MASTER
IN CONTROL AND ROBOTICS

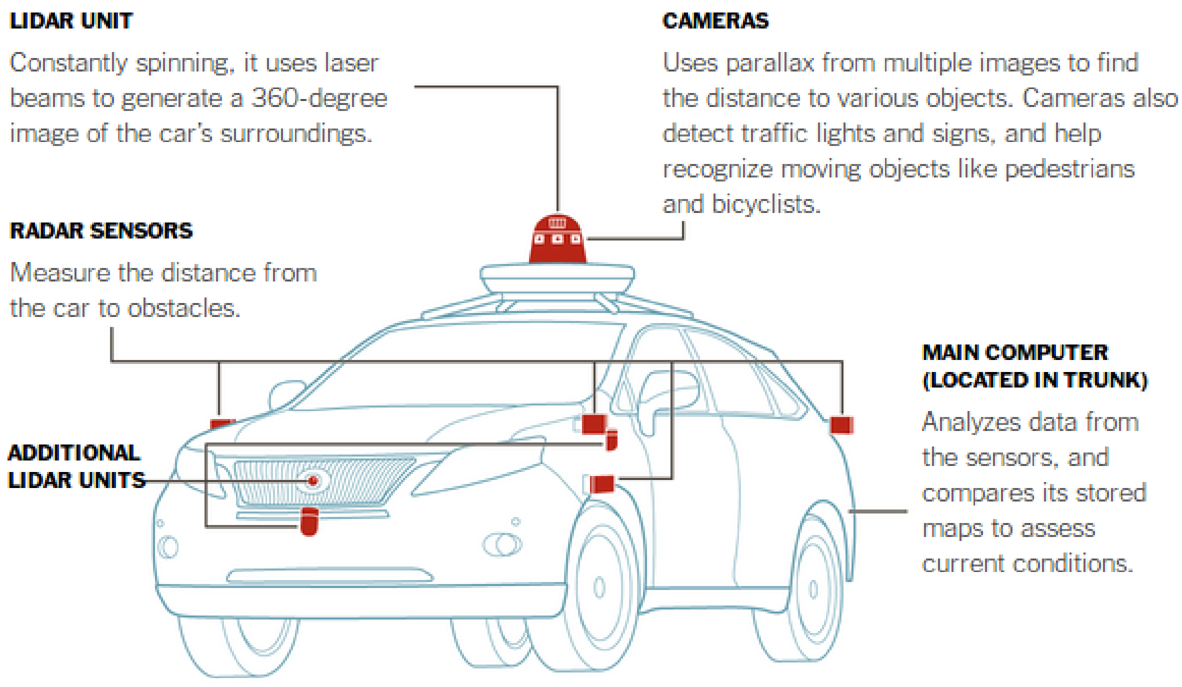
OBSTACLE DETECTION

AUTHORS JINGJING Lei & DURAI Vignesh

SUPERVISORS Arnaud Hamon

June 2018

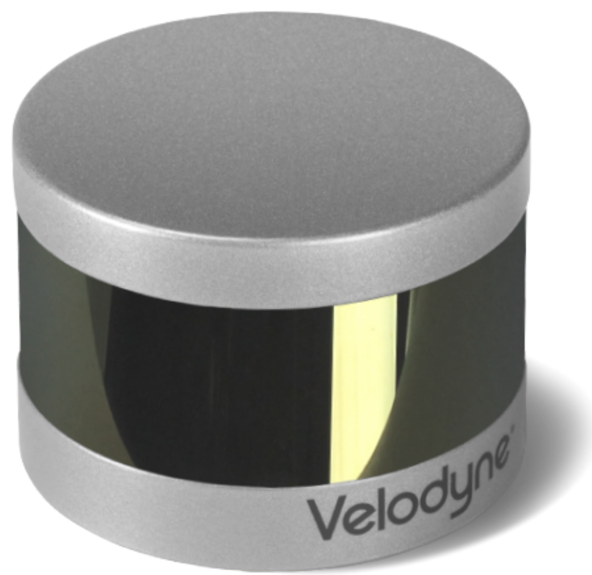
HOW AUTONOMOUS CAR SEE WORLD AROUND THEM



