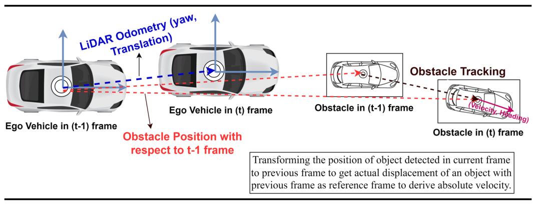
Sensor applications

L3D-OTVE: LiDAR-Based 3-D Object Tracking and Velocity Estimation Using LiDAR Odometry

Abhishek Thakur* and P Rajalakshmi**

Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285, India
*Graduate Student Member, IEEE
**Senior Member, IEEE

Manuscript received 14 May 2024; revised 11 June 2024; accepted 15 June 2024. Date of publication 19 June 2024; date of current version 27 June 2024.



Transforming the position of object detected in current frame to previous frame to get actual displacement of an object with previous frame as reference frame to derive absolute velocity.

Abstract—Object tracking and velocity estimation are essential aspects of autonomous vehicles (AVs). AVs take actions based on tracking and estimating the velocity of obstacles moving around. Estimating the absolute velocity of obstacles poses challenges for optical sensors, such as LiDAR or cameras, in the absence of a global reference frame. Not much work has been done on estimating obstacle velocity using a LiDAR sensor. This letter proposes a novel method for tracking the velocity of obstacles using a LiDAR sensor, which involves transforming the current detection onto the previous axes of the LiDAR frame using LiDAR odometry, considering parameters, such as rotation and translation, between two consecutive frames. The estimated velocity is evaluated against ground truth data from the TiAND dataset. This dataset employs global navigation satellite system (GNSS) sensors equipped on a car acting as an obstacle, providing precise velocity measurements. The proposed algorithm demonstrates a root-mean-square deviation (RMSD) of 0.13 m/s, showcasing its accuracy in velocity estimation. The algorithm successfully tracks high-speed obstacles, reaching up to 100 km/h in highway data from the TiAND dataset and has also been tested on the KITTI dataset in real-time.

Index Terms—Sensor applications, autonomous vehicle (AV), LiDAR, object tracking, odometry, velocity estimation.

I. INTRODUCTION

LiDAR is one of the key sensors for autonomous vehicles (AVs). When an ego vehicle equipped with a LiDAR sensor is in motion, estimating the velocity of an object becomes challenging. Radar is used to track the velocity of the obstacles using the Doppler shift of the returned radio waves [1]. However, this capability is a limitation of the LiDAR sensor. First, an object is detected through a 3D-point cloud segmentation and clustering process [2]. Most LiDAR-based object tracking algorithms rely on deep learning, such as CenterPoint tracking [3]. However, these algorithms are computationally expensive and unsuitable for real-time applications in self-driving cars. In general, the relative velocity of an obstacle can be estimated by calculating the difference in the distance of an object in consecutive frames and dividing it by the time taken between the two consecutive frames. However, determining the absolute velocity becomes challenging when the ego-vehicle equipped with LiDAR is in motion in the absence of a reference frame. Subsequently, the absolute velocity can be determined by transforming the position of the obstacle detected in the current frame in the previous frame using LiDAR odometry. The absolute velocity of an obstacle is estimated by tracking and measuring its displacement from the common frame at the last timestamp of LiDAR data. The distance of an object measured in two consecutive frames is detected by a LiDAR sensor equipped on a moving ego-vehicle. Consequently, the relative velocity of the object can be estimated, taking into account that the LiDAR sensor is also in motion. The object's position detected in the current frame needs to be transformed to the previous frame using LiDAR odometry to derive the absolute velocity, as detailed in the methodology and illustrated in Fig. 1. Most of the work discusses object tracking using LiDAR. There is a lack of research focused on tracking the velocity of objects with a moving LiDAR. The major contributions of this work are as follows.

1) This letter introduces a real-time method that tracks obstacles using displacement thresholds and Intersection over Union (IOU) matching. A connection matrix facilitates unique identity (ID) assignments for effective tracking.
2) A novel LiDAR-based approach for obstacle velocity tracking is proposed. This method involves transforming current detections onto the previous LiDAR frame's axes using LiDAR odometry, considering rotation and translation between two consecutive frames.
3) The estimated velocity is compared with the ground truth in our TiAND dataset [5], which is the velocity measured by the global navigation satellite system (GNSS) sensor with real-time kinematic (RTK) equipped in the car as an obstacle moving in front. The estimated velocity has a root-mean-square deviation (RMSD) of 0.13 m/s.
4) The algorithm is validated by successfully tracking high-speed obstacles, reaching up to 100 km/h, on highway data in the TiAND dataset and tested on the KITTI dataset in real-time.

