Preprocess: remove useless data

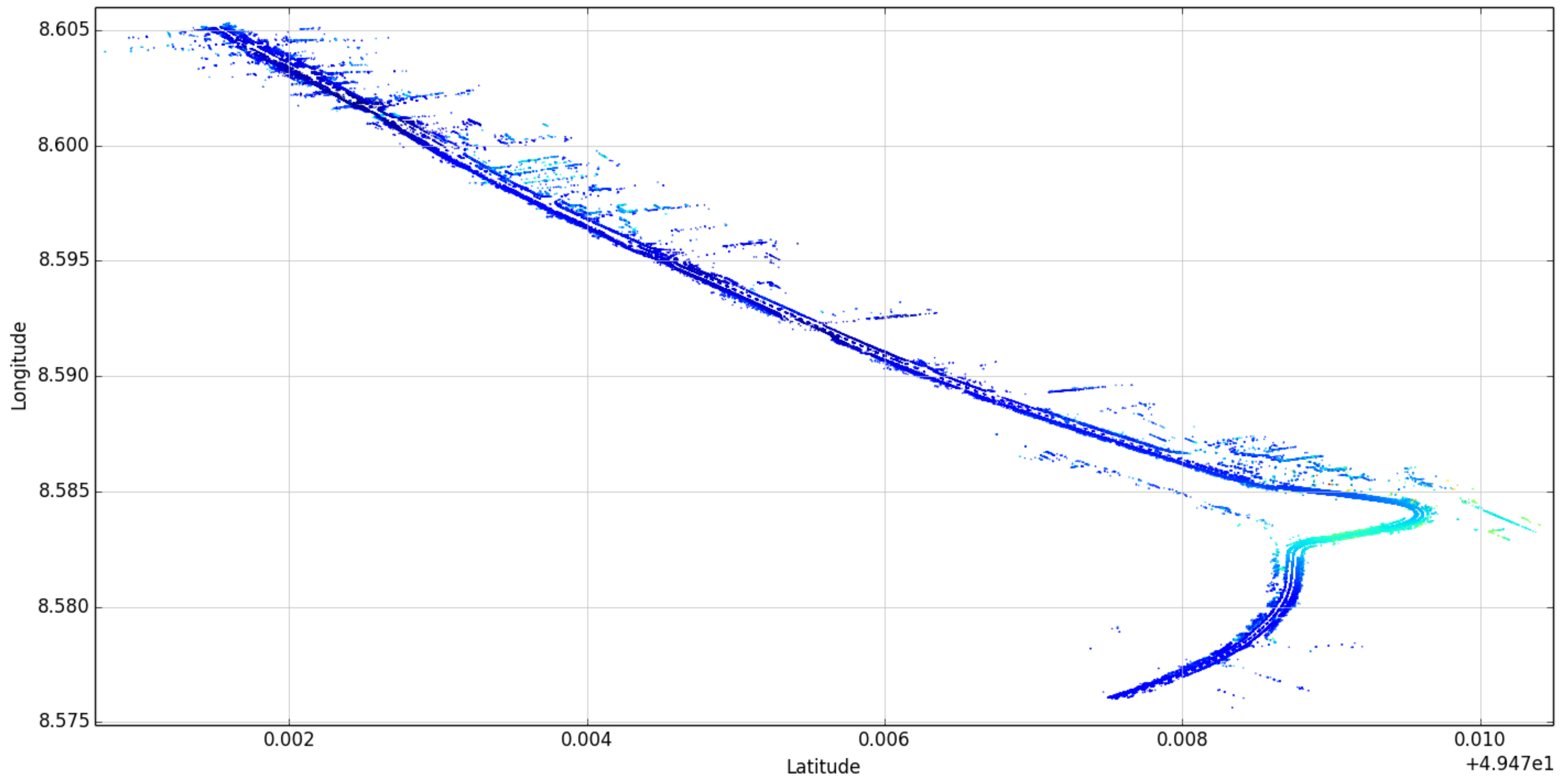
Idea: reduce the amount of points in the point cloud to make it more easy to process.

Solution: delete points that won't help to find the road boundaries.
Criterions:

- Remove points that have an intensity value less than a threshold
- Remove points that are below/above a threshold

Problem: The car elevation is not the same during the entire data acquisition, it changes with the road elevation ➔ preprocess the chunk one by one and use a variable threshold for the elevation: keep only the points that have an elevation between the 35th percentile and the 65th percentile.