VLP-16 - Velodyne LiDAR

PUCK

VLP-16

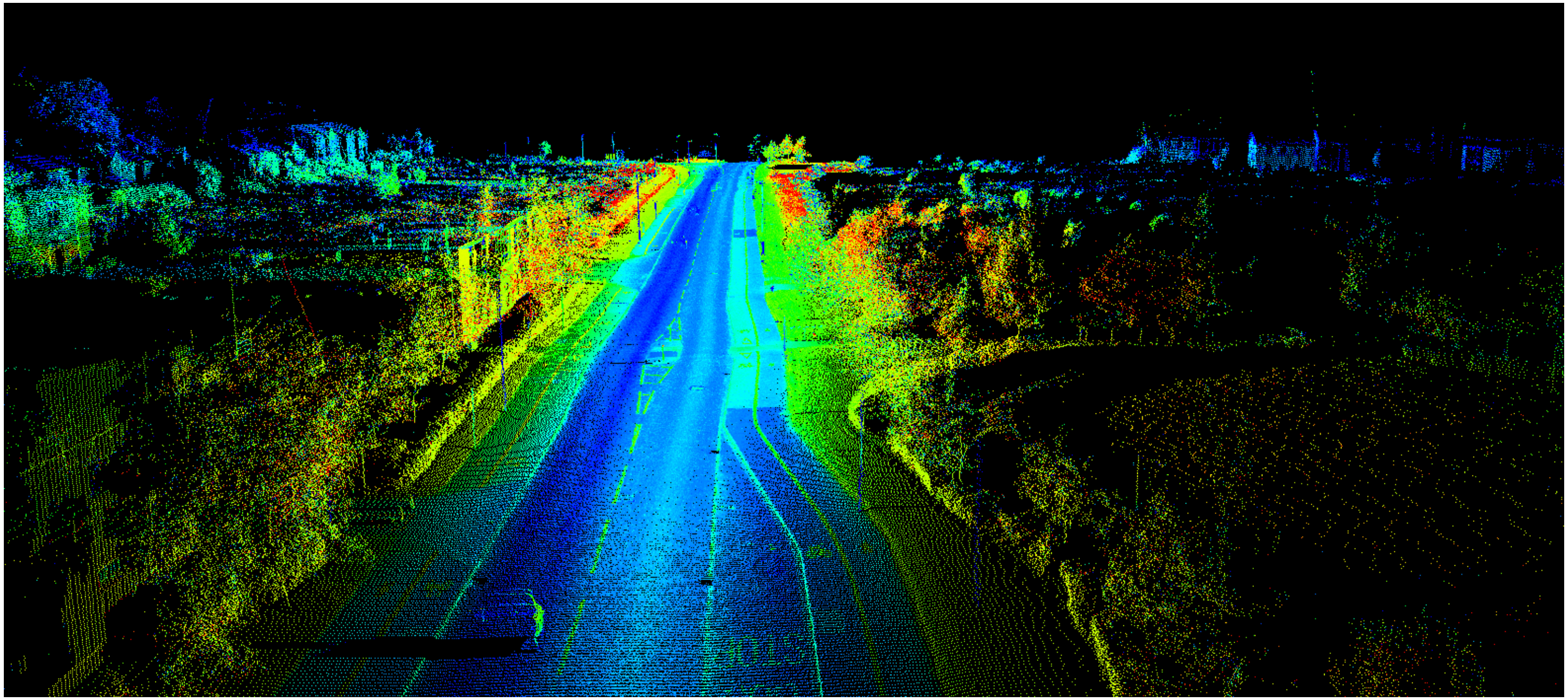


KEY FEATURES

- Dual Returns
- 830 grams
- 16 Channels
- 100m Range
- Up to 600,000 Points per Second
- 360° Horizontal FOV
- ± 15° Vertical FOV
- Low Power Consumption
- Protective Design