Corresponding author: Abhishek Thakur (e-mail: ee20resch11014@iith.ac.in).
Associate Editor: Francisco Falcone.
This work was supported by Prime Minister Research Fellowship (PMRF) and the Technology Innovation Hub on Autonomous Navigation (TiHAN) Foundation.
Code is available online at https://github.com/abhishekt711/L3D-OTVE.
Digital Object Identifier 10.1109/LSENS.2024.3416411

II. RELATED WORKS

The object tracking begins with the detection of obstacles. Anand et al. [2] discussed point cloud segmentation techniques to

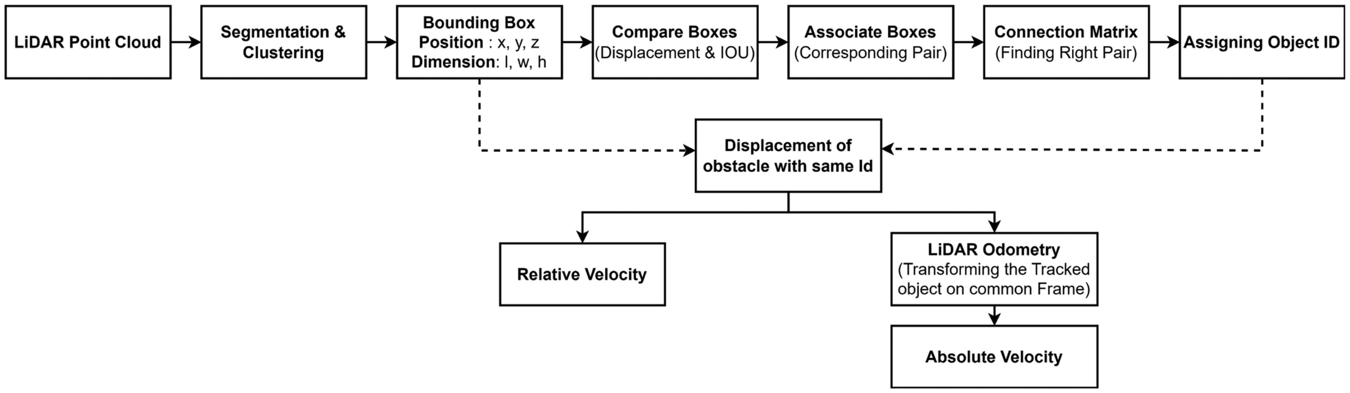

LiDAR Point Cloud
Segmentation & Clustering
Bounding Box Position : x, y, z Dimension: l, w, h
Compare Boxes (Displacement & IOU)
Associate Boxes (Corresponding Pair)
Connection Matrix (Finding Right Pair)
Assigning Object ID
Displacement of obstacle with same Id
Relative Velocity
LiDAR Odometry (Transforming the Tracked object on common Frame)
Absolute Velocity

Fig. 1. Block diagram of our algorithm (L3D-OTVE), LiDAR-based 3-D object tracking and velocity estimation using LiDAR odometry.

filter ground points reflected from the road surface. The authors in [3] and [4] first obtained 3-D detections from a LiDAR point cloud and then the combination of a 3-D Kalman filter, and the Hungarian algorithm is used for state estimation and data association. Peng et al. [6] discussed a 3-D object tracking method with the assistance of vision to improve the performance of object association. Feng et al. [7] track objects based on factor graph optimization, simultaneously tracking multiple objects, and reconstructing their models. Zhang et al. [8] used stationary roadside LiDAR as traffic flow monitoring sensors for accurate vehicle counting and speed estimation. However, this method cannot estimate the absolute speed with the LiDAR mounted on a moving AV. The authors in [9], [10], and [11] provide an efficient method for segmenting moving objects in dynamic environments using LiDAR. Most research conducted in LiDAR object tracking primarily concentrates on the tracking of objects [12]. However, there is a scarcity of studies specifically addressing the tracking of object velocity using a mobile LiDAR sensor.

III. METHODOLOGY

A. Object Tracking

The object tracking involves the following steps.

1) Point Cloud Clustering: The LiDAR point cloud segmentation is performed to filter out ground points and points reflected from obstacles using the PCL library in the robot operating system (ROS). Through the clustering technique, the obstacle is enclosed, and the coordinates of the centroid, along with the length, width, and height of the bounding box, are determined.

