3D Perception Project Report

Dong-Won Shin

1. Filtering and RANSAC plane fitting

I implemented the code block related to the filtering and RANSAC plane fitting at the execute\_segmentation function in the helper\_segmentation.py file. In this file, I first converted the ros message to pcl structure, and applied the voxel grid filter and the pass through filter. After that, the RANSAC plane fitting was performed on the filtered point cloud. Finally, this function returns the separated plane and object point clouds. Figure 1 and Figure 2 show the point cloud for the table and objects, respectively.



Figure Point cloud for the table



Figure Point cloud for the objects

1. Clustering

I implemented the code block related to the clustering procedure at the execute\_clustering function in the helper\_clustering.py file. I constructed the kd-tree on the input point cloud and performed the Euclidean clustering. I used the parameters for the min cluster size=50, the max cluster size=4000, the cluster tolerance=0.025. Figure 3 shows the clustering result on the objects point clouds.



Figure Clustering and recognition result for the object point cloud

1. Recognition

I implemented the code block related to the recognition procedure at the execute\_recognition function in the helper\_recognition.py file. In this function, the for loop on the cluster\_indices is perfomed. For the point cloud corresponding to each cluster index, the feature vector is computed on the create\_feature function. In the create\_feature function the color histogram and normal histogram is calculated and those are concatenated. In the make\_prediction function, it classifies a cluster by the computed features. The pre-trained model from SVM classifier is used (model.sav file). Figure 3 also shows the recognition result via labels above each object.

* Confusion matrix

1. Confusion matrix for the test\_world1



1. Confusion matrix for the test\_world2



1. Confusion matrix for the test\_world3