Point cloud

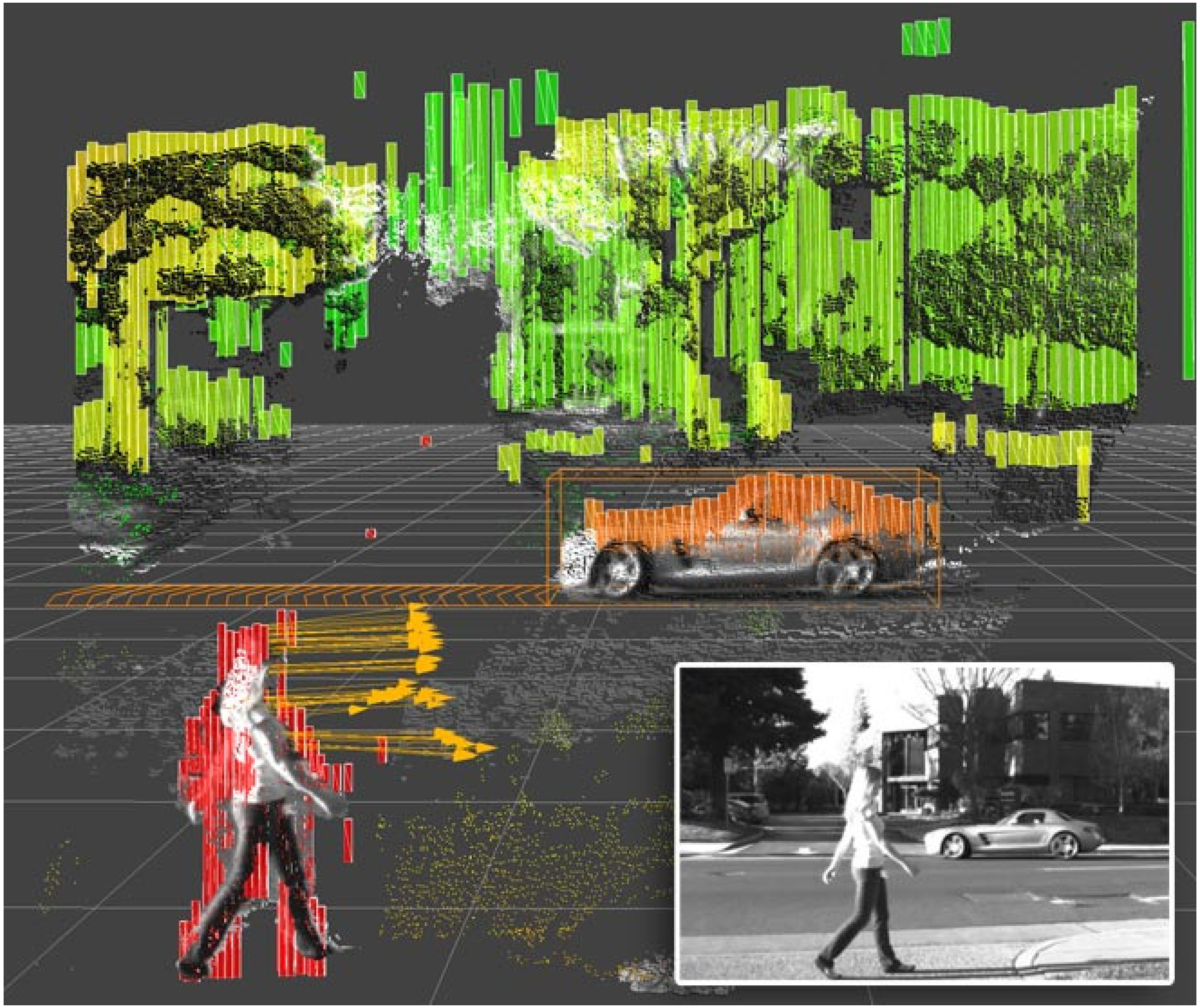
- **Point cloud** are generally produced by 3D scanners, which measure a large number of points on the external surfaces of objects around them



PCL - Point Cloud Library (PCL)

Open-source library of algorithms
for point cloud processing tasks
and 3D geometry processing,
such as occur in three-dimensional computer
vision.

