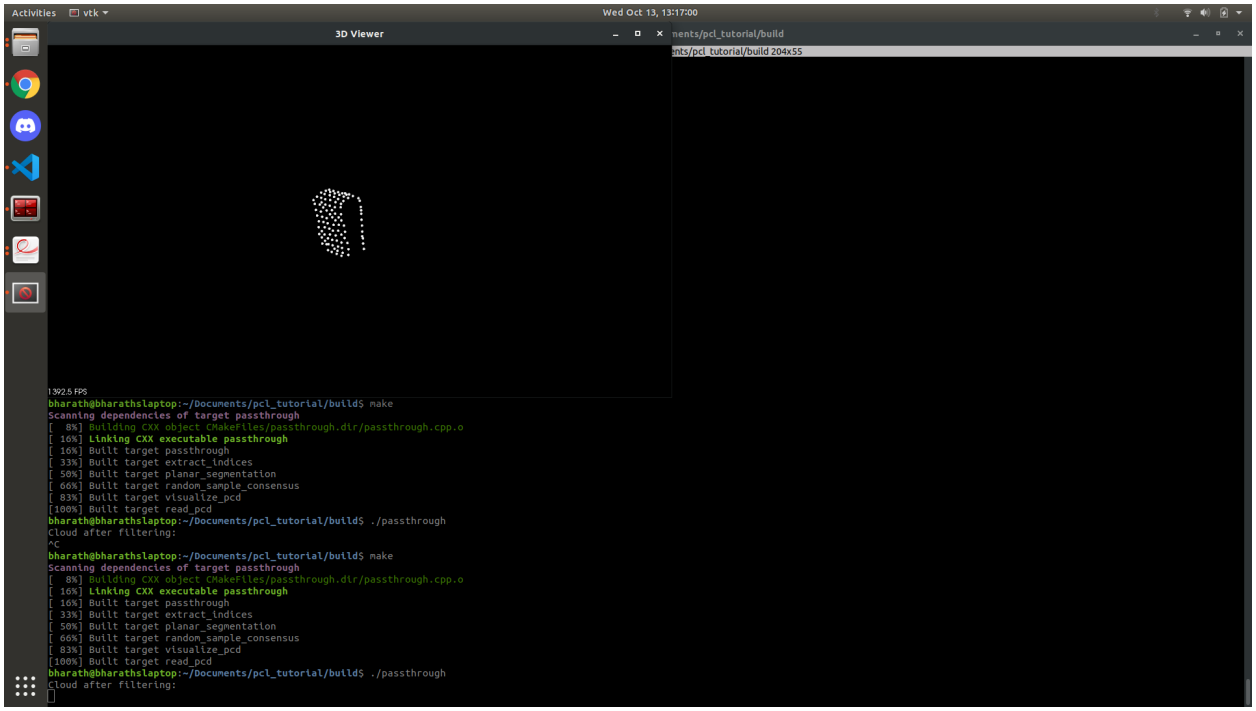
The point cloud of the entire scene



The point cloud of the scene after removing the major planes (Table and Floor) with unwanted points.



The point cloud of the major plane (Table)



The point cloud of the object after filtering unwanted points from the segmented point cloud

Random Sampling Consensus (RANSAC) algorithm is used over the point cloud. The randomly sampled points are made to fit over a planar model. Once a sufficient number of points fit the model, the parameters are recomputed and the final inliers are calculated.

The inliers are filtered out and segmented from the outliers. The inliers and outliers are visualized separately. The RANSAC algorithm runs in a loop until 10 percent of the points from the original cloud are left unfiltered. This looping allowed us to filter out the ground plane and the table plane.

After filtering the major planes, we were left with the object and other unwanted points (table legs). The table legs are filtered using the passthrough filter. The threshold is selected in such a way that the points below the table plane are filtered out leaving only the object. This way the object and the table are filtered separately.