Preprocess: remove useless data



Noise removal

Idea: delete the noise we have on the previous results to improve the process of line fitting.

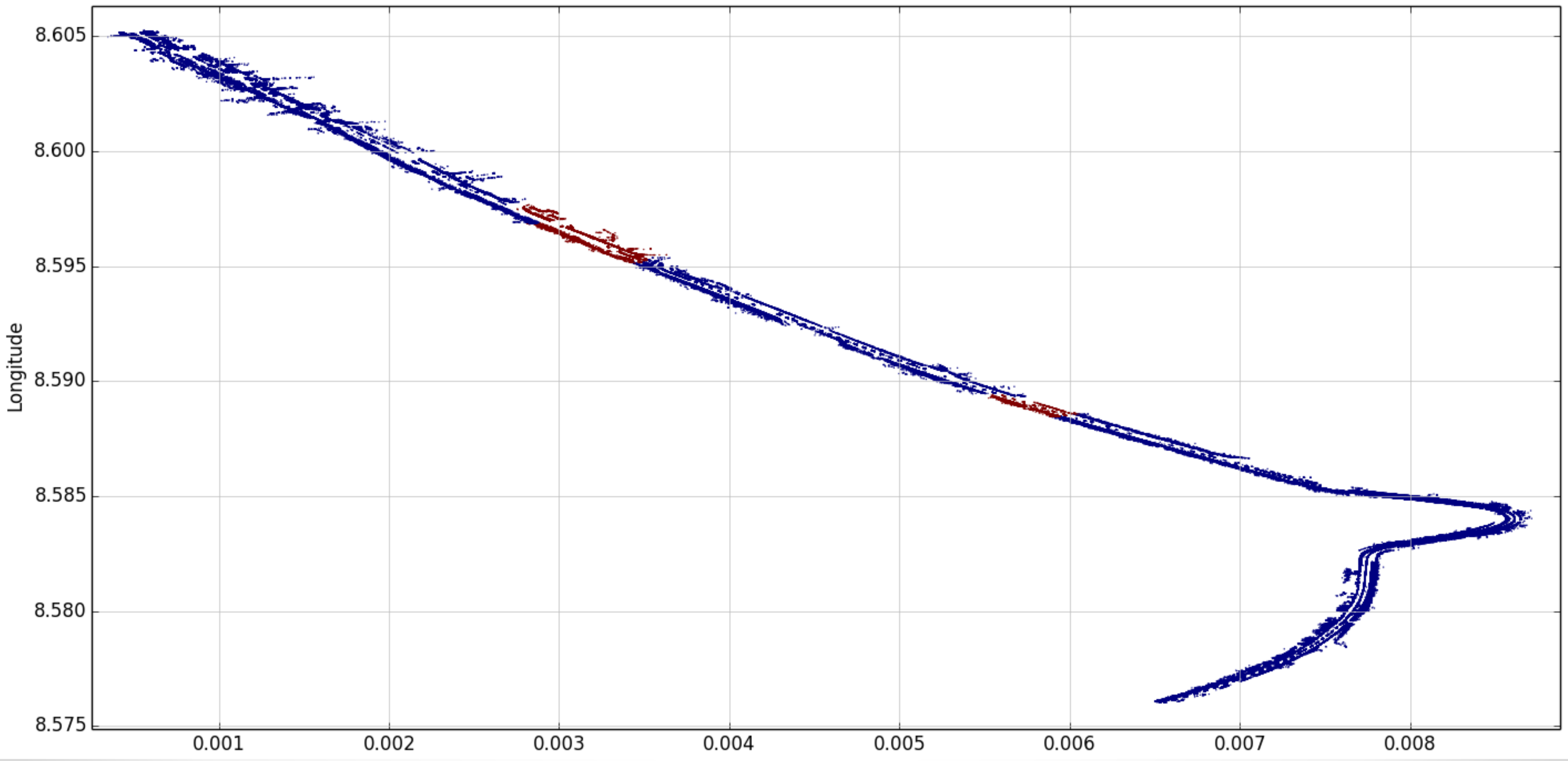
First try: use the Point Cloud Library (PCL) that provides functions of noise removal, but not used to work with it and did not know what it does.