What is our goal



How to achieve our objective

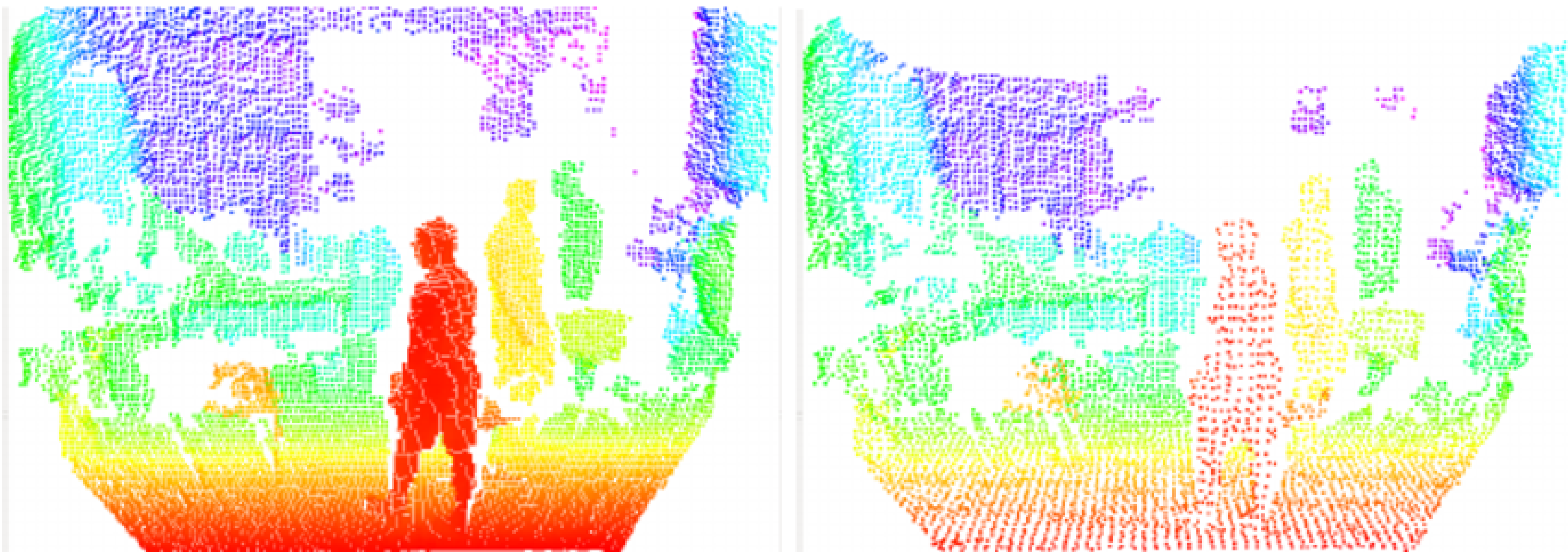
Step 1: Down sampling a Point Cloud using a Voxel Grid filter

Step 2: Selecting the points that are close to the predicted path of the car

Step 3: Clustering of points using DBSCAN algorithm

Step 1: Down sampling a Point Cloud using a Voxel Grid filter

The VoxelGrid class creates a 3D voxel grid over the input point cloud data. Then, in each voxel all the points present will be approximated with their centroid

