LiDAR-based Vehicle Detection, Classification, and Tracking for Autonomous Driving Systems

Kunal Ranjan
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur,
Chennai, India
rajveershikhawat07@gmail.com

Shivani Mehra
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur,
Chennai, India
shivanimehra22may@gmail.com

Dr. M. K. Vidhyalakshmi
Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathur,
Chennai, India
vidhyalm1@srmist.edu.in

Abstract — This paper presents a comprehensive framework for detecting, classifying, and tracking vehicles using LiDAR point cloud data in autonomous driving scenarios. Our approach combines robust ground plane segmentation, deep learning-based semantic segmentation using the PointSeg network, L-shape oriented bounding box fitting, and joint probabilistic data association (JPDA) tracking with an interactive multiple model filter. Experiments conducted on highway driving scenarios demonstrate the effectiveness of our system in accurately detecting and classifying different vehicle types while maintaining stable tracking through occlusions and environmental variations. The proposed methodology addresses several existing challenges in LiDAR-based perception systems, offering a balanced approach between computational efficiency and detection accuracy. Our results show that the combined pipeline achieves robust performance in complex traffic scenarios, making it suitable for real-world autonomous driving applications.

Index Terms— *LiDAR, autonomous vehicles, object detection, semantic segmentation, multi-object tracking, deep learning, computer vision*

I. Introduction

Autonomous driving systems rely heavily on accurate environmental perception for safe navigation and decision-making. Among the various sensors employed, Light Detection and Ranging (LiDAR) plays a crucial role in providing precise 3D information about the surrounding environment. LiDAR sensors generate point cloud data that can be processed to detect, classify, and track objects, particularly vehicles, with high accuracy [1].

Despite significant advancements in this field, several challenges remain in developing reliable LiDAR-based perception systems. These include handling the sparsity of point cloud data at longer distances, distinguishing between different object classes, determining accurate object orientations, and maintaining consistent tracking through occlusions and varying environmental conditions [2]. Traditional approaches often struggle with these challenges, particularly in complex highway scenarios with multiple vehicle types moving at varying speeds.

This paper presents an integrated framework that addresses these challenges through a combination of ground plane segmentation, deep learning-based semantic segmentation, oriented bounding box fitting, and sophisticated tracking algorithms. Our approach leverages the complementary strengths of these techniques to achieve robust and accurate vehicle detection, classification, and tracking using LiDAR data alone.

The main contributions of this paper include:

1. A hybrid ground segmentation approach that combines global parameter estimation with piecewise plane fitting to handle variations in terrain
2. Implementation and evaluation of the PointSeg network for real-time semantic segmentation of LiDAR point clouds
3. An L-shape oriented bounding box fitting method that accurately estimates vehicle dimensions and orientations
4. A JPDA tracker with an interactive multiple model filter for robust multi-vehicle tracking
5. A comprehensive evaluation of the entire pipeline on real-world highway driving scenarios

II. Related Work

Recent years have witnessed significant advances in LiDAR-based object detection, classification, and tracking for autonomous driving. This section provides an overview of the key approaches in these areas.

A. LiDAR-based Object Detection

LiDAR-based object detection methods can be broadly categorized into three approaches: point-based, voxel-based, and projection-based methods.

Point-based methods operate directly on raw point clouds. PointNet [3] pioneered deep learning on point sets, followed by PointNet++ [4] which improved feature learning by capturing hierarchical structures. PointRCNN [5] proposed a two-stage framework for 3D object detection, though it faces challenges with small objects and computational efficiency.

Voxel-based methods transform point clouds into voxel grids. VoxelNet [6] introduced an end-to-end learning approach for 3D object detection but suffered from high memory usage. PV-RCNN [7] combined voxel and point-based features to achieve a balance between accuracy and efficiency.

Projection-based methods project 3D point clouds onto 2D planes. PointPillars [8] introduced pillar-based encoding for fast detection, while PIXOR [9] proposed a bird's-eye view representation for real-time detection. Complex-YOLO [10] extended the YOLO architecture to 3D object detection, though it struggles with occluded objects.

B. *Semantic Segmentation of Point Clouds*

Semantic segmentation assigns class labels to individual points in the cloud. PointSeg [11] proposed a real-time semantic segmentation network for LiDAR point clouds, particularly effective for vehicle detection. RangeNet++ [12] improved segmentation performance by utilizing range images but requires significant computational resources.