Alternative solution: use the scikit-learn toolkit
- Divide the points in n subsets of points
- For each subset, keep the points as if they were in a plane, run a clustering algorithm
- Keep the largest clusters

How to choose the clustering algorithms: choice amongst K-Means, Affinity Propagation, DBSCAN, Ward, and Mean Shift.
- Want an algorithm that do not take a number of clusters as an input
- Want an algorithm that scale with our amount of data

➔ DBSCAN: Density-based spatial clustering of applications with noise

Noise removal



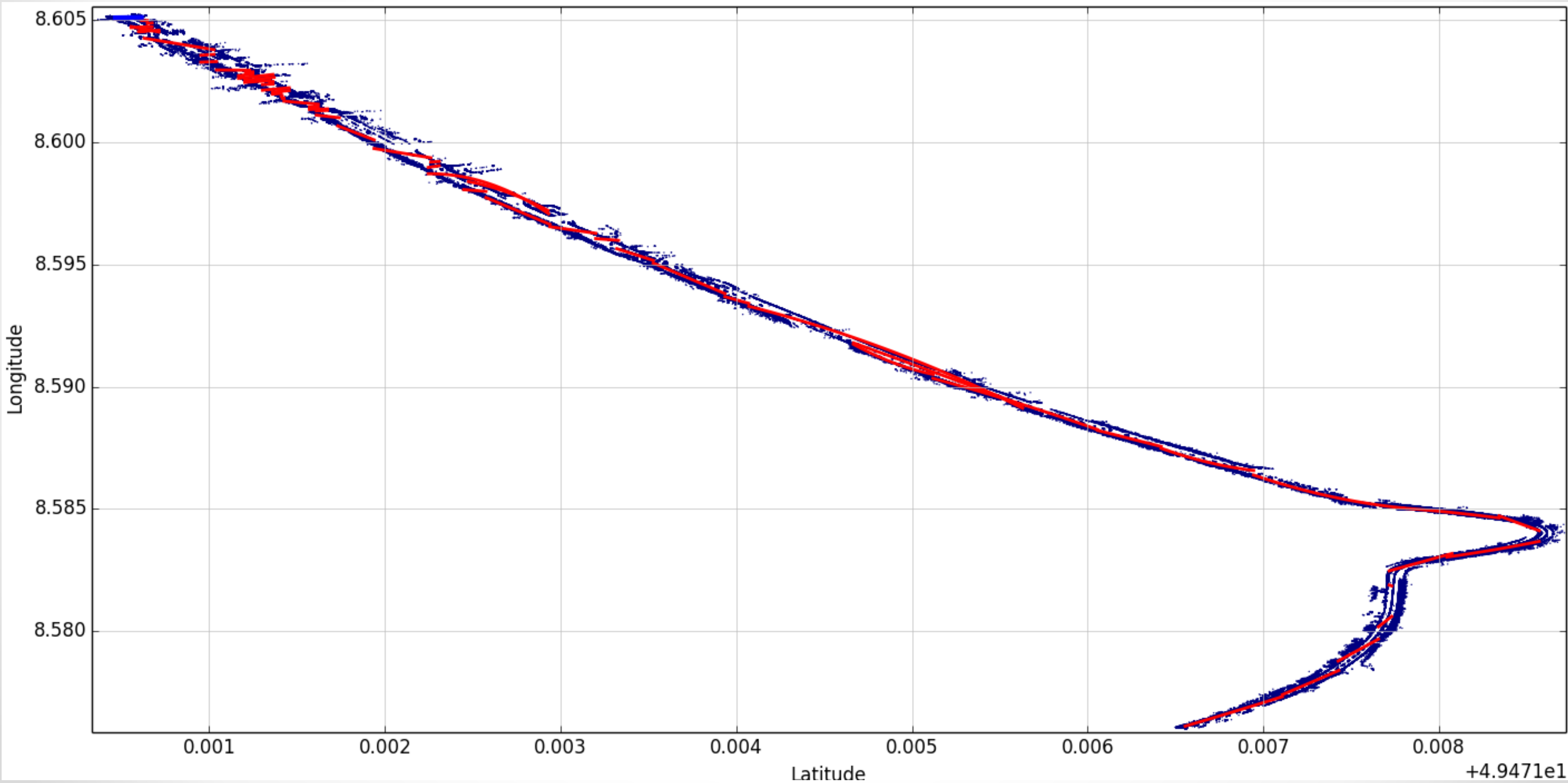
Line fitting

Idea: find functions that describe the curve of the road

Solution:

- Divide the data in n subsets of points
- For each subset, fit the points with a polynom of degree 3 and display the polynom only in the interval of the subset

Line fitting


Longitude
Latitude
+4.9471e1