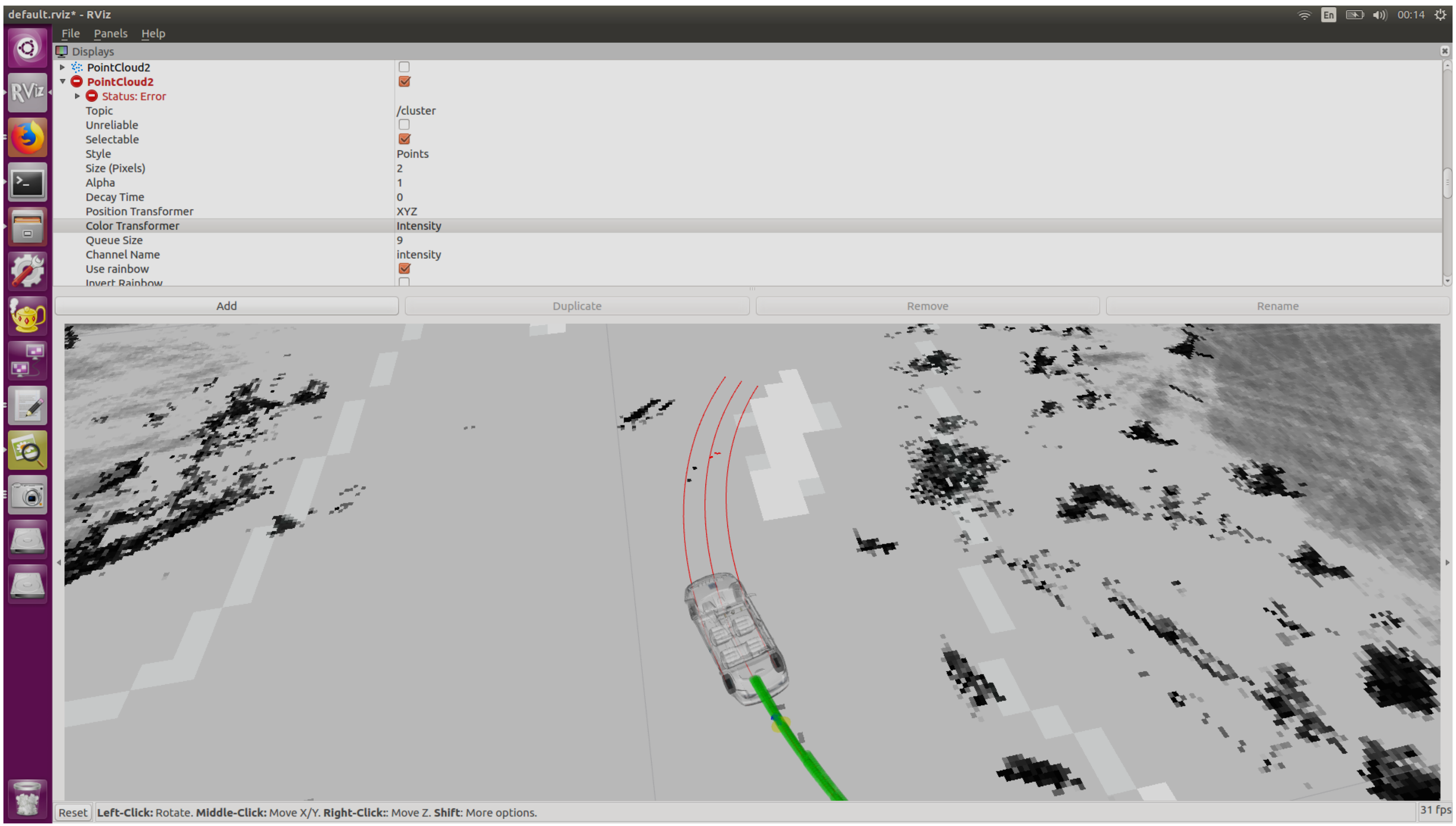


The default filtering values are set to filter data on the z-axis between 0.01 and 1.5 meters, and downsample the data with a leaf size of 0.01 meters.

Step 2: Selecting the points that are close to the path of the car

We find a \texttt{rostopic} called '/FLUENCE/path\textunderscore follower/CurrTraj'.

