

Low-Cost Camera and 2-D LIDAR Fusion for Target Vehicle Corner Detection and Tracking: Applications to Micromobility Devices

Hamidreza Alai ^a and Rajesh Rajamani ^{a *}

^a Department of Mechanical Engineering, University of Minnesota, Twin Cities, MN 55455, USA

Key Words: sensor fusion, electric scooters, collision prediction, vehicle detection, state estimation, vehicle tracking

* Corresponding author, rajamani@umn.edu, tel: 612-626-7961, 111 Church Street SE, Minneapolis, MN 55455, USA.

Nomenclature

A	: acceleration of the vehicle
$f(\cdot)$	: nonlinear function for vehicle dynamics in companion form
f_c	: focal length of the camera
$\tilde{f}(\cdot)$	: estimation error for nonlinear function of vehicle dynamics in companion form
γ_1, γ_2	: Lipschitz constants for the vehicle nonlinear functions in companion form
H	: output matrix
l	: wheelbase length of vehicle
l_f	: vehicle front length
l_r	: vehicle rear length
L	: observer (High-) gain matrix
$\lambda_{max}(\cdot)$	: maximum eigenvalue of a positive definite matrix
$p_{x,0}$	: horizontal principal offset of the camera
$p_{x,i}$	: horizontal location of a pixel in the image
P	: positive definite matrix in Lyapunov function
Q	: transformed observer gain matrix in the LMI problem
$T(\theta)$	: observer transformation matrix
T_B	: block matrix used in observer transformation matrix
$ TTC_{x,y}$	: longitudinal or lateral time-to-collision values
V	: speed of vehicle
x	: states of system
X	: relative longitudinal position of vehicle from e-scooter
y	: output vector of the system
Y	: relative lateral position of vehicle from e-scooter
β	: vehicle slip angle
δ_F	: steering angle of vehicle
φ_i	: corresponding azimuth angle of pixel $p_{x,i}$
ψ	: yaw angle of vehicle
z	: transformed states of system
$\hat{z}$	: estimated transformed states of system
$\tilde{z}$	: state estimate error vector

Abstract

This paper develops a cost-effective vehicle detection and tracking system based on fusion of a 2-D LIDAR and a monocular camera to protect electric micromobility devices, especially e-scooters, by predicting the real-time danger of a car-scooter collision. The cost and size disadvantages of 3-D LIDAR sensors make them an unsuitable choice for micromobility devices. Therefore, a 2-D RPLIDAR Mapper sensor is used. Although low-cost, this sensor comes with major shortcomings such as the narrow vertical field of view and its low density of data points. Due to these factors, the sensor does not have a robust output in outdoor applications, and the measurements keep jumping and sliding on the vehicle surface. To improve the performance of the LIDAR, a single monocular camera is fused with the LIDAR data not only to detect vehicles, but also to separately detect the front and side of a target vehicle and to find its corner. It is shown that this corner detection method is more accurate than strategies that are only based on the LIDAR data. The corner measurements are used in a high-gain observer to estimate the location, velocity, and orientation of the target vehicle. The developed system is implemented on a Ninebot e-scooter platform, and multiple experiments are performed to evaluate the performance of the algorithm.

1. Introduction

Scooters not only provide personal pleasure to many commuters but also serve as an alternative to ride-sharing and public transit, especially as a last-mile commute option. They also provide a full commute option for workers who live within a few miles of their workplace. Popularity of e-scooter riding has increased significantly over the last few years, as it is an inexpensive and easily available mode of transportation. It is predicted that the e-scooter market will grow from US\$14 billion in 2014 to \$37 billion in 2024 [1]. While e-scooters provide convenient benefits, e-scooter riders constitute a vulnerable population on roads. Research shows that the number of severe injuries for e-scooter riders is increasing [2]. E-scooter riders are in significant danger of suffering from severe injuries and fatalities in any car-scooter collision. While relatively few deaths have occurred so far on e-scooters, most of them have resulted from a collision with a motor vehicle [3]. A more comprehensive data analysis of hospital data by the US Consumer Product Safety Commission [2] found that emergency room visits in the US from scooter-related injuries surged 450% from 7,700 in 2017 to 42,200 in 2021, with these estimates likely being an undercount [4].

The goal in this project is to develop a vehicle tracking system on e-scooters capable of predicting potential car-scooter crashes, based on low-cost LIDAR and camera fusion. If the danger of a car-scooter crash is predicted, a loud audio alert is sounded to warn the vehicle driver about the presence of the e-scooter. In [5], a vehicle tracking system that was based on a single beam laser sensor was developed to track vehicles in specific scenarios. This system was restricted to tracking rear vehicles behind the e-scooter. A Rutgers University team has explored a rear-approaching vehicle detection system for use on bicycles using computer vision techniques [6]. Garmin has also developed a rear-collision prevention system for bicycles using radar [7]. Researchers have also explored a laser-sensor-based bicycle protection system [8]. The sensor systems explored so far for bicycles have been simplistic and have not used nonlinear observers and sensor fusion for tracking complex maneuvers, unlike the system being studied here for e-scooters.

To have a cost-effective vehicle tracking system suitable for e-scooters, an inexpensive 2-D LIDAR needs to be used instead of a 3-D LIDAR, to measure the distances to vehicles. The measurement data from the 2-D LIDAR has very low-density, especially when it comes to vehicles that are not close to the sensor. Even for a 3D LIDAR, papers in the literature have discussed that it is hard to detect and track objects far from the ego vehicle due to the sparsity issue. A recent work that addresses this issue and focuses on enhancing object detection and tracking performance when the distance is far, or the point cloud is sparse,

is a system called Hydro-3D [9]. This work [9] proposed a spatial-temporal deep neural network that combines object detection features from a state-of-the-art object detection algorithm with historical information from the algorithm to enhance object detection. The sparsity issue intensifies when it comes to low-cost 2-D LiDAR sensors. Also, the measurements from the LIDAR sensor could be from any arbitrary object other than vehicles [10]. In this work, camera is fused with the LIDAR data not only to detect vehicles, but also to separately detect the front and side of a target vehicle and to find its corner. This helps measure its lateral and longitudinal distances accurately and subsequently estimate all trajectory variables of the vehicle.

When it comes to 3-D LIDAR and camera fusion, there are algorithms in the literature to enhance the LIDAR data using camera information [11]. In [12] for example, the 3-D LIDAR data is first mapped to the image space to find the regions of interest for vision-based object detection. These regions were then passed to a convolutional neural network (CNN) for object detection. This technique notably speeds up the object detection procedure, as CNN only searches for objects in areas proposed by the LIDAR. However, such an approach significantly relies on the performance of the LIDAR and is not a good solution for a low-cost 2-D LIDAR sensor. Our experiments show that in many cases, the camera picks up the object much sooner than the 2-D LIDAR sensor. Therefore, it is better to search for the vehicles in the entire image instead of just focusing on areas that include LIDAR measurements like in [12]. In [13], a 3-D vehicle detection system is proposed based on camera and LIDAR fusion. This work again, relies on the high-density measurements of the LIDAR sensor. In this work however, we focus on fusing camera and 2-D LIDAR information. The contributions of the proposed technology are two-fold:

- 1) This is the first paper that recognizes the importance of differentiating between the front and side regions of car. To the best of our knowledge, no other paper has previously recognized the importance of separately identifying the front and side regions of a car using computer vision algorithms. Reflection readings from the side of a car can constitute lateral distance measurement while reflection readings from the front of a car can constitute longitudinal distance measurement. Furthermore, the distance to the line of intersection between the front and side regions (i.e. the car corner) can provide both the lateral and longitudinal distances.
- 2) The paper shows how to combine a low-cost 2D-Lidar sensor with a monocular camera and how to utilize this sensor fusion using a high gain observer for an e-scooter protection application.

The outline of the paper is as follows. The vehicle detection and localization system are described in section 2, including the fusion of an object detection algorithm and the LIDAR measurements. The vehicle tracking and estimation system based on a high-gain observer are described in section 3, and section 4 contains extensive experimental results.

2. Vehicle Detection and Localization System

2.1. Overview and Hardware Configuration

This section describes the vehicle detection and localization system, developed based on a low-cost LIDAR sensor and a camera. The cost and size disadvantages of 3-D LIDAR sensors make them an inappropriate choice for e-scooters. Therefore, a 2-D RPLIDAR Mapper sensor with specifications mentioned in Table 1 is used. Fig. 2 shows a prototype scooter instrumented with the camera and LIDAR sensors to track vehicles and objects ahead of it. The specifications of the camera are shown in Table 2.