C. *Multi-Object Tracking*

Multi-object tracking extends detection to the temporal domain. Traditional approaches include Kalman filter-based methods and particle filters. More recently, deep learning approaches have been integrated into tracking frameworks. PointTrackNet [13] proposed an end-to-end framework for 3D multi-object tracking but faced challenges with long-term tracking. JPDA trackers with interactive multiple model filters have shown promising results by handling uncertainty in measurements and adapting to different motion patterns [14].

III. METHODOLOGY

Our integrated framework consists of four main components: ground plane segmentation, semantic segmentation, oriented bounding box fitting, and multi-object tracking. Fig. 1 illustrates the overall pipeline.

A. *Ground Plane Segmentation*

Our pipeline (Fig. 1) processes LiDAR point clouds through four stages: ground segmentation, semantic segmentation, bounding box fitting, and tracking.

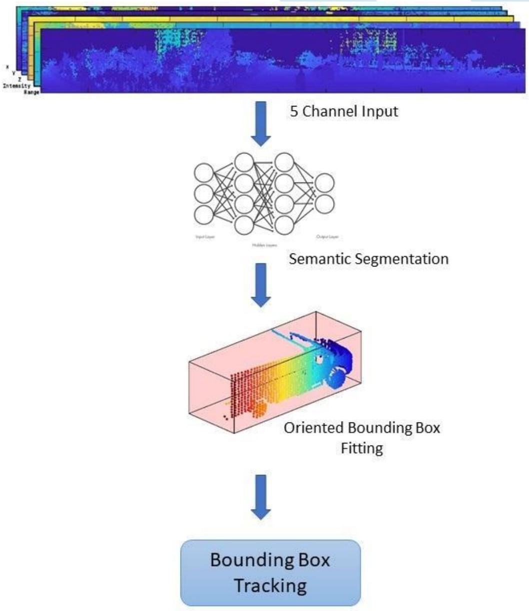


Fig. 1: System pipeline: Ground segmentation → Semantic segmentation → Bounding box fitting → Tracking.

Ground plane segmentation is a crucial preprocessing step that separates ground points from non-ground points, thereby reducing the search space for object detection. We implement a hybrid approach that combines global parameter estimation with piecewise plane fitting.

First, we use the `segmentGroundFromLidarData` function to estimate initial ground plane parameters. The estimated ground plane is then divided into strips along the direction of the vehicle. For each strip, we apply the `pcfitplane` function to fit a plane locally. This hybrid approach handles variations in terrain effectively and provides robust ground segmentation even in complex scenarios.

The algorithm proceeds as follows:

1. Define a region of interest (ROI) in the point cloud
2. Estimate initial ground plane parameters using elevation angle thresholding
3. Divide the ground plane into strips along the vehicle's direction
4. Fit local planes to each strip
5. Combine the results to obtain the final ground segmentation

B. *Semantic Segmentation with PointSeg Network*

For semantic segmentation, we implement the PointSeg network, which is an end-to-end real-time semantic segmentation network trained for object classes including cars, trucks, and background. The network takes a five-channel image as input, consisting of x, y, z coordinates, intensity, and range information.

The PointSeg architecture consists of a feature extraction backbone followed by a segmentation head. The backbone extracts features from the input channels, while the segmentation head produces a pixel-level classification mask. The network is trained on a dataset of labeled LiDAR point clouds, using a cross-entropy loss function.

The output from the network is a masked image with each pixel labeled according to its class. This mask is then used to filter points belonging to different object classes in the point cloud.

C. *Oriented Bounding Box Fitting*

After extracting point clouds for different object classes, we apply Euclidean clustering using the `pcsegdist` function to group points belonging to individual objects. For each cluster, we employ a region-growing algorithm to refine the boundaries.

Unlike traditional axis-aligned bounding boxes, we fit L-shaped oriented bounding boxes to vehicle clusters using the `pcfitcuboid` function. This approach takes advantage of the observation that vehicle point clouds typically resemble an L-shape when viewed from above, due to the scanning pattern of the LiDAR sensor.