It contains all the points which are along the predicted driving path

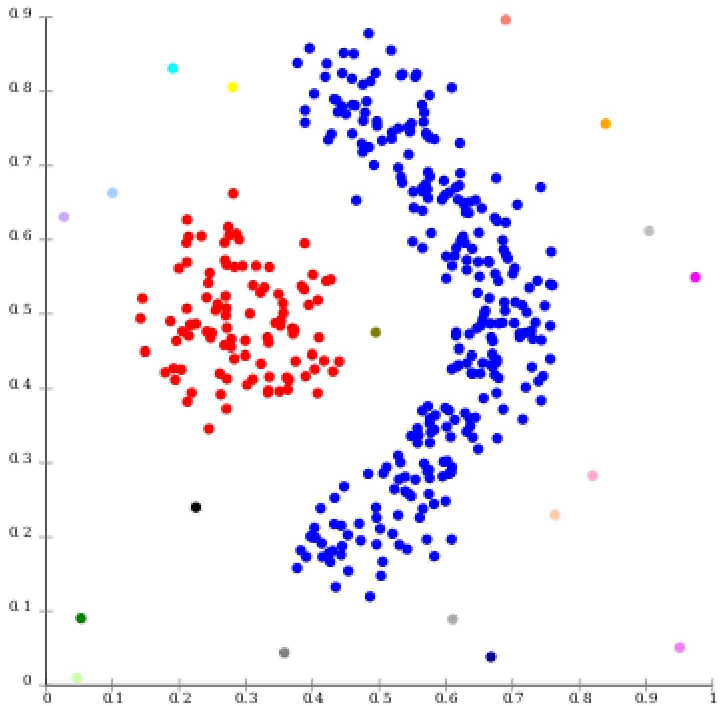


Step 3: Clustering of points using DBSCAN algorithm

WHAT IS CLUSTER ?

Clustering is the task of grouping a set of objects.

One object can be represented by a cluster.



Step 3: Clustering of points using DBSCAN algorithm

HOW WE DO CLUSTERING?

It can be achieved by various algorithms that differ significantly in their understanding of **what constitutes a cluster and how to efficiently find them**.

Step 3: Clustering of points using DBSCAN algorithm

HOW WE DO CLUSTERING?

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold,	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction,	Euclidean distance between points

Step 3: Clustering of points using DBSCAN algorithm

Why DBSCAN

1. Object can be various shape and size

2. Nonfixed cluster number

3. Tuning parameters

Step 3: Clustering of points using DBSCAN algorithm

DBSCAN Algorithm

Step 1: Start with an arbitrary starting point that has not been visited.

Step 2: Extract the neighborhood of this point using epsilon (All points which are within the epsilon distance are neighborhood).

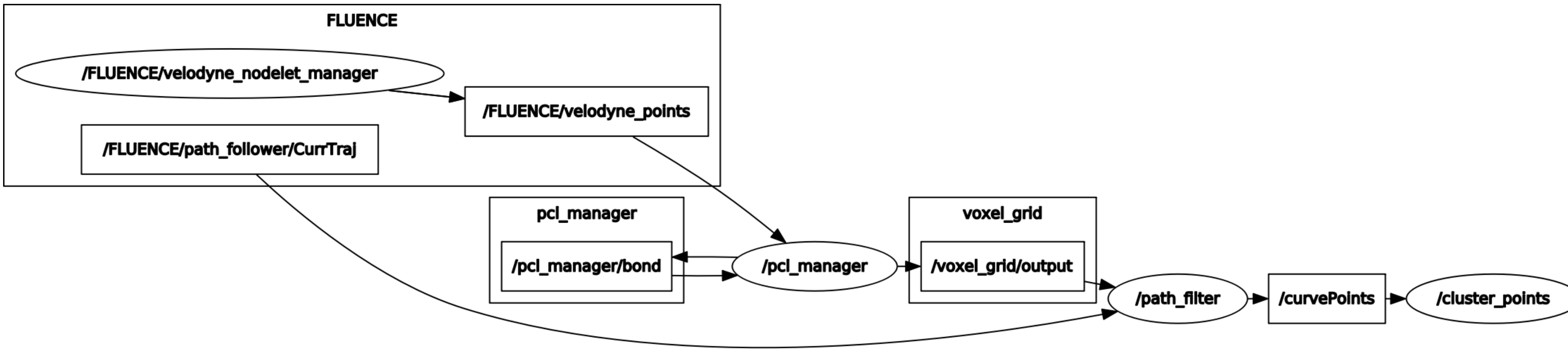
Step 3: If there are sufficient neighborhood around this point then clustering process starts and point is marked as visited else this point is labeled as noise (Later this point can become the part of the cluster).

Step 4: If a point is found to be a part of the cluster then its ε neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all ε neighborhood points. This is repeated until all points in the cluster is determined.