Although low-cost, the RPLIDAR sensor comes with two major shortcomings: its narrow vertical field of view (FoV) and its low density of data points. First, the narrow vertical FoV makes the performance of this sensor susceptible to the road's slope variations and limits the longitudinal range of measurement (to

values less than the nominal range mentioned in Table 1). Second, only a few measurement points can be obtained from objects, especially at longer distances. Due to these factors, the sensor does not have a robust output in outdoor applications, and consecutive measurements keep jumping and sliding on the vehicle surface. To improve the RPLIDAR measurements, we use images from a single camera to further determine:

- 1) Which measurement points are from a target vehicle (vehicle detection).
- 2) Whether the measurement points are from the side or front of the target vehicle.
- 3) The location of the critical points of the target vehicle (including front-right or front-left corners).

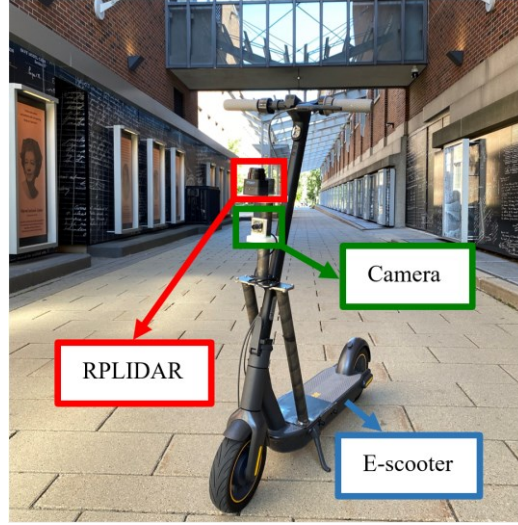


Fig. 2. RPLIDAR sensor and the camera on the Ninebot MAX e-Scooter.

Table. 1. RPLIDAR Mapper Sensor Specifications.

Sensor	Price	Range	Frequency of Data Refresh	Angular Resolution
RPLIDAR Mapper	~\$650	~40 m	10 Hz	0.75°

Table. 2. Camera Specifications.

Camera Model	Price	FPS	Resolution	FoV
Arducam 8MP 1080P USB Camera	~\$60	30	640×480	H = 90°, V = 65°

An overview of the vehicle detection algorithm is shown in Fig. 1. This is the algorithm used to detect whether an object is a vehicle. Other details of the LIDAR- Camera fusion algorithm to obtain smooth and reliable vehicle tracking (once a vehicle has been detected) are described in the following sections.

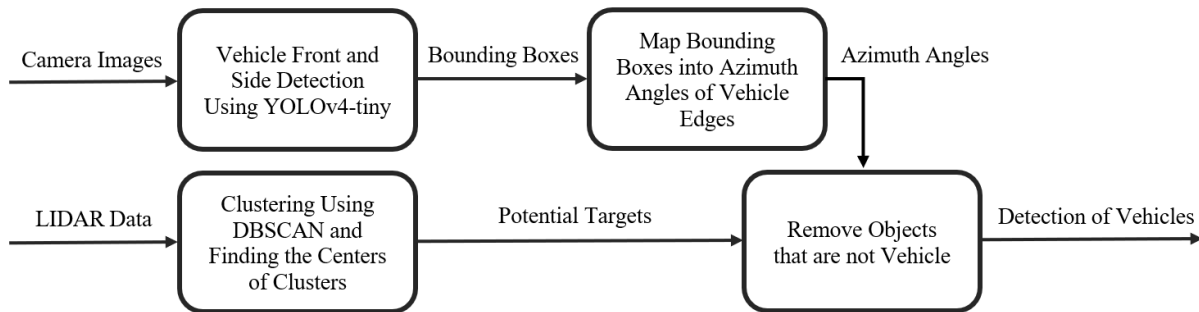


Fig. 1. Overview of the vehicle detection algorithm based on sensor fusion.

2.2. LIDAR- Camera Calibration

Multiple calibration factors need to be determined prior to the fusion of the camera and LIDAR measurements. First, the camera's intrinsic parameters and distortion coefficients are found using the *calibrateCamera(.)* function developed in OpenCV [14]. The procedure includes detecting the corners of a chessboard in multiple images (*findChessboardCorners(.)*) to be used for calibration. After calibration, the camera will have a smaller FoV than the original specifications that should be taken into account.

In addition to the intrinsic parameters calibration, there might be an angular yaw offset between the camera and LIDAR as a result of small installation misalignments. It is important to address this error for the camera-LIDAR fusion. To address this issue, we compare the azimuth angle of narrow objects (e.g., pedestrians at long distances) measured by RPLIDAR with the azimuth angle of the same objects measured using the images. The difference between these two azimuth angles is used as the offset yaw angle. Fig. 3 shows how the camera and LIDAR calibration and synchronization, enables us to localize with the two sensors and follow two different pedestrians at different distances perfectly. LIDAR measurements accurately lie on pedestrians:



Fig. 3. LIDAR measurement points (shown as red crosses) and images are synchronized such that the measurements accurately lie on the pedestrians.

2.3. Vehicle Front and Side Detection Using YOLOv4-Tiny

The LIDAR data is a very coarse 2-D map of the objects surrounding the e-scooter, including the trees, buildings, pedestrians, and vehicles. Our interest is particularly to track the trajectory variables of vehicles to predict the danger of any car- scooter collisions. Hence, vehicle detection in camera images is utilized to process the LIDAR data points and remove any measurements that are not coming from vehicles. Furthermore, the goal is to improve the 2-D LIDAR measurements, by deciding whether the LIDAR data points are from the side or front of a vehicle. This will be later also used to find the location of the right-front or left-front corner of the vehicle.

One possible approach in image processing is to find the key points of vehicles [15] to find their fronts and sides. However, the existing key point detection algorithms are slow and cannot be used in real-time. Another way to determine the side and front of the vehicle is to use object detection algorithms. There are many developed object detection algorithms in computer vision, including variations of the YOLO algorithm [16] that are the fastest ones. Among all these variations, YOLOv4-tiny (a scaled YOLOv4) is a small model that is suitable for cost-effective single board computers (SBC) that lack a powerful GPU. Due to its small-sized neural network, this algorithm works faster than most object detection algorithms on SBCs. Therefore, we utilize this model to detect and localize the fronts and sides of vehicles in the camera images.



Fig. 4. Laser sensor system, audio system, and the camera on the Ninebot MAX e-Scooter.

There is no open-source dataset available in the literature with separate annotations for the side and front of the vehicles as shown in Fig. 4 (closest dataset is [17]). Therefore, we created our own dataset of images with {"car", "car_front", "car_side", "car_back", "person", "bus", "bus_front", "bus_side", "bus_back", "truck", "truck_front", "truck_side", "truck_back"} as labels. Currently, we only use the labels "car_front" and "car_side" to train the YOLOv4-Tiny network. Our dataset contains 256 images, that

include 894 cars and 1337 unique annotations (670 “car_front” bounding boxes, and 667 “car_side” bounding boxes). The total unique annotations in the dataset are 2,723 that include other objects that are not utilized during the training process (e.g., person). We split the images into 90% of training images and 10% of test images.

With YOLOv4-tiny network trained for the labels “car_front” and “car_side,” we achieved 83% mAP (mean average precision) in localization for test images. It is noteworthy to mention that more recent object detection algorithms such as YOLOv7 [18] will potentially perform better in terms of precision. However, YOLOv4-Tiny is a small network suitable for SBC computers and while it maintains an acceptable precision, it is also light-weight and fast on SBC. Examples of the performance of the vehicle detection system are shown in Fig. 5.



Fig. 5. Examples of the vehicle detection system on experimental images. The output of the YOLOv4-tiny is shown as yellow and green bounding boxes for car_front and car_side classes respectively.

2.4. Mapping Bounding Boxes into Azimuth Angles

The camera images are a visual projection of the 3-D environment on a 2-D plane, and every pixel on the image represent a line in the 3-D space. Knowing the location of a pixel in the image plane and the camera focal length (in pixel numbers), one can extract the azimuth and elevation angles of the corresponding line in the 3-D space. We use the following model to find the azimuth angles of the vehicles based on the detected bounding boxes:

$$\varphi_i = \text{atan}\left(\frac{p_{x,i} - p_{x,0}}{f_c}\right) \quad (1)$$

where φ_i , $p_{x,i}$, $p_{x,0}$, and f are the azimuth angle, horizontal location of the pixel, horizontal principal offset, and the focal length (in pixels) respectively. Examples of mapping bounding boxes to azimuth angles are shown in Fig. 6.