2) Compare Boxes: The Compare function compares the calculated displacement and dimension similarity values against the provided thresholds. It computes the displacement between the positions of two boxes and measures the similarity in dimensions (length, width, and height) of the two boxes along each axis (*x, y, and z*) in two continuous frame. Let suppose M obstacles are detected in frame t denoted as L_t and N obstacles are detected in previous frame $(t-1)$ denotes as L_{t-1}, here $i \in (0, 1, \ldots, M)$ and $j \in (0, 1, \ldots, N)$. The displacement and the IOU matching matrix between each obstacle are calculated as follows:

$$d_{ij} = \begin{bmatrix} d_{11} & \cdots & d_{1N} \\ \cdots & \cdots & \cdots \\ d_{M1} & \cdots & d_{MN} \end{bmatrix}; I_{ij} = \begin{bmatrix} I_{11} & \ldots & I_{1N} \\ \cdots & \cdots & \cdots \\ I_{M1} & \cdots & I_{MN} \end{bmatrix}. \quad (1)$$

3) Associate Boxes: The associate boxes function creates pairs of indices between boxes in two consecutive frames that are deemed similar based on the criteria defined in the compare boxes function. It iterates through each box in the previous and the current frame. If the compare boxes function returns true for a pair of boxes (one from the previous frame and one from the current frame), indicating their similarity based on displacement and dimension similarity thresholds, it adds a pair of indices (previous box ID, current box ID) to the connection pairs vector. The associate boxes function represents pairs of indices indicating similar boxes between consecutive frames.

4) Connection Matrix: Connection Matrix $\mathbb{C}^{MxN}$ is initialized and filled with zeros. If there is a connection between the boxes of objects in consecutive frames, then the value at that pair is set to 1

$$C_{ij} = \begin{pmatrix} C_{11} & \ldots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{M1} & \ldots & C_{MN} \end{pmatrix}. \quad (2)$$

Initialize $C_{ij} = [0]$, if $d_{ij} < d\text{th}$ and $I_{ij} < I\text{th}$; then, $C_{ij} = 1$. The pair of box IDs representing potential matches between the previous and current frames are found by iterating through connection pairs. It extracts unique box IDs from each pair and stores them in left and right vectors, representing boxes from the previous and current frames, respectively. It iterates through connection pairs again, finding the indices of the corresponding box IDs in the left and right vectors

$$L_i = [L_1, \ldots, L_i, \ldots, L_M] \text{ and } R_i = [R_1, \ldots, R_j, \ldots, R_N] \quad (3)$$

5) Assigning IDs: A recursive function is implemented to explore potential assignments in a connection matrix. A valid assignment is discovered when $C_{i,j} = 1$ and $C[:\, j] \rightarrow \text{False}$. Mark $C[:\, j] \rightarrow \text{True}$ to prevent revisiting this column. Recursively checks for conflicts or potential assignments for the current assignment of j. If no conflicts are found, assign i to j in the right pair vector. The function outputs the number of matches found between the current and previous frames. The final list of objects in the current frame is given as

$$O = [L_{i1}, L_{i2}, \ldots R_{j1}, Rj2, \ldots]. \quad (4)$$

Here, $(L_{i1}, L_{i2}, \ldots)$ are the index of objects, which are common in both frame and $R_{j1}, Rj2, \ldots$ are the new objects found in current LiDAR frame. For each box in the vector, it checks if the ID of the box matches the provided ID. If a box with a matching ID is found, it returns the index of that box within the vector. A new ID is assigned for a new object as shown in Fig. 2.

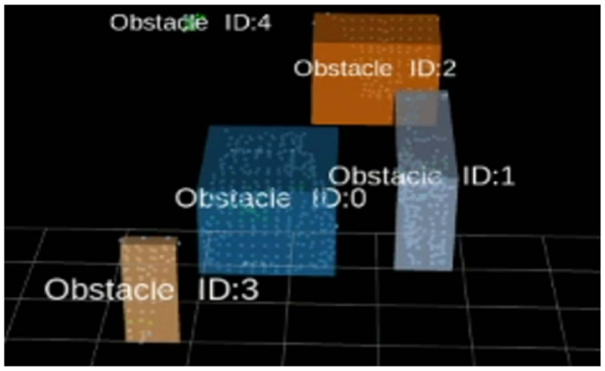


Fig. 2. Objects are assigned an ID number using our approach for object tracking. The ground points are segmented and separated from the point cloud data of LiVOX HAP LiDAR.