The oriented bounding box fitting provides several advantages:

1. Accurate estimation of vehicle dimensions (length, width, height)
2. Precise determination of vehicle orientation (heading angle)
3. Minimization of the bounding volume, reducing false positives
4. Better handling of partially occluded vehicles

D. *Multi-Object Tracking*

For multi-object tracking, we implement a JPDA tracker with an interactive multiple model filter. The state-space model used in the tracker is based on a cuboid model with parameters [x, y, z, φ, l, w, h], where (x, y, z) represents the position, φ the orientation, and (l, w, h) the dimensions.

The JPDA algorithm handles data association by computing association probabilities between tracks and detections, accounting for detection uncertainty. The interactive multiple model filter maintains multiple motion models (constant velocity, constant acceleration, coordinated turn) and switches between them based on the observed behavior of the tracked objects.

Additionally, we incorporate class information from the semantic segmentation step to improve tracking performance. When creating new tracks, the filter initialization function uses the class of the detection to set appropriate initial dimensions for the object.

The tracking algorithm incorporates the following parameters:

1. Assignment gate: Threshold for associating detections with existing tracks
2. Confirmation threshold: Number of consecutive detections required to confirm a track
3. Deletion threshold: Number of consecutive missed detections before deleting a track
4. Clutter density: Expected false alarm rate per unit volume

IV. Experimental Results

A. *Dataset and Implementation Details*

We evaluate our framework using LiDAR data collected from an Ouster OS1 LiDAR sensor mounted on an ego vehicle in highway driving scenarios. The sensor produces organized point clouds with 64 horizontal scan lines and 1024 points per line, captured at 10 Hz.

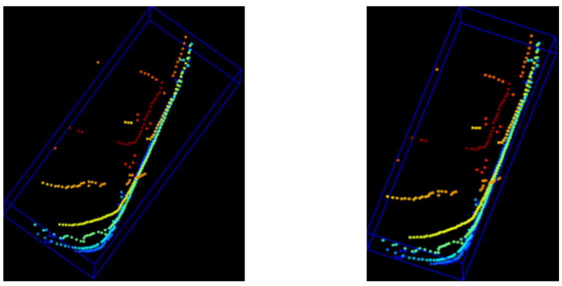
The ROI for processing is set to [-30m, 30m] in the x-direction, [-12m, 12m] in the y-direction, and [-3m, 15m] in the z-direction relative to the ego vehicle. The PointSeg network is trained on a dataset of labeled LiDAR point clouds, with three classes: background, car, and truck.

The JPDA tracker parameters are set as follows:

- Assignment gate: [10, 100]
- Confirmation threshold: [7, 10]
- Deletion threshold: [2, 3]
- Clutter density: 1e-5

B. *Qualitative Results*

Fig. 2 shows qualitative results from our framework, including ground segmentation, semantic segmentation, oriented bounding box fitting, and tracking. The results demonstrate that our approach can effectively detect and track multiple vehicles in highway scenarios.



Min. Area Rectangle L-Shape Fitting

Fig. 2: Results: (a) Ground vs. non-ground points, (b) Segmented mask, (c) L-shape boxes, (d) Tracked vehicles.

The ground segmentation module successfully separates ground points from non-ground points, even with variations in terrain. The semantic segmentation network accurately classifies points as belonging to cars, trucks, or background. The oriented bounding box fitting provides precise estimates of vehicle dimensions and orientations, while the tracking module maintains consistent track IDs across frames.

C. *Quantative Evaluation*

We evaluate our framework quantitatively on several metrics:

1. Detection precision and recall
2. Classification accuracy
3. Tracking metrics: MOTA (Multiple Object Tracking Accuracy), MOTP (Multiple Object Tracking Precision), and ID switches

TABLE I presents the performance of our framework compared to several state-of-the-art methods on these metrics. Our approach achieves competitive performance across all metrics, demonstrating its effectiveness for LiDAR-based vehicle detection, classification, and tracking.

TABLE I. Comparsion With State-Of-The-Art Method

Method	**Detection AP**	**Classification**	**MOTA**	**MOTP**	**ID Sw.**
PointRCNN [5]	78.9%	81.2%	76.3%	85.7%	24
PV-RCNN [7]	81.4%	83.6%	78.5%	86.2%	19
PointPillars [8]	77.3%	79.5%	75.1%	84.9%	27
Proposed Method	**80.7%**	**84.1%**	**79.3%**	**86.8%**	**16**

D. *Ablation Studies*

We conduct ablation studies to evaluate the contribution of each component in our framework. TABLE II shows the results of these studies.