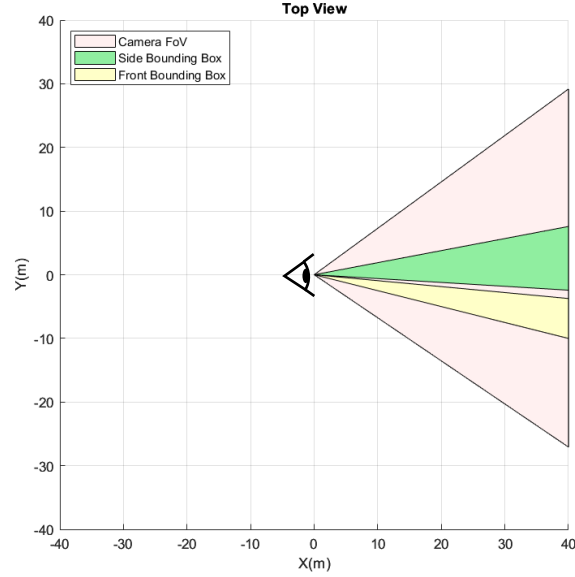
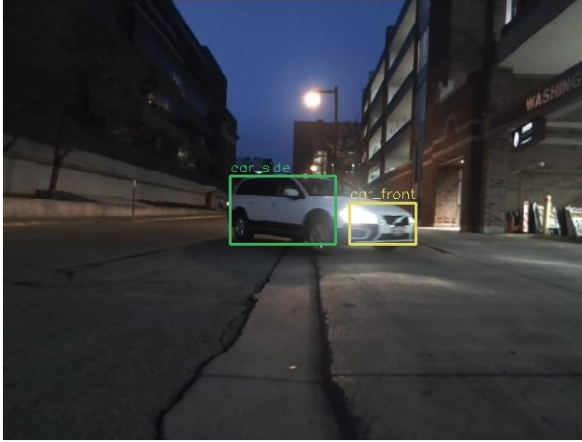


Fig. 6. Mapping bounding boxes to azimuth angles shown from top view.

2.5. 2-D LIDAR Output Processing

The output of the 2-D LIDAR sensor includes locations of several measurement points from various objects in the environment. The DBSCAN clustering algorithm [19] is used to remove outliers and find clusters corresponding to potential objects. DBSCAN identifies clusters by examining the local density of data in spatial data sets based on two pre-defined parameters: a minimum radius ρ and a minimum number of points within the radius N_{min} . DBSCAN groups data points together as a cluster if, for each point of the cluster, the neighborhood of a given radius ρ contains at least minimum number of points N_{min} . After implementing DBSCAN, a center is found for each cluster to represent it as the main location of the object.

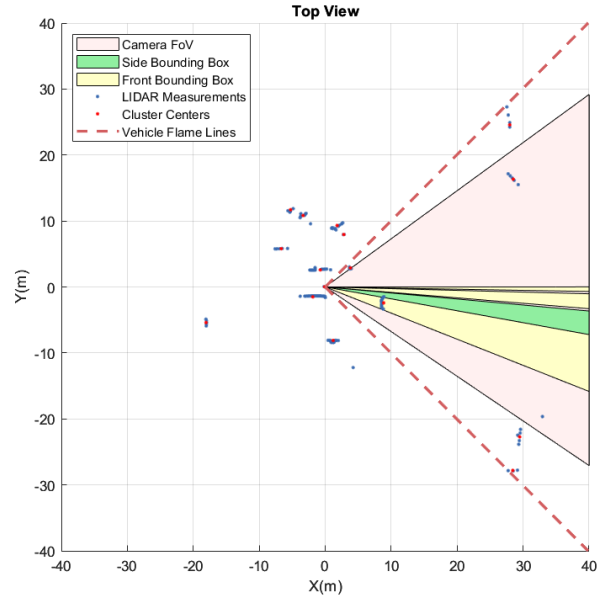
2.6. LIDAR- Camera Fusion for Vehicle Detection and Localization

Among all the objects (clusters) detected in the LIDAR point cloud, the clusters that are within detected bounding boxes created by YOLOv4-tiny are considered to be the target vehicles. In cases that there are more than one cluster within a bounding box range (as a result of errors in the bounding box detection), the cluster that covers a bigger azimuth angle portion will be chosen as the target vehicle. Figure 7 shows LIDAR and camera data before and after sensor fusion. As seen in Fig. 7b, fusion of the two sensors results in a clean set of LIDAR measurements corresponding only to a vehicle. Measurements from all other objects are eliminated.

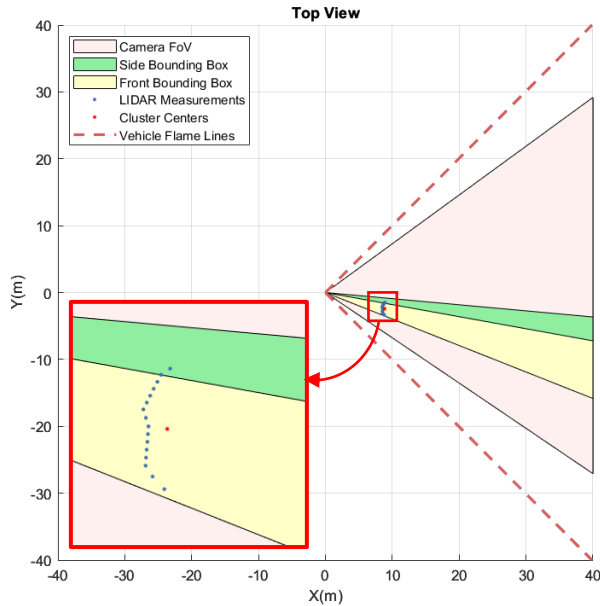
2.7. LIDAR- Camera Fusion for Vehicle Corner Detection and Localization

The location measurement of the target vehicle could be used to estimate its states including speed and orientation [20]. The location measurement however should be taken consistently from an identical point on the vehicle on all time frames. Otherwise, the estimation results will not be steady or accurate. Among all the points on the surface of the vehicle, the corners are the most recognizable ones to track. Depending on which bounding boxes are available, different strategies should be used for corner detection as shown in the flowchart in Fig. 8.

If only one of the front or side bounding boxes are available for a vehicle, the location of the corner of the vehicle is identified by drawing a line based on the existing LIDAR data and finding its intersection with the lines representing the edges of the bounding boxes. The accuracy of corner detection in this case will rely on whether the bounding boxes are tightly representing the object (with big intersection of union) or not. A schematic example of the procedure is shown in Fig. 9. To have robust performance in the presence of outliers, RANSAC [21] is used to fit a line based on the LIDAR data.



a) LIDAR and camera information before LIDAR- camera fusion.



b) LIDAR and camera information after LIDAR- camera fusion.

Fig. 7. Filtering LIDAR data based on the bounding boxes. The pictures on the first row show the entire LIDAR data, and the pictures on the second row show the filtered LIDAR data.