B. Velocity Estimation

As the obstacle moves in the road in the x–y plane, we have considered yaw rotation only between two consecutive LiDAR frames. After matching the previous frame to current frame using LiDAR odometry and mapping [13], rotation and translation of the point cloud frame is determined with respect to the previous frame. Let $\gamma_{(t-1)\to t}$ be the yaw rotation, $L_{x,(t-1)\to t}$, and $L_{y,(t-1)\to t}$ be the displacement in the x and y directions from the previous frame to the current frame. The position of the centroid of an obstacle Id N is given as $[C_{x,t-1}, C_{y,t-1}]$ in $(t-1)$-frame and $[C_{x,t}, C_{y,t}]$ in t-frame. The position of the obstacle ID K in frame t is transformed to frame $(t-1)$ using a transformation matrix, which contains rotational and translational parameters as follows:

$$\begin{bmatrix} C_x \\ C_y \\ 1 \end{bmatrix}_{t\to t-1} = \begin{bmatrix} \cos\gamma_{(t-1)\to t} & -\sin\gamma_{(t-1)\to t} & L_{x,(t-1)\to t} \\ \sin\gamma_{(t-1)\to t} & \cos\gamma_{(t-1)\to t} & L_{y,(t-1)\to t} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ 1 \end{bmatrix}_t . \tag{5}$$

From (5), The position of obstacle detection in current frame with respect to the previous frame in x and y directions can be given as

$$C_{x,(t)\to t-1} = \cos\gamma_{(t-1)\to t} C_{x,t} - \sin\gamma_{(t-1)\to t} C_{y,t} + L_{x,(t-1)\to t} \tag{6}$$

$$C_{y,(t)\to t-1} = \cos\gamma_{(t-1)\to t} C_{x,t} - \sin\gamma_{(t-1)\to t} C_{y,t} + L_{y,(t-1)\to t} . \tag{7}$$

Now, we have the obstacle position detected in two continuous frame on a common frame $(t-1)$, so the velocity can be calculated in the x and y directions as

$$V_x = \frac{C_{x,t\to t-1} - C_{x,t-1}}{T} ; V_y = \frac{C_{y,t\to t-1} - C_{y,t-1}}{T} . \tag{8}$$

Here, T is the difference between the time stamps of two continuous frames of LiDAR point cloud data. The speed and heading (δ) of an obstacle can be given as

$$|V| = \sqrt{|V_x|^2 + |V_y|^2} ; \delta = \tan^{-1}\left(\frac{V_y}{V_x}\right) . \tag{9}$$

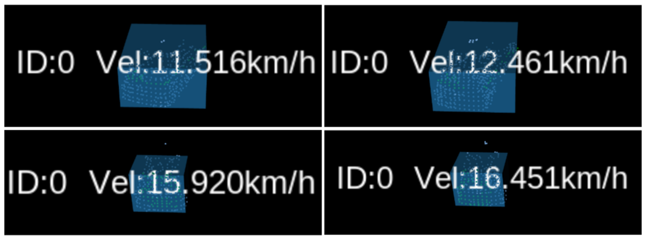
IV. RESULTS

The algorithms are executed within the ROS Noetic on Ubuntu 20. The experiments are conducted on an Intel i7-1165G7 processor running at 4.7 GHz, equipped with 16 GB RAM. The velocity tracking performance is evaluated on our TiAND Dataset [5]. In this velocity tracking data, the ground truth is obtained using the NovAtel PwrPak7 GNSS sensor with RTK connection, providing accuracy at the 1 cm level of measurement. The GNSS sensor is equipped on a car acting as an obstacle moving ahead of the data-collection vehicle. The data are gathered at the Technology Innovation Hub on Autonomous Navigation testbed facility [14], which is specifically designed for testing autonomous navigation technologies for both unmanned ground vehicles and unmanned aerial vehicles. The velocity tracking algorithm demonstrated robust performance, as presented in Table 1. Out of 1049 frames, 985 frames (93.89%) exhibited velocity tracking within a deviation range of 0.5 km/h (0.138 m/s). The plot for velocity tracked by our algorithm is compared with the ground truth data as shown in Fig. 3(b), and the tracked obstacle is shown in Fig. 3(a). The RMSD in velocity compared with GNSS ground truth was 0.13 m/s.