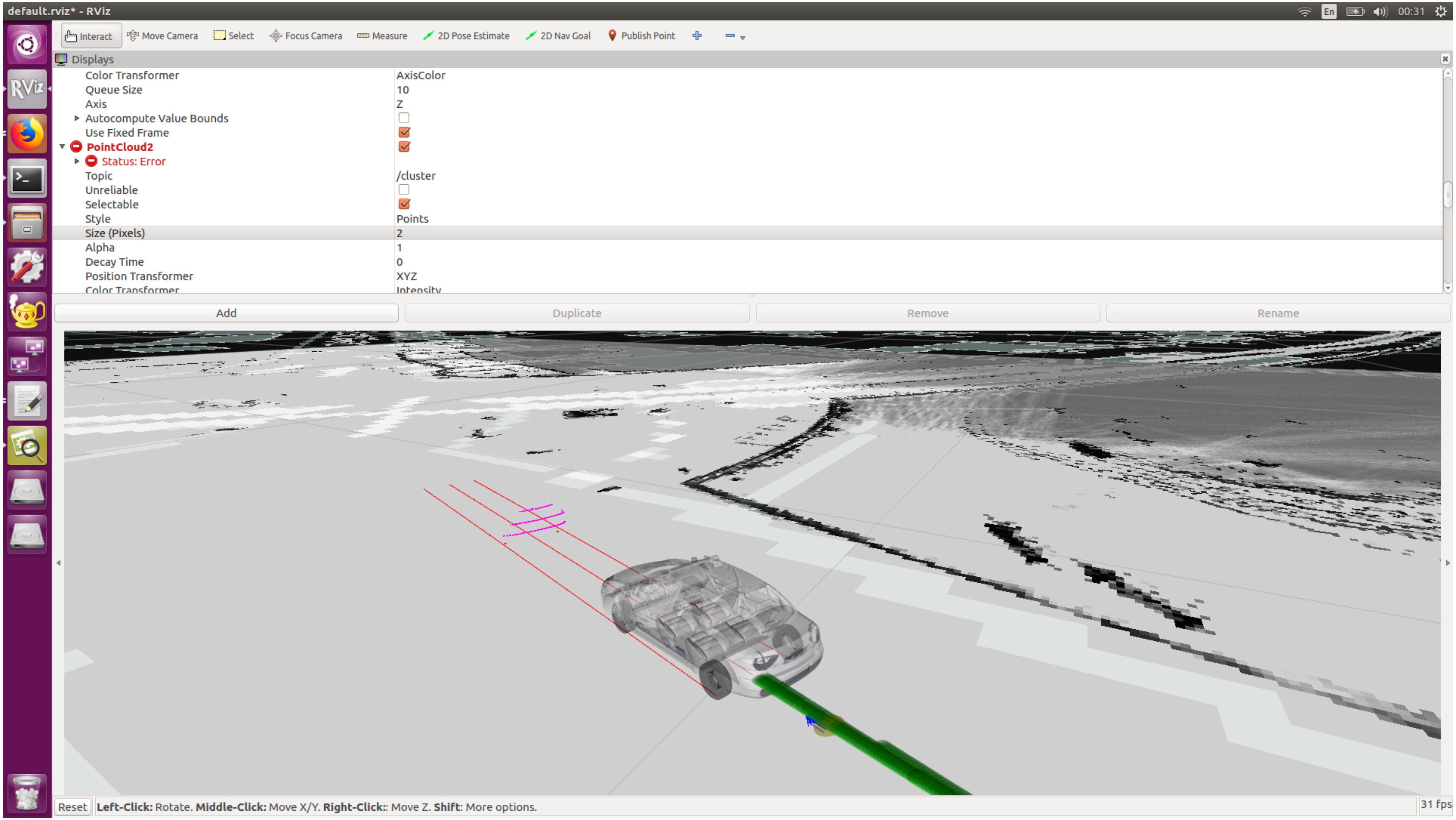
Step 5: A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

Step 6: This process continues until all points are marked as visited.