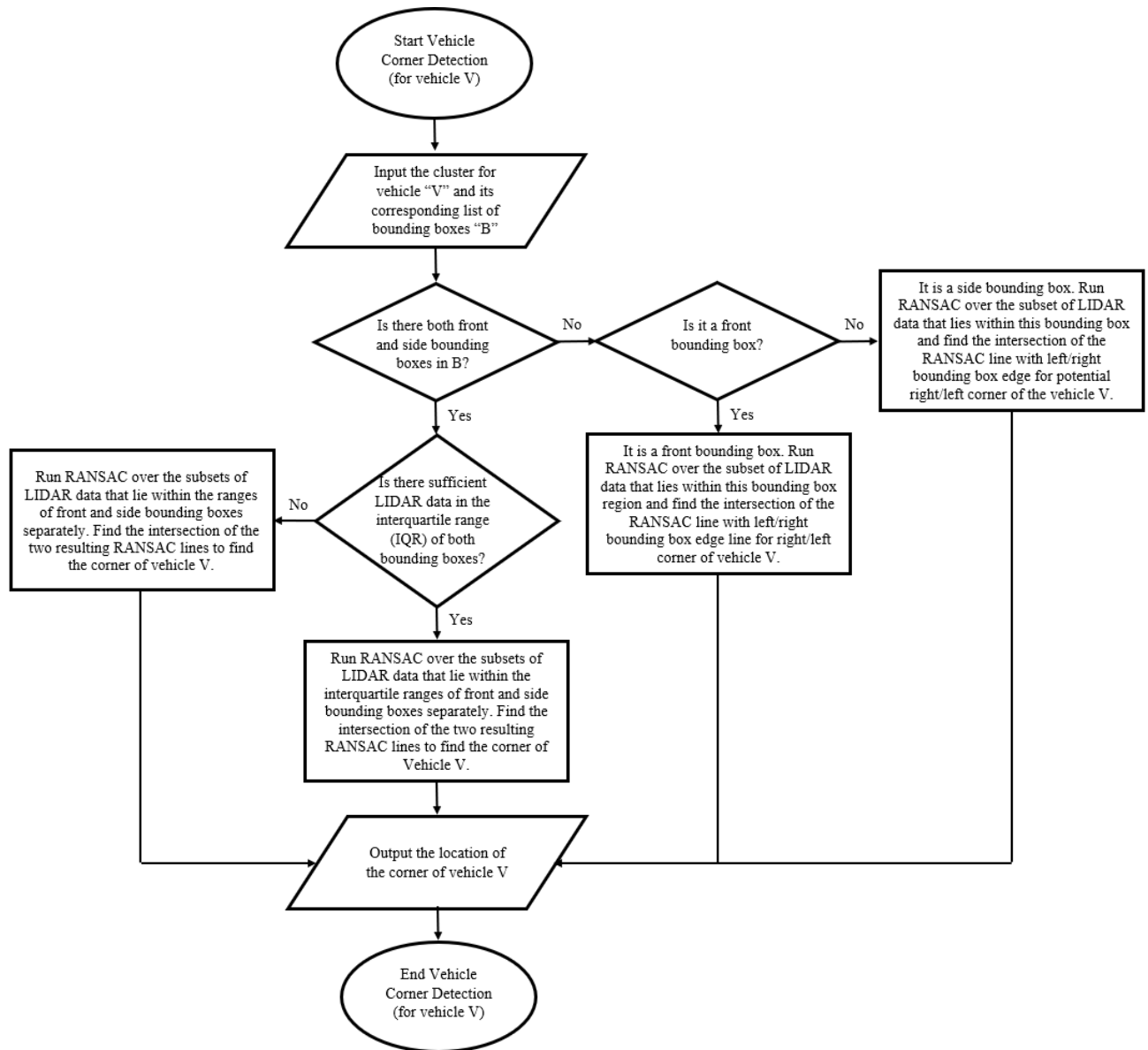


Fig. 8. Flowchart of the vehicle corner detection system.

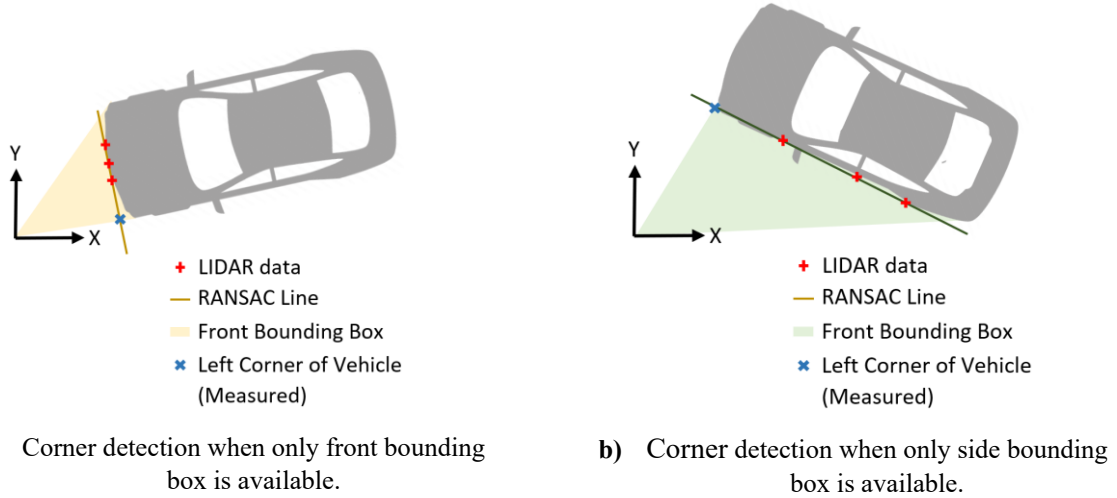


Fig. 9. Corner detection **a)** when only front bounding box is available and **b)** when only side bounding box is available.

When only one bounding box is available, the corner detection system relies on the edges of the bounding boxes that could be inaccurate. A different strategy is used when both the front and side bounding boxes are available for a vehicle. First, the LIDAR data is divided into two subsets based on the azimuth angles of both side and front of the vehicle: 1) LIDAR measurements from the side and 2) LIDAR measurements from the front. Second, the LIDAR measurements that lie within the interquartile range (IQR) of the bounding box azimuth angle ranges are counted. If the counted measurements do not exceed certain threshold value, two RANSAC lines are defined based on all the available LIDAR measurements for each subset. The intersection of the two lines is the corner of the vehicle as shown in Fig. 10 a). If the counted measurements meet the threshold value, two RANSAC lines are defined based on the LIDAR measurements that lie within the IQR of the bounding boxes. The intersection of the two lines is the corner of the vehicle as shown in Fig. 10 b).

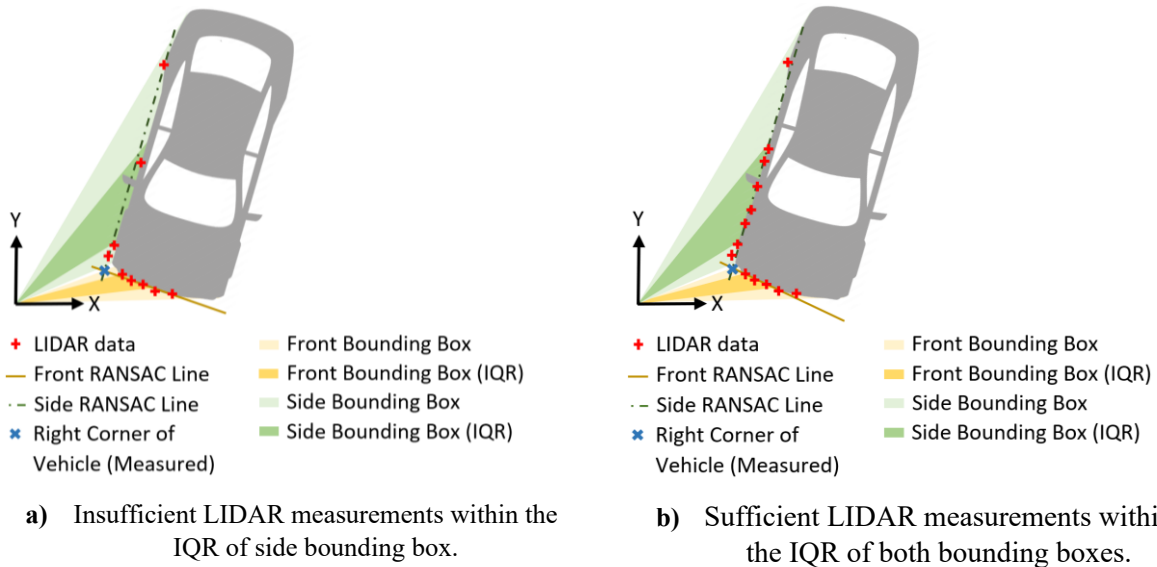


Fig. 10. Corner detection when both bounding boxes are available **a)** insufficient LIDAR measurements within IQR **b)** sufficient LIDAR measurements within IQR.

For example, Fig. 10 a) shows a schematic of the case where there is only one measurement from the side of the vehicle that lies within the IQR of the side bounding box. Therefore, the entire side LIDAR measurements are used in RANSAC to find the corresponding line. On the other hand, Fig. 10 b) shows the schematic of the case where there are multiple measurements from both the side and front of the vehicle that lie within the IQR of the side and front bounding boxes respectively. In this case, only the measurement points that lie within the IQR are used in RANSAC to find the corresponding lines.

3. Vehicle Tracking and Estimation

This section describes how the corner detection of each vehicle is used to estimate its trajectory variables including lateral and longitudinal position, velocities, and vehicle orientation.

3.1. Data Association

To find the new location of a previously tracked vehicle, the current closest vehicle to the previous time frame location is found. If there is no vehicle in the new time frame that is close enough to the previous tracked vehicle, that vehicle is classified as lost. Fig. 11 shows an example of the data association system and Fig. 12 shows the flowchart of the algorithm.