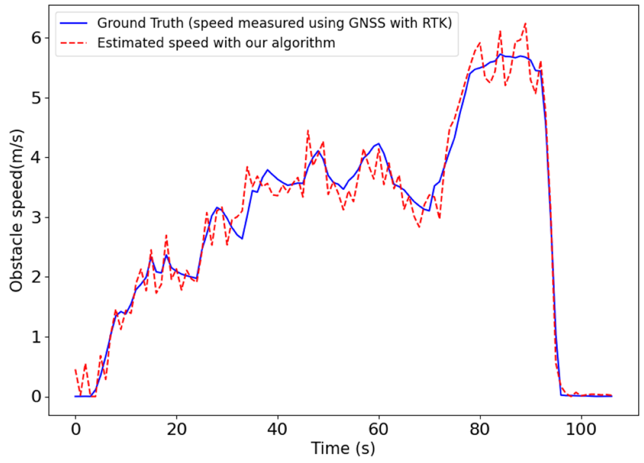
Furthermore, using our TiAND dataset collected on a national highway with a 128-channel Velodyne LiDAR, we assess whether our algorithm can effectively track and estimate the velocity of obstacles traveling at very high speeds, up to 100 km/h is tracked in real-time, as illustrated in Fig. 4. Also, the algorithm is tested on the KITTI dataset as shown in Fig. 5. Our algorithm works efficiently in real-time on all datasets taken from different types of LiDAR. Table 2 compares

Table 1. Performance Metrics for Velocity Tracking on Our TiAND Dataset, Where the Obstacle is Equipped With GNSS With RTK Connection to Provide Ground Truth Velocity

Total Number of Frame	Velocity tracked within the deviation of 0.5 km/h (0.138m/s)	Accuracy within the deviation of 0.5 km/h (0.138m/s)	RMSD Compared with GNSS ground truth
1049	985 frame	93.89%	0.13



(a)



(b)

Fig. 3. Velocity of the moving object is tracked using our algorithm and compared with the ground truth velocity measured by GNSS with RTK connection mounted in the tracked vehicle. (a) Obstacle with ID 0 tracked in different frames. (b) Comparison of speed estimates from our algorithm with ground truth data.

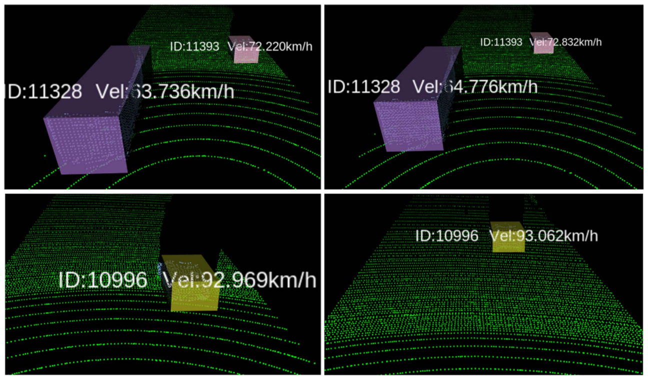


Fig. 4. High-speed moving objects up to 100 km/h are tracked in real-time in the TiAND data collected on the highway.

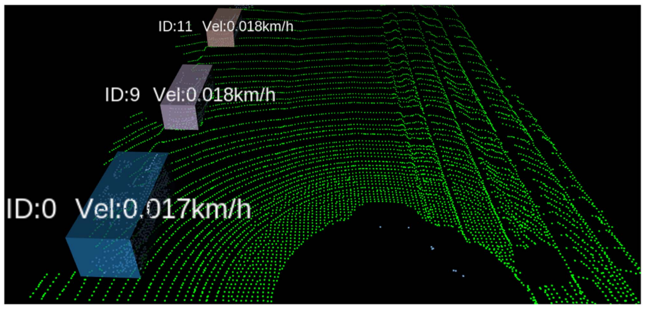


Fig. 5. Object tracking and velocity estimation in real-time on the Kitti dataset with parked cars.

Table 2. Average LiDAR Processing Time Breakdown for Point Cloud Segmentation, Clustering, Filtering, and Odometry for Velocity Tracking

LiDAR	Segmentation, Filtering	LiDAR Odometry Velocity Tracking	Total Time
Kitti Dataset (Velodyne 64 Channel)	21 ms	32 ms	53 ms
TiAND Highway Data (Velodyne 128 Channel)	35 ms	47 ms	82 ms
TiAND Data (Livox HAP LiDAR)	7 ms	13 ms	20 ms

the average processing times (in milliseconds) for various LiDAR operations, including point cloud segmentation, clustering, filtering, and LiDAR odometry for velocity tracking. Results are presented for different LiDAR sensors on the KITTI dataset (Velodyne 64 Channel), TiAND Highway Data (Velodyne 128 Channel), and TiAND solid-state LiDAR data using the LiVOX HAP sensor. The algorithm operates in real-time as the point cloud is downsampled and filtered in front of the vehicle, and the odometry is computed between two consecutive frames only. The total time for velocity tracking is less than the LiDAR data acquisition rate of 100 ms per frame at 10 Hz.

