UDACITY

Return to Classroom

DISCUSS ON STUDENT HUB

Lidar Obstacle Detection

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on completing this assignment! Understanding, filtering and processing point clouds is not an easy skill to master and you've shown fluency in this assessment. I enjoyed reading through your source. Take note that I was your previous reviewer so there are no further comments to your source as I've reviewed it already. Good luck and happy learning!

Compiling and Testing

The project code must compile without errors using `cmake` and `make`.

Obstacle Detection

Bounding boxes enclose vehicles, and the pole on the right side of the vehicle. There is one box per detected object.

Excellently done! The output is what is expected and you've detected the pole now.

Most bounding boxes can be followed through the lidar stream, and major objects don't lose or gain bounding boxes in the middle of the lidar stream.

As seen above, excellent!

The code used for segmentation uses the 3D RANSAC algorithm developed in the course lesson.

The code used for clustering uses the Euclidean clustering algorithm along with the KD-Tree developed in the course lesson.

Code Efficiency

Your code does not need to sacrifice comprehension, stability, or robustness for speed. However, you should maintain good and efficient coding practices when writing your functions.

Here are some things to avoid. This is not a complete list, but there are a few examples of inefficiencies.

- Running the exact same calculation repeatedly when you can run it once, store the value and then reuse the value later.
- Loops that run too many times.
- Creating unnecessarily complex data structures when simpler structures work equivalently.
- Unnecessary control flow checks.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START