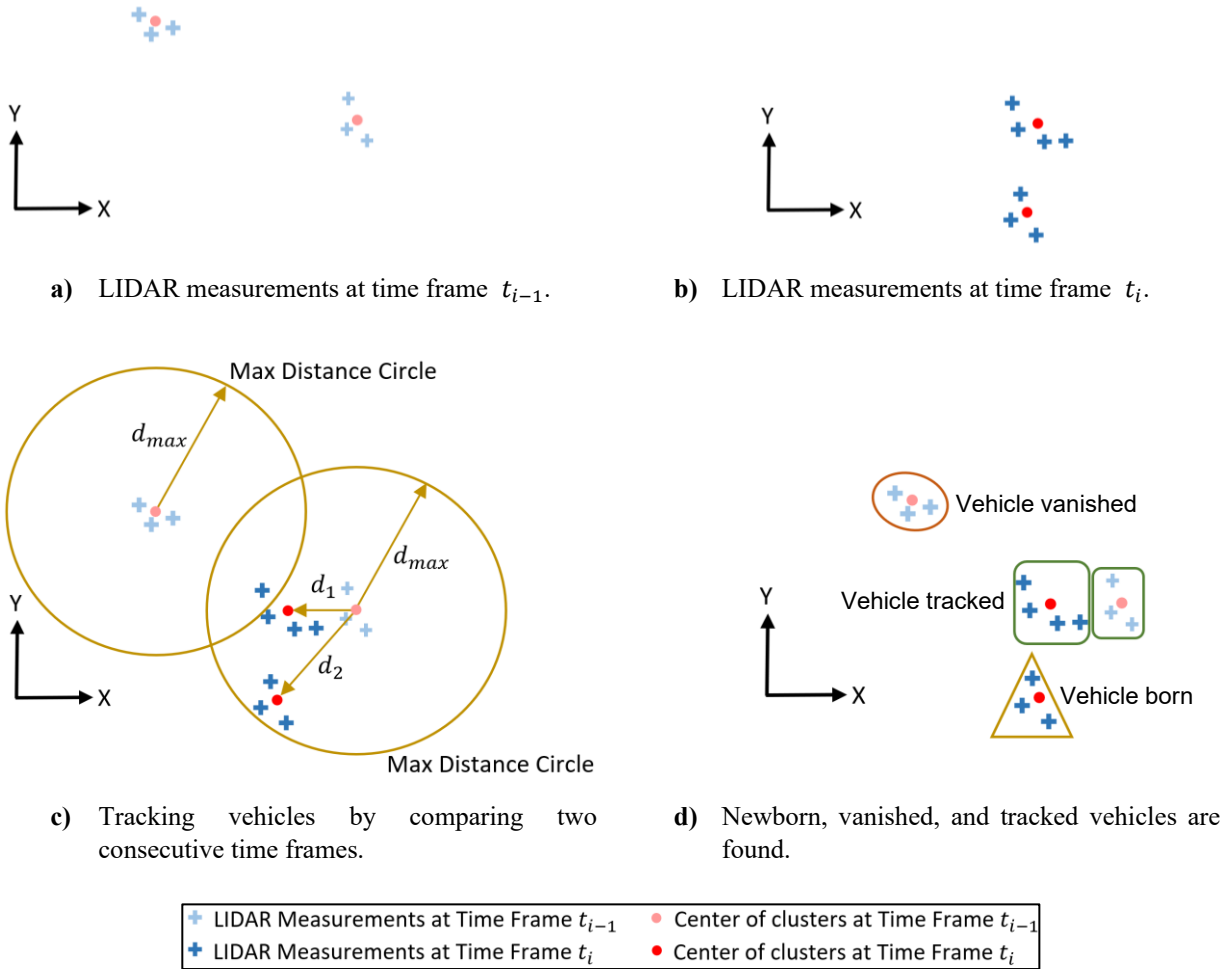


Fig. 11. Example of Vehicle Tracking on few clusters for two consecutive frames.

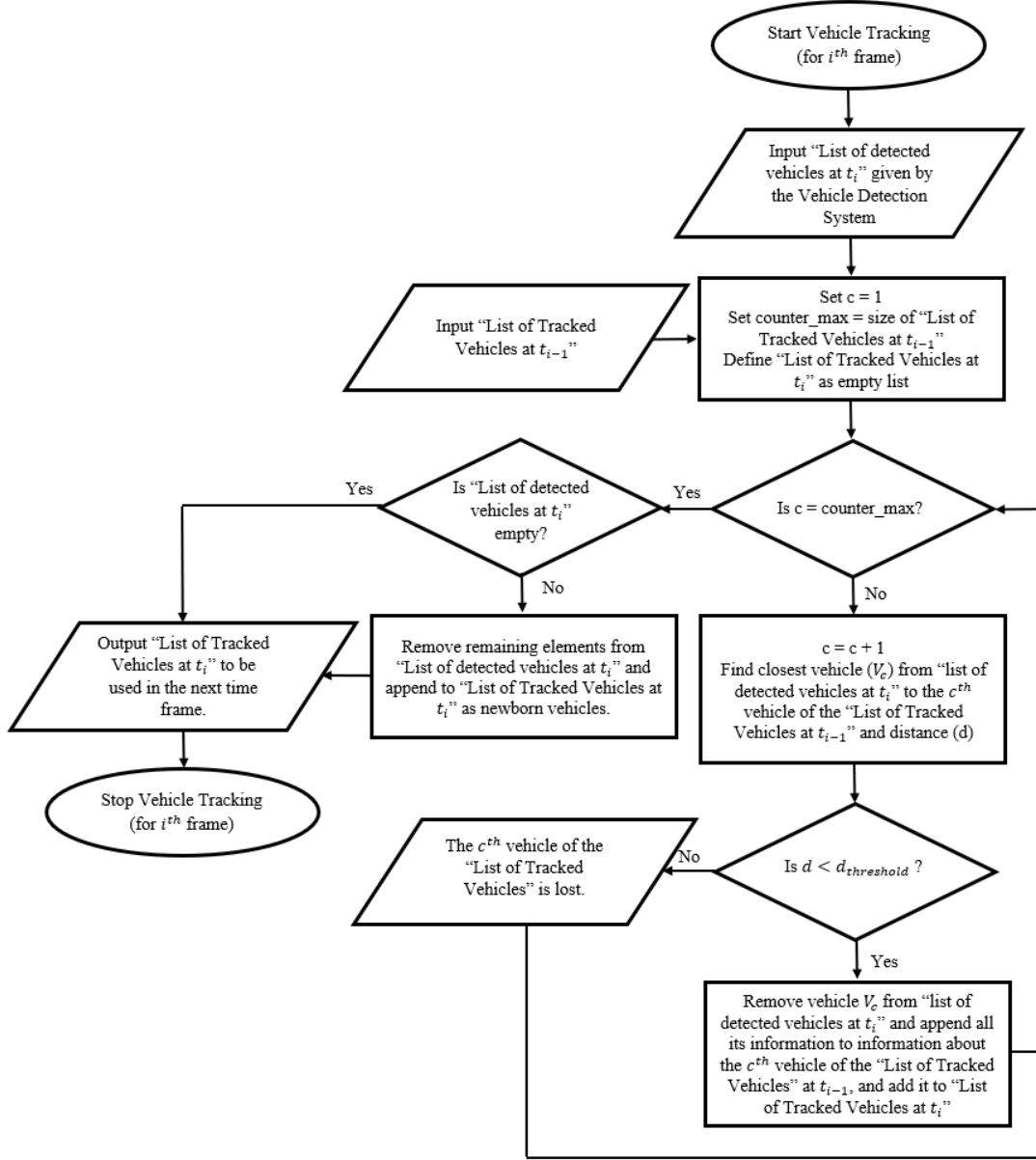


Fig. 12. Flowchart of the vehicle tracking system.

