The way I tuned the pre-processing algorithms was pretty much trial and error.

1. To start, I ran the program using the values given in the open3d point cloud tutorial.
2. I changed 1 value to a very small number (e.g. anywhere between 0.01 – 2 depending on the value) to see what happened.
3. I changed the same value to a very large number (e.g. 1000) to see what happened.
4. I narrowed down values to optimal ranges by increasing or decreasing from the extreme that most closely represented what seemed like the desired outcome.

Once I had an initial, complete program, I re-tweaked some of the values after seeing how each section related to the others.

## **Filtering**

***Down-sampling***

Initially, the downsizing value was 0.05. This didn’t seem to do much so I tried a larger number (20) and most of the pcd disappeared. Then I tried 0.1 and that also looked like it didn’t do much, so I tested 0.2 which seemed significantly lower in quality so I reverted back to 0.1. I then noticed that while downsizing by 0.1 didn’t look very different, it reduced the pcd from 119978 to 49077 points, making it faster to process

***Cropping***

I used the crop feature to remove obvious outliers in the y-plane. Initially, I did it in the z-plane however I found it difficult to remove the y-plane outliers in the statistical outlier removal and therefore swapped them.

***Removing outliers***

I decided to use both statistical and radial outlier removal methods because I found that the statistical method was better at removing ‘decayed’ objects near the outer edges of the cloud, while radial outlier removal was better at removing isolated sections throughout the cloud (e.g. the small spray of points in the centre circle of the pcd).

Originally, I was more conservative with outlier removal, trying only to remove the ‘fraying’ edges of the pcd. However, for segmentation and clustering, I decided that it would be better to have clarity of objects closer to the vehicle rather than depth of perception and so I decided to trim the outlying edges of the pcd more aggressively (see image comparison below).

Scatter chart

Description automatically generatedShape, arrow

Description automatically generated

**Segmentation**

The first thing I did was change the number of iterations from 1000 to 2000 – as with any random sampling, the more iterations, the more accurate the result. Similarly, I believed that the more random samples, the better – so I increased the ransac\_n to 250 from 3 and noticed that it identified more of the road. The final thing I did was tune the threshold value until the entire road was selected and there was minimal loss of details on the objects (such as car wheels). However, when presented with the choice to remove more of the road or more of the surrounding vehicles, I chose to remove more of the road.

**Clustering**

To tune the clustering, I first thought about the amount of points needed to highlight a the smallest surrounding vehicle. I estimated it to be about 30. However, with the default eps value of 0.05, nothing was being selected. So I increased the value to 0.5 and could see some coloured clusters appearing, although they were incomplete clusters – either splitting a vehicle into 2 or not identifying a complete wall. I thought this was an issue with the min\_points being too high so I lowered it to 20, which made a small difference but it wasn’t significant. Then I changed the epsilon value and realised that it had been too small, so I increased it to 1. Then, as described previously, I made small adjustments to the values and eventually settled on 0.75 and 20.

Sometimes, some of the road wasn’t segmented out properly – this changed every time I ran the program – and so I made sure that at least they weren’t identified as clusters.

**Key Resources:**

* <http://www.open3d.org/docs/release/tutorial/geometry/pointcloud.html#Point-cloud>
* <http://www.open3d.org/docs/release/tutorial/geometry/pointcloud_outlier_removal.html>
* <http://www.open3d.org/docs/release/tutorial/visualization/interactive_visualization.html>