UDACITY

PROJECT

3D Perception

A part of the Robotics Software Engineer Program

PROJECT REVIEW

CODE REVIEW 1

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Congratulations on successfully completing this project. You certainly did a *good job.*

Advanced tips:

- Object detection using deep learning
- Object Classification with 3D cloud and CNN
- Volumetric Shapes
- PointNet

Some fun application

Writeup

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled. The writeup should include a discussion of what worked, what didn't and how the project implementation could be improved going forward.

Your `writeup` is very well structured and documented. It looks good and addresses well to each and every rubric point well. Good job in putting up such a wonderful report.

Exercise 1, 2 and 3 Pipeline Implemented

The `pcl_callback()` function within the template Python script has been filled out to include filtering and RANSAC plane fitting. Not required, but to help your reviewer consider adding screenshots of output at different steps in your writeup with brief explanations.

`pcl_callback()` function block is very well implemented.

Steps for cluster segmentation have been added to the `pcl_callback()` function in the template Python script. Not required, but to help your reviewer consider adding screenshots of output at different steps in your writeup with brief explanations.

Cluster segmentation is well executed.

Both `compute_color_histograms()` and `compute_normal_histograms()` functions have been filled out and SVM has been trained using `train_svm.py`. Please provide a snapshot of your normalized confusion matrix (output from `train_svm.py` in your writeup / README. Object recognition steps have been implemented in the `pcl_callback()` function within template Python script. Not required, but to help your reviewer consider adding screenshots of output at different steps in your writeup with brief explanations.

Both `compute_color_histograms()` and `compute_normal_histograms()` functions were filled with codes to return normalized feature vectors .Nice work

Pick and Place Setup

You can add this functionality to your already existing ros node or create a new node that communicates with your perception pipeline to perform sequential object recognition. Save your PickPlace requests into `output_1.yaml`, `output_2.yaml`, and `output_3.yaml` for each scene respectively. Add screenshots in your writeup of output showing label markers in RViz to demonstrate your object recognition success rate in each of the three scenarios. Note: for a passing submission, your pipeline must correctly identify 100% of objects in `test1.world`, 80% (4/5) in `test2.world` and 75% (6/8) in `test3.world`.

Good job by being able to detect all objects(7/8 World 3) in all three world scenario 👏🏽

DOWNLOAD PROJECT

1 CODE REVIEW COMMENTS ›

RETURN TO PATH

Student FAQ