3.2. Vehicle State Estimation

Fig. 13 shows a vehicle with speed V , orientation (yaw) angle ψ , slip angle β , and steering angle δ_F . In [20], it is shown that with certain assumptions, the model (2) could be used for representing vehicle motion.

$$\dot{x} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{V} \\ \dot{A} \\ \dot{\psi} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} V \cos(\psi + \beta) \\ V \sin(\psi + \beta) \\ A \\ 0 \\ V \sin \beta / l_r \\ 0 \end{bmatrix}, \quad TTC_x = \frac{X}{\dot{X}} = \frac{X}{V \cos(\psi + \beta)}, \quad TTC_y = \frac{Y}{\dot{Y}} = \frac{Y}{V \sin(\psi + \beta)} \quad (2)$$

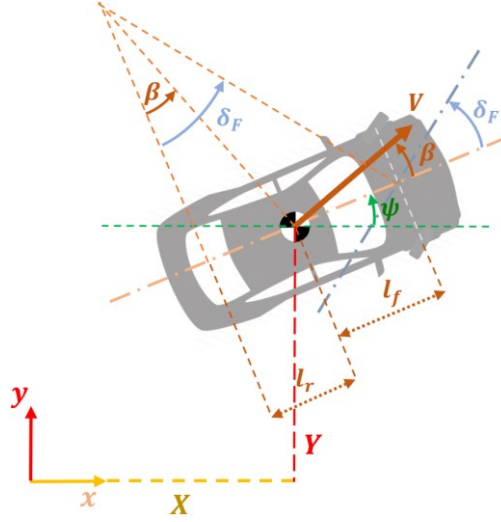


Fig. 13. Model variables for the motion of a target vehicle.

where parameter l_r is shown in Fig. 13. It is important to note that the model (2) is written w.r.t. a fixed frame. For a moving sensor frame that does not rotate, model (2) will represent the relative motion of the target vehicle w.r.t. the sensor frame (e.g., ψ will be the relative orientation). For the cases that the sensor frame rotates, the variables V , A , ψ , and β no longer represent the true relative speed, acceleration, orientation, and slip angle. But model (2) still will be valid for time-to-collision (TTC) estimation. It is also shown in [20] that it is possible to transform model (2) to model (3)-(5) which is in companion form.

$$\dot{z} = Fz + Gf(z), \quad y = Hz, \quad z = [X \quad \dot{X} \quad \ddot{X} \quad Y \quad \dot{Y} \quad \ddot{Y}]^T, \quad y = [X \quad Y]^T \quad (3)$$

in which:

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4)$$

and

$$f(z) = \begin{bmatrix} f_1(z) \\ f_2(z) \end{bmatrix} = \begin{bmatrix} -3z_6 \frac{z_6 z_2 - z_3 z_5}{z_2^2 + z_5^2} + 2z_2 \left(\frac{z_6 z_2 - z_3 z_5}{z_2^2 + z_5^2} \right)^2 \\ 3z_3 \frac{z_6 z_2 - z_3 z_5}{z_2^2 + z_5^2} + 2z_5 \left(\frac{z_6 z_2 - z_3 z_5}{z_2^2 + z_5^2} \right)^2 \end{bmatrix} \quad (5)$$

Define an observer with the following dynamics where $\hat{z}$ is the estimate states vector:

$$\dot{\hat{z}} = F\hat{z} + Gf(\hat{z}) + L(y - H\hat{z}) \quad (6)$$

Find the state error dynamics $\tilde{z}$ by subtracting (6) from (3) and defining $\tilde{f}(z, \hat{z}) = f(z) - f(\hat{z})$:

$$\dot{\tilde{z}} = \dot{z} - \dot{\hat{z}} = F\tilde{z} + G\tilde{f}(z, \hat{z}) + L(y - H\hat{z}) = F\tilde{z} + G\tilde{f}(z, \hat{z}) + LH\tilde{z} \quad (7)$$

Our goal is to find a single observer gain L such that the error dynamics (7) becomes stable in regions that the function $f_i(z)$ in (5) is Lipschitz. Assume that the Lipschitz constants for $f_i(z)$ are given as γ_1 and γ_2 :

$$\|\tilde{f}_1(z)\|_2 = \|f_1(z) - f_1(\hat{z})\|_2 \leq \gamma_1 \|\tilde{z}\|_2, \quad \|\tilde{f}_2(z)\|_2 = \|f_2(z) - f_2(\hat{z})\|_2 \leq \gamma_2 \|\tilde{z}\|_2 \quad (8)$$

Theorem 1. If there exists $P > 0$, $\lambda > 0$, Q , and $\theta > 1$ such that:

$$F^T P + P F - H^T Q - Q^T H < -\lambda I, \quad \theta > \theta_0 = \frac{2k_f \lambda_{\max}(P)}{\lambda} \quad (9)$$

in which $k_f = \sqrt{\gamma_1^2 + \gamma_2^2}$ and $\lambda_{\max}(\cdot)$ is the maximum eigenvalue, then the estimation error $\tilde{z}$ is exponentially stable by taking:

$$L = T(\theta) \times P^{-1} Q^T \quad (10)$$

where $T(\theta)$ is given by

$$T_B(\theta) = \begin{bmatrix} \theta & 0 & 0 \\ 0 & \theta^2 & 0 \\ 0 & 0 & \theta^3 \end{bmatrix}, T(\theta) = \begin{bmatrix} T_B(\theta) & 0 \\ 0 & T_B(\theta) \end{bmatrix} \quad (11)$$

The proof of this theorem is given in [20]. Using the SEDUMI solver in MATLAB to solve the LMI given in (9), the observer gain is then obtained as:

$$L = \begin{bmatrix} 67.20 & 823.1 & 2818.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 67.20 & 823.1 & 2818.5 \end{bmatrix}^T \quad (12)$$

The performance of the high-gain observer is highly sensitive to the quality of the raw measurements due to the large gain values and due to the fact that there is no noise-rejection property in the formulation of the high-gain observer. Therefore, the LIDAR-camera fusion could significantly improve the estimation results by directly enhancing the LIDAR raw measurements and consistently tracking the same point on the corner, namely a front vehicle corner, as explained in the previous section.

4. Experimental Results

4.1. Vehicle Tracking and Corner Detection

In this section, the experimental performance of vehicle tracking, and corner detection is evaluated in different scenarios. The vehicle corner detection is compared to strategies that could be exploited when there is no computer vision information. It is shown that the corner detection based on the computer vision and LIDAR is more accurate than cases where only the LIDAR is used. The hardware configuration being used for experiments is explained in section 2.1. It includes a prototype e-scooter equipped with an RPLIDAR sensor on top of an ArduCam camera. The specifications of these sensors are provided in Table 1 and Table 2. In the experiments, the e-scooter is set stationary on the street, facing oncoming traffic from the opposite direction.

Fig. 14 shows a vehicle moving straight in the opposite direction on an adjacent lane to the e-scooter. The vehicle's front and side bounding boxes are well detected. When the vehicle is not close to the sensors attached to the e-scooter, the available measurements are limited such that finding the corner of the vehicle will be challenging in the absence of images. In the absence of computer vision analysis, two strategies might be taken when the vehicle is moving as shown in Fig. 14: 1) taking the closest LIDAR point to the e-scooter 2) taking the point with the minimum lateral distance to the e-scooter. It is shown that with both

strategies, the vehicle corner detection will be weaker than the combination of LIDAR and computer vision as shown in Fig. 16.

Fig. 15 shows a vehicle turning right at the traffic intersection. In the initial frames, there are limited LIDAR measurements from the vehicle such that corner detection without computer vision is not possible. With the well-detected side bounding boxes however, it is possible to find the location of the corner with the strategies mentioned in the previous section. Fig. 17 shows that using only LIDAR measurements result in wrong location of the corner of vehicle especially in the initial frames based on the images. Fig. 18 shows the results of corner detection for a similar experiment on another set of data. Again, the results emphasize how using the vision- LIDAR fusion could improve the raw measurements for vehicle tracking.