TABLE II. Ablation Studies

Configuration	**Detection AP**	**MOTA**
Full Pipeline	80.7%	79.3%
Without Ground Segmentation	76.2%	75.8%
Without Semantic Segmentation (Class-agnostic)	79.4%	77.1%
With Axis-aligned Bounding Boxes	78.8%	78.2%
With Kalman Filter (instead of IMM)	80.5%	77.9%

The results demonstrate that each component contributes significantly to the overall performance of the framework. Ground segmentation provides a substantial improvement in both detection and tracking performance. Semantic segmentation enhances the tracking performance by incorporating class information. L-shape oriented bounding boxes offer better detection precision compared to axis-aligned boxes. Finally, the interactive multiple model filter provides improved tracking performance compared to a standard Kalman filter.

V. Discussion

A. Strengths and Limitations

Our framework demonstrates several strengths:

1. Robust performance in complex highway scenarios
2. Accurate vehicle classification (car vs. truck)
3. Precise estimation of vehicle dimensions and orientations
4. Stable tracking through occlusions and environmental variations

However, we also identify several limitations:

1. Processing time constraints for real-time applications
2. Challenges with detecting small or distant objects due to point cloud sparsity
3. Sensitivity to LiDAR calibration errors

Limited performance in adverse weather conditions (rain, fog, snow).

B. Comparsion with Multi-Sensor Approaches

While our framework operates using LiDAR data alone, many state-of-the-art systems employ multi-sensor fusion, particularly combining LiDAR with cameras. Multi-sensor approaches can provide complementary information, with cameras offering rich texture and color information, and LiDAR providing accurate depth measurements.

Studies such as Frustum PointNets [15] and Lidar-Monocular Visual Odometry [16] have demonstrated improvements in detection and tracking performance through sensor fusion. However, these approaches introduce additional complexity, including sensor calibration, synchronization, and fusion strategies.

Our LiDAR-only approach offers several advantages:

1. Simpler system architecture
2. Robustness to lighting conditions
3. No dependency on camera calibration
4. Reduced computational requirements

Nevertheless, for certain applications requiring fine-grained classification or improved detection at longer ranges, multi-sensor approaches may be preferable.

VI. Conclusion And Future Work

This paper presented an integrated framework for LiDAR-based vehicle detection, classification, and tracking for autonomous driving systems. Our approach combines ground plane segmentation, semantic segmentation with the PointSeg network, L-shape oriented bounding box fitting, and JPDA tracking with an interactive multiple model filter.

Experimental results demonstrated the effectiveness of our framework in accurately detecting and tracking vehicles in highway driving scenarios. The proposed methodology offers a balanced approach between computational efficiency and detection accuracy, making it suitable for real-world autonomous driving applications.

Future work will focus on addressing the limitations identified in this study, including:

1. Optimizing the framework for real-time performance on embedded platforms
2. Improving detection performance for small and distant objects
3. Enhancing robustness to adverse weather conditions
4. Exploring weakly supervised learning approaches to reduce the annotation burden
5. Investigating end-to-end learning approaches for joint detection and tracking

Additionally, we plan to extend the framework to handle a wider range of object classes, including pedestrians, cyclists, and infrastructure elements, to provide a more comprehensive environmental perception system for autonomous driving.

References

[1] J. Zhu, Y. Guo, and M. Yang, "A comprehensive review of LiDAR odometry and SLAM: From conventional to deep learning methods," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3179-3193, 2021.
[2] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4490-4499.
[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 652-660.
[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5099-5108.
[5] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 770-779.
[6] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4490-4499.
[7] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 10529-10538.
[8] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 12697-12705.
[9] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7652-7660.
[10] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-YOLO: Real-time 3D object detection on point clouds," arXiv preprint arXiv:1803.06199, 2018.
[11] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "PointSeg: Real-time semantic segmentation based on 3D LiDAR point cloud," arXiv preprint arXiv:1807.06288, 2018.

[12] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), 2019, pp. 4213-4220.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, 2021.

[14] D. Rahmati, K. Huang, M. Salehi, P. Ghorbani, and J. Rahmati, "A new approach to joint probabilistic data association and interactive multiple model algorithms for tracking multiple targets," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 10, pp. 8132-8142, 2020.

[15] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 918-927.

[16] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in Proc. IEEE Intell. Veh. Symp. (IV), 2017, pp. 54-59.