V. CONCLUSION

The novel approach for object velocity tracking proposed in this work has shown promising and robust results during the evaluation using our TiAND dataset. the algorithm achieved a RMSD in velocity, measured against GNSS ground truth of 0.13 m/s. The algorithm's effectiveness was further validated on a national highway, showcasing its capability in tracking and estimating velocities, especially for high-speed objects moving up to 100 km/h in real-time. The comparison of processing times underscores the algorithm's reliability and suitability for real-world applications in dynamic scenarios. In the future, further exploration of sensor fusion techniques can improve tracking for very high-speed objects by integrating IMU, camera, and LiDAR for enhanced robustness. In addition, this work can be extended to track and predict the future trajectory of moving objects, enabling efficient path planning for autonomous vehicles.

ACKNOWLEDGMENT

This work was supported in part by the Prime Minister Research Fellowship (PMRF) and in part by the Indian Institute of Technology Hyderabad through the Technology Innovation Hub on Autonomous Navigation (TiHAN) foundation.

REFERENCES

[1] B. Ubezio, C. Schöffmann, L. Wohlhart, S. Mülbacher-Karrer, H. Zangl, and M. Hofbaur, "Radar based target tracking and classification for efficient robot speed control in fenceless environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 799–806, doi: 10.1109/IROS51168.2021.9636170.
[2] B. Anand, M. Senapati, V. Barsaiyan, and P. Rajalakshmi, "LiDAR-INS/GNSS-based real-time ground removal, segmentation, and georeferencing framework for smart transportation," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 8504611, doi: 10.1109/TIM.2021.3117661.
[3] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021 pp. 11779–11788, doi: 10.1109/CVPR46437.2021.01161.
[4] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10359–10366, doi: 10.1109/IROS45743.2020.9341164.
[5] TiHAN-IIT Hyderabad, "TIAND: TiHAN IITH autonomous navigation dataset," 2023. [Online]. Available: https://tihan.iith.ac.in/tiand
[6] Z. Peng, Z. Xiong, Y. Zhao, and L. Zhang, "3-D objects detection and tracking using solid-state LiDAR and RGB camera," *IEEE Sensors J.*, vol. 23, no. 13, pp. 14795–14808, Jul. 2023, doi: 10.1109/JSEN.2023.3279500.
[7] S. Feng et al., "Accurate and real-time 3D-LiDAR multiobject tracking using factor graph optimization," *IEEE Sensors J.*, vol. 24, no. 2, pp. 1760–1771, Jan. 2024, doi: 10.1109/JSEN.2023.3336221.
[8] J. Zhang et al., "Vehicle tracking and speed estimation from roadside LiDAR," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5597–5608, 2020, doi: 10.1109/JSTARS.2020.3024921.
[9] N. Wang et al., "InsMOS: Instance-aware moving object segmentation in LiDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 7598–7605, doi: 10.1109/IROS55552.2023.10342277.
[10] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021, doi: 10.1109/LRA.2021.3093567.
[11] B. Mersch et al., "Receding moving object segmentation in 3D LiDAR data using sparse 4D convolutions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7503–7510, Jul. 2022, doi: 10.1109/LRA.2022.3183245.
[12] C. Wisultschew et al., "3D-LiDAR based object detection and tracking on the edge of IoT for railway level crossing," *IEEE Access*, vol. 9, pp. 35718–35729, 2021, doi: 10.1109/ACCESS.2021.3062220.
[13] A. Thakur, B. Anand, H. Verma, and P. Rajalakshmi, "Real time LiDAR odometry and mapping and creation of vector map," in *Proc. 8th Int. Conf. Autom. Robot. Appl.*, 2022, pp. 181–185, doi: 10.1109/ICARA55094.2022.9738576.
[14] TiHAN-IIT Hyderabad, "TiHAN: TiHAN IITH autonomous navigation testbed," 2020. [Online]. Available: https://tihan.iith.ac.in/tihan-iith-testbed