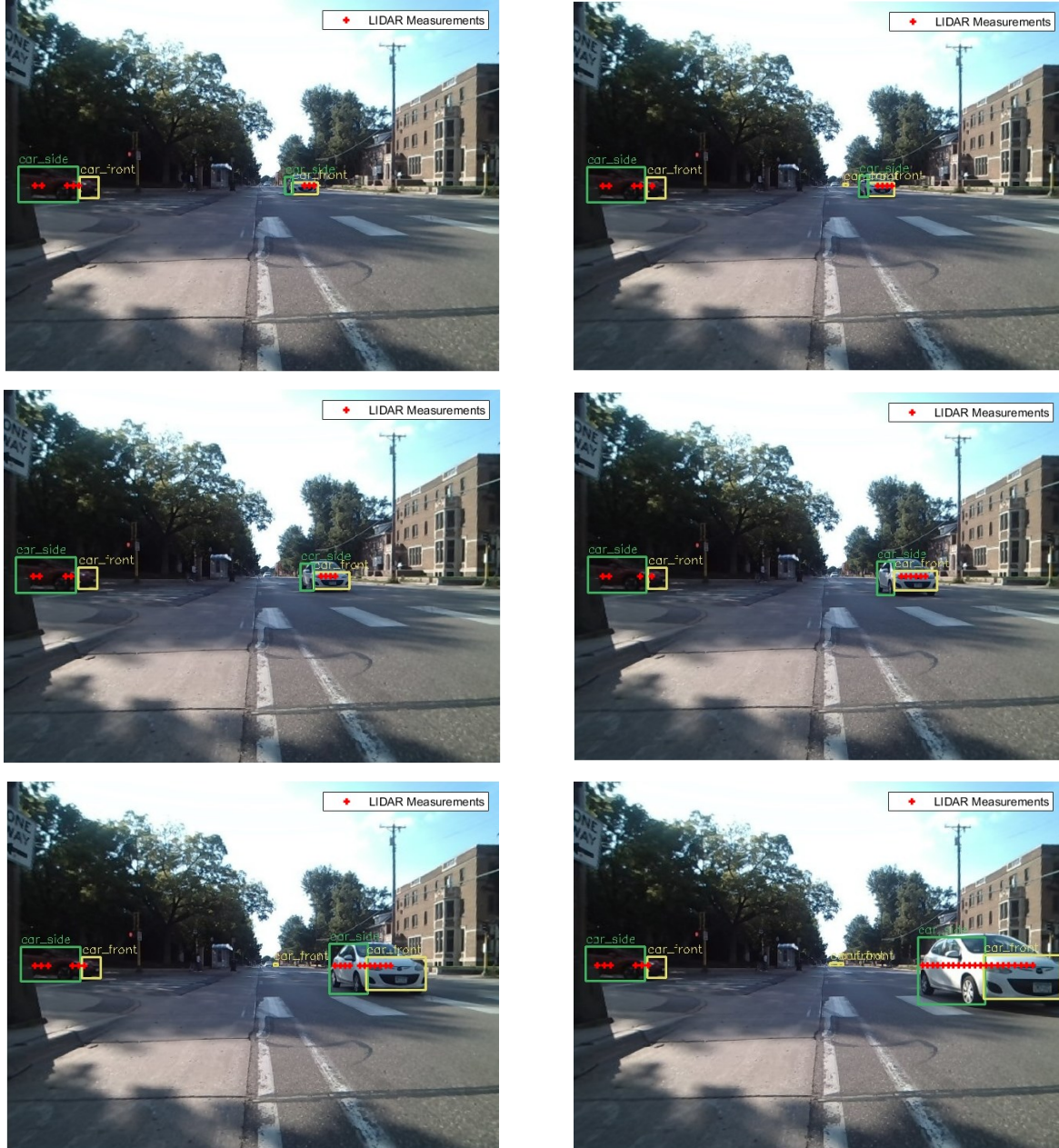


Fig. 14. Camera images and the LIDAR measurements for two vehicles. The white vehicle is moving straight in an opposite direction to the e-scooter, while the other vehicle is stationary.



Fig. 15. Camera images and the LIDAR measurements for two oncoming vehicles. The first vehicle is moving straight, while the other vehicle is turning right.

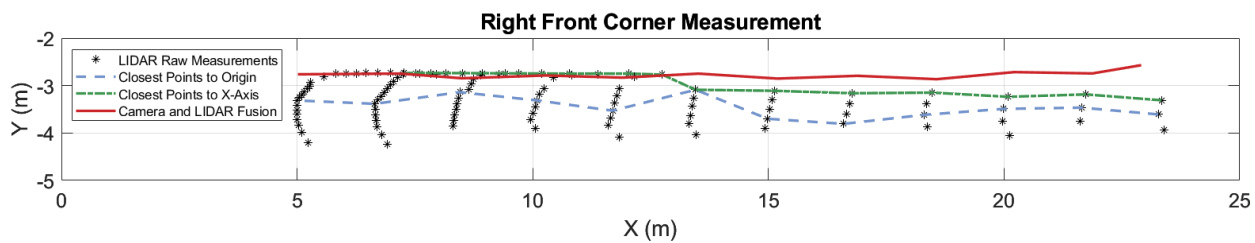


Fig. 16. Comparing various corner detection strategies for the experiment shown in Fig. 14. The vision- LIDAR fusion has the best performance.

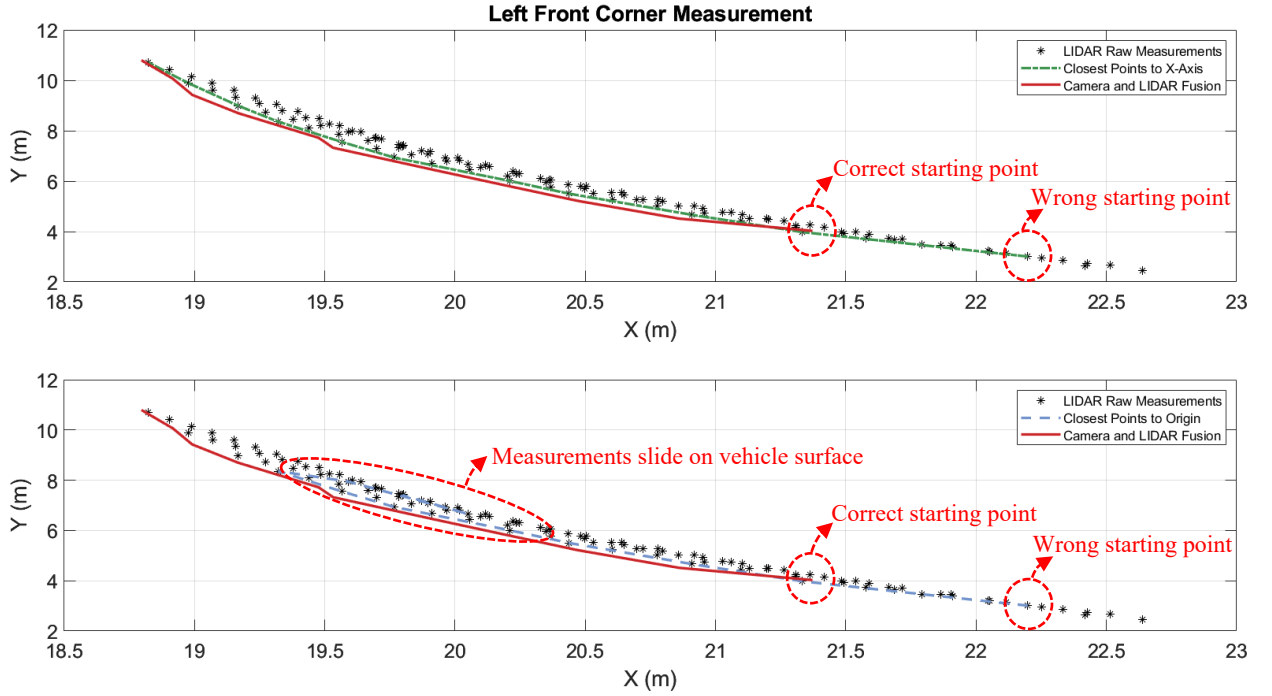


Fig. 17. Comparing various corner detection strategies for the experiment shown in Fig. 15. The vision- LIDAR fusion has the correct starting point and better performance than the other two strategies.

