UDACITY

PROJECT SPECIFICATION

Lidar Obstacle Detection

Compiling and Testing

CRITERIA	MEETS SPECIFICATIONS
The submission must compile.	The project code must compile without errors using `cmake` and `make`.

Obstacle Detection

CRITERIA	MEETS SPECIFICATIONS
Bounding boxes enclose appropriate objects.	Bounding boxes enclose vehicles, and the pole on the right side of the vehicle. There is one box per detected object.
Objects are consistently detected across frames in the video.	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.

CRITERIA	MEETS SPECIFICATIONS
Segmentation is implemented in the project.	The code used for segmentation uses the 3D RANSAC algorithm developed in the course lesson.
Clustering is implemented in the project.	The code used for clustering uses the Euclidean clustering algorithm along with the KD-Tree developed in the course lesson.

Code Efficiency

CRITERIA	MEETS SPECIFICATIONS
The methods in the code should avoid unnecessary calculations.	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.