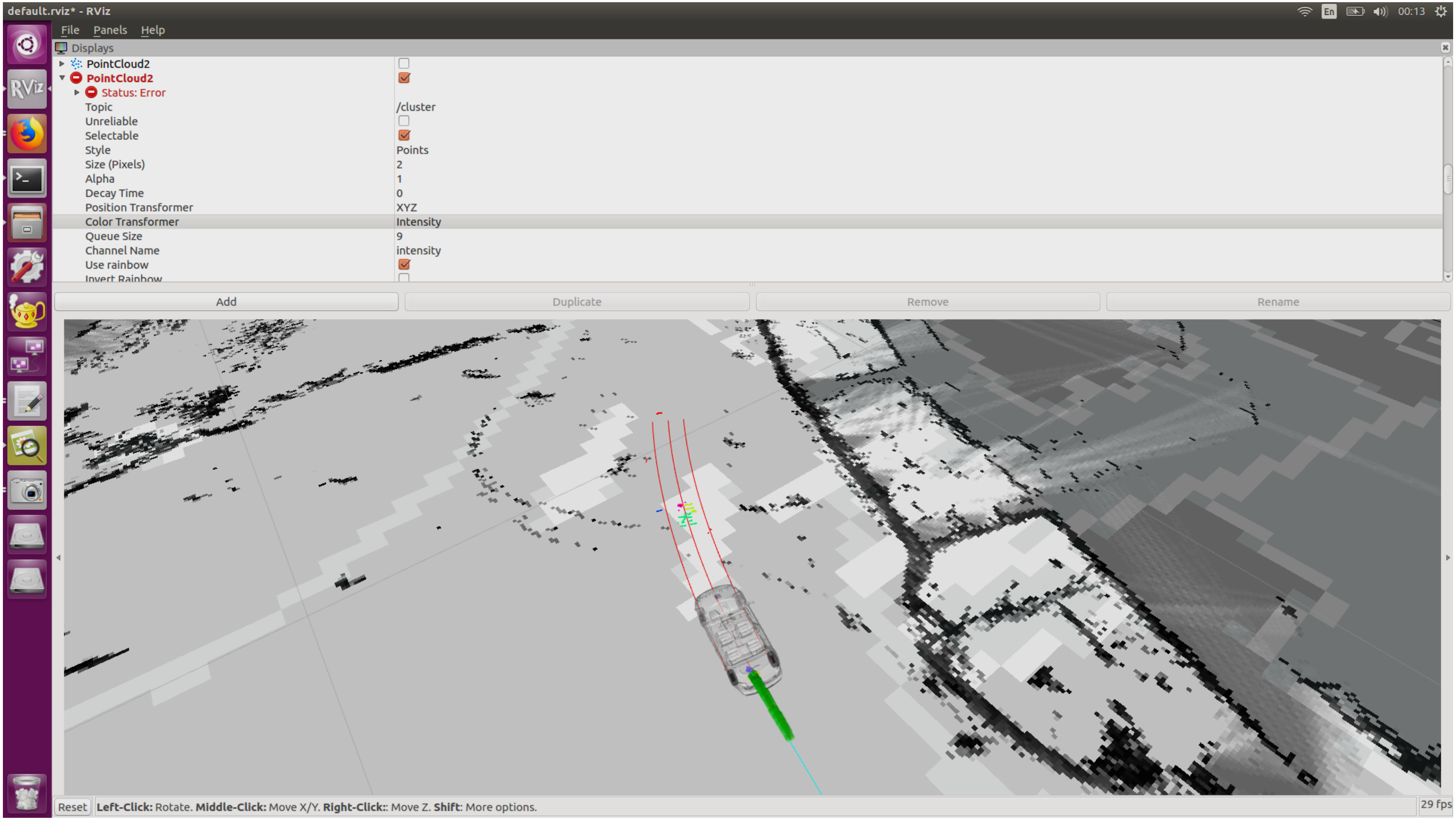
ROS RQT GRAPH



A car is detected



A passengeris detected



Further work

1. Parameter Tuning using OPTICS

2. Overcome the high speed real time challenge

3. Delay (Processing speed)

4 Improve accurency (noise)

Thanks for your attention