UDACITY

PROJECT

3D Perception

A part of the Robotics Software Engineer Program

PROJECT REVIEW

CODE REVIEW 3

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Well done for a good first submission! Just a few things to fix-up and you should be fine. Keep up the good work!

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.

The report is written very well, with nice explanations and a good mixture of diagrams and code samples included. However the future work section is missing. Please discuss how you could attempt to further improve the project in the future.

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.

Awesome work here! 👏

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 has been implemented correctly! 😄

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.

Based on your final results I expect you have implemented this correctly. However the features.py code file is missing from the submission, so I can't verify this. Also the screenshot of the confusion matrix is missing from the report. Please add both for next time!

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`.

Well done for showing 7/8 performance in world 3! From the .yaml files of world 1 and world 2 they also look probably OK, but please add screenshots of them to your report too for verification (final screenshot only with label is sufficient).

Also, it seems like you have separately trained a classifier for each world individually. This is not the expected situation. You should only have one model that contains all objects and therefore can be used in all the different worlds. Model 3 contains everything, so you can re-use that model for worlds 1 & 2. Probably you will get the same results, but please confirm and use those for the final submission.

This is important if you consider real-world use-cases of such models. We won't normally know exactly what subset of objects are on the table, so we have to have one model that contains all possibilities.

RESUBMIT

DOWNLOAD PROJECT

3 CODE REVIEW COMMENTS ›

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Student FAQ