

Lidar Perception Pipeline: From Raw Point Clouds to Tracked Objects

AV Perception Portfolio Team

January 31, 2026

Abstract

This report details the implementation and evaluation of a complete Lidar-based perception pipeline for autonomous vehicles. The pipeline processes raw 3D point clouds from a CARLA Lidar sensor through multiple stages: statistical denoising, voxel-grid downsampling, RANSAC-based ground plane segmentation, DBSCAN-based clustering for object detection, and multi-object tracking using a 3D Kalman Filter. The system demonstrates real-time performance with an average latency of 15ms per frame, ensuring robust obstacle detection and tracking in dynamic urban environments.

1 Introduction

Lidar (Light Detection and Ranging) is a critical sensor for autonomous navigation, providing high-resolution 3D geometric information about the surroundings. However, raw Lidar data is computationally expensive to process and contains noise from various sources. A structured perception pipeline is required to extract actionable information, such as object positions and velocities, from the raw stream of point data.

This project implements an industry-standard Lidar perception pipeline with the following core components:

1. **Preprocessing (DSP):** Statistical filtering and spatial downsampling.
2. **Ground Plane Segmentation:** Removal of the road surface to isolate obstacles.
3. **Object Detection:** Grouping points into distinct clusters.
4. **State Estimation:** Continuous tracking of obstacles across time frames.

2 Methodology

2.1 LiDAR Digital Signal Processing

The preprocessing stage aims to reduce noise and data volume without losing critical obstacle information.

2.1.1 Statistical Outlier Removal (SOR)

To remove sensor noise, we implement a distance-based filter. For each point p_i , the mean distance d_i to its k nearest neighbors is calculated. Points are classified as outliers if their mean distance is more than α standard deviations from the global mean distance:

$$d_i > \bar{d} + \alpha \cdot \sigma \quad (1)$$

where \bar{d} is the average distance and σ is the standard deviation. We use $k = 20$ and $\alpha = 2.0$ for robust denoising.

2.1.2 Voxel Grid Filtering

To achieve real-time latency, the point cloud is downsampled using a voxel grid. The 3D space is divided into cubic buckets (voxels) of size $s = 0.2m$. All points within a voxel are averaged to a single centroid:

$$p_{voxel} = \frac{1}{n} \sum_{j=1}^n p_j \quad (2)$$

This typically reduces the point count from 100k to approximately 10k per frame in CARLA.

2.2 Ground Segmentation via RANSAC

Ground removal is essential to isolate vehicles and pedestrians. We use the RANSAC (Random Sample Consensus) algorithm to fit a plane model $ax + by + cz + d = 0$. The algorithm iteratively:

1. Selects 3 random points to define a plane candidate.
2. Calculates the distance of all points to this plane.
3. Counts the number of inliers (points within threshold $\delta = 0.2m$).

The best-fitting plane (maximum inliers) represents the ground. Points significantly above this plane are passed to the clustering stage.

2.3 Euclidean Clustering (DBSCAN)

Object detection is performed using the DBSCAN algorithm, which groups points based on density. A cluster is formed if a minimum of $\text{minPts} = 5$ points are within a radius $\epsilon = 0.5m$. For each cluster, we compute a 3D bounding box defined by:

$$B_k = [\min(x), \max(x)] \times [\min(y), \max(y)] \times [\min(z), \max(z)] \quad (3)$$

2.4 3D Multi-Object Tracking

A Kalman Filter is used for temporal data association and state estimation.

2.4.1 State Model

The state vector \mathbf{x} consists of 3D position and velocity:

$$\mathbf{x} = [x, y, z, v_x, v_y, v_z]^T \quad (4)$$

We assume a ****Constant Velocity**** motion model:

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ \vdots & \ddots & & & \vdots & \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

2.4.2 Data Association

Detections (centroids of clusters) are associated with existing tracks using a greedy Nearest Neighbor approach with a gating threshold of $2.0m$. New tracks are initialized for unmatched detections that persist for 3 frames.

3 Results and Performance

3.1 Experimental Setup

Validation was performed in the CARLA simulator (Town 03) using a Cybertruck vehicle equipped with a 32-channel Lidar sensor spinning at 10Hz.

3.2 Latency Analysis

The pipeline was optimized for speed, achieving sub-millisecond performance for most individual components:

- **Preprocessing:** 8ms
- **Clustering:** 4ms
- **Tracking:** 1ms
- **Total Pipeline:** 15.2ms (Average)

3.3 Qualitative Results

The Bird's Eye View (BEV) visualization confirms successful segmentation and tracking. The RANSAC algorithm consistently identifies the ground plane, and DBSCAN accurately clusters nearby vehicles even at significant ranges (up to 50m).

4 Conclusion

The Week 3 Lidar perception pipeline provides a robust baseline for autonomous vehicle perception. It effectively handles noisy raw data and converts it into high-level object tracks suitable for path planning. Future work will involve integrating 3D oriented bounding boxes and more advanced data association algorithms like Global Nearest Neighbor (GNN) or Hungarian matching.