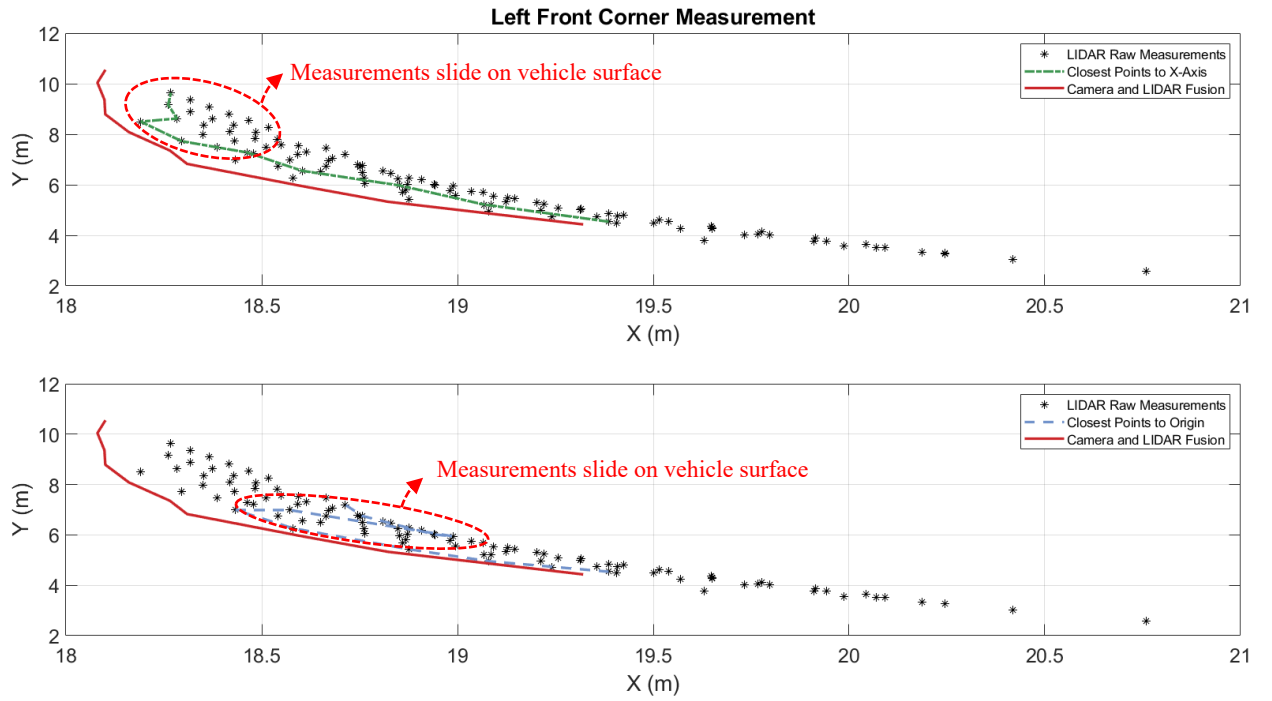


Fig. 18. Comparing various corner detection strategies for a similar experiment to the one shown in Fig. 15. The vision- LIDAR fusion has a better performance than the other two strategies.

4.2. Vehicle State Estimation

In this section, the results of the high-gain observer are compared with a number of different corner detection strategies. We use a forward and backward digital filter (filtfilt) to obtain the reference signals for the states in Fig. 19, Fig. 20, and Fig. 21. Note that while future measurements are used to find the reference signals, the high-gain observer only uses the current measurement.

Fig. 19 shows the estimation results of the case that the vehicle moves straight in the opposite direction to the e-scooter (case shown in Fig. 14). The initial LIDAR measurements are closer to the left corner of the vehicle, while the right corner is closer to the e-scooter. The corner detection results shown in Fig. 16 show how the LIDAR- camera fusion is a better than other strategies. Subsequently, Fig. 19 shows that the estimation results are better when both camera and LIDAR are being used. In addition to the position estimation where other strategies have offset errors, the speed and orientation estimation results are also much smoother when it comes to the camera and LIDAR fusion.

Fig. 20 and Fig. 21 show the estimation results of the case that the vehicle turns right (case shown in Fig. 15). The corner detection results shown in Fig. 17 and Fig. 18 show how the LIDAR- camera fusion is a better than other strategies. Subsequently, Fig. 20 and Fig. 21 show that the estimation results are better when both camera and LIDAR are being used. In addition to the position estimation where other strategies have errors, the speed and orientation estimation results are also much smoother when it comes to the camera and LIDAR fusion.

Table 3 compares the errors of various methods in the Figures 19- 21. The errors are computed by differences from the forward-backward filter reference values. In all cases, the camera and LIDAR fusion strategy works the best. After that, closest points to X-Axis and closest points to origin strategies are the best for the turning and straight motion scenarios respectively.

Table. 3. Observer's RMSE in experiments compared to forward and backward digital filter reference values.

Case #	Method	$\tilde{w}$ (m)	$\tilde{V}_x$ (m/s)	$\tilde{V}_y$ (m/s)	$\tilde{\psi}$ (rad)
Fig. 14 and 19	Camera and LIDAR Fusion	0.069	1.059	0.431	0.034
	Closest Points to X-Axis	0.842	4.231	1.190	0.095
	Closest Points to Origin	0.518	1.359	1.608	0.080
Fig. 15 and 20	Camera and LIDAR Fusion	0.041	0.464	0.800	0.110
	Closest Points to X-Axis	0.366	2.142	2.864	0.938
	Closest Points to Origin	1.098	2.981	7.041	1.458
Fig. 16 and 21	Camera and LIDAR Fusion	0.038	0.405	0.491	0.056
	Closest Points to X-Axis	0.362	0.469	1.637	0.214
	Closest Points to Origin	1.456	57.273	98.758	1.321

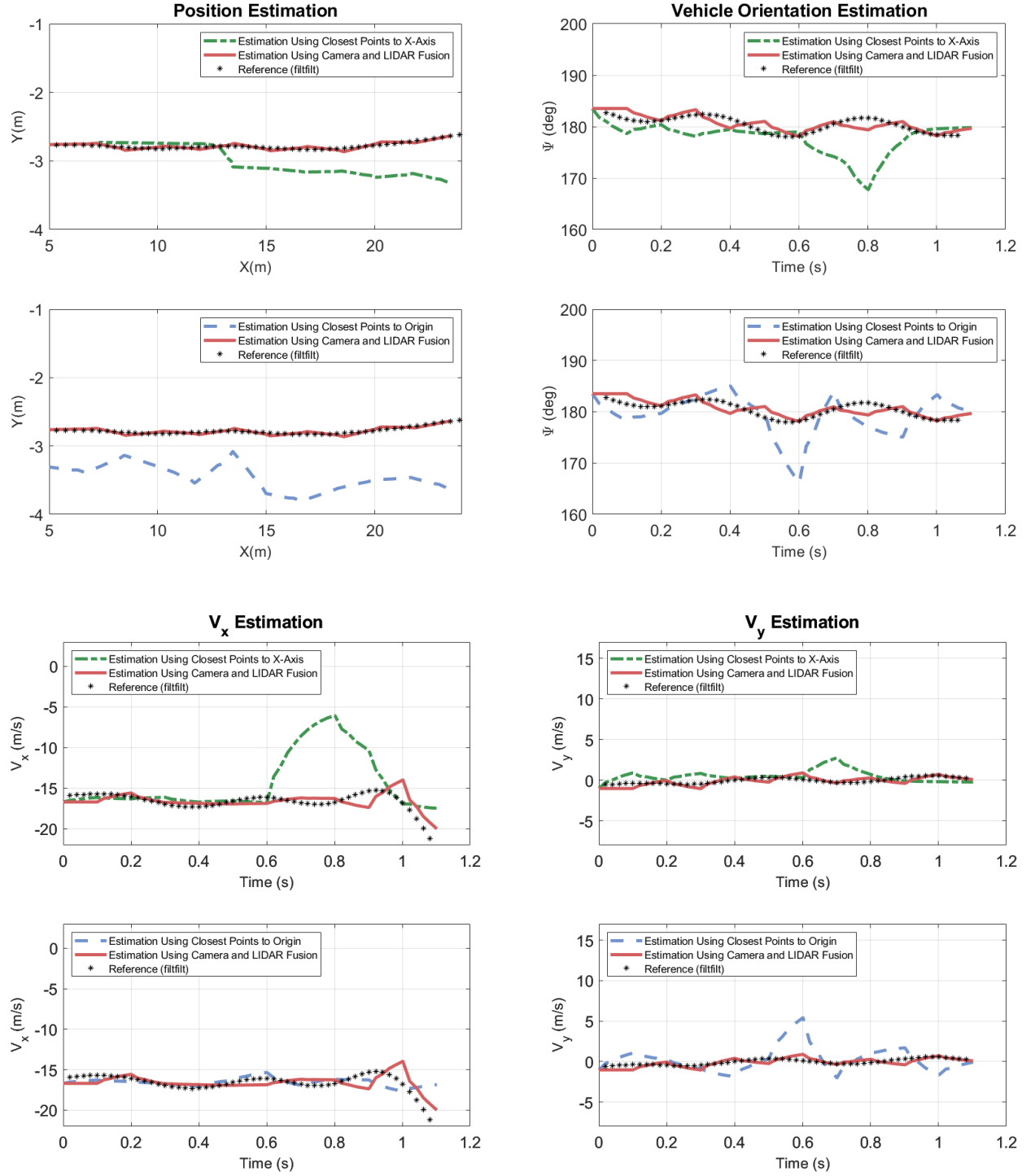


Fig. 19. Comparing various corner detection strategies for the scenario that the vehicle travels in opposite direction. The images for this scenario are shown in Fig. 14. The vision- LIDAR fusion has better performance than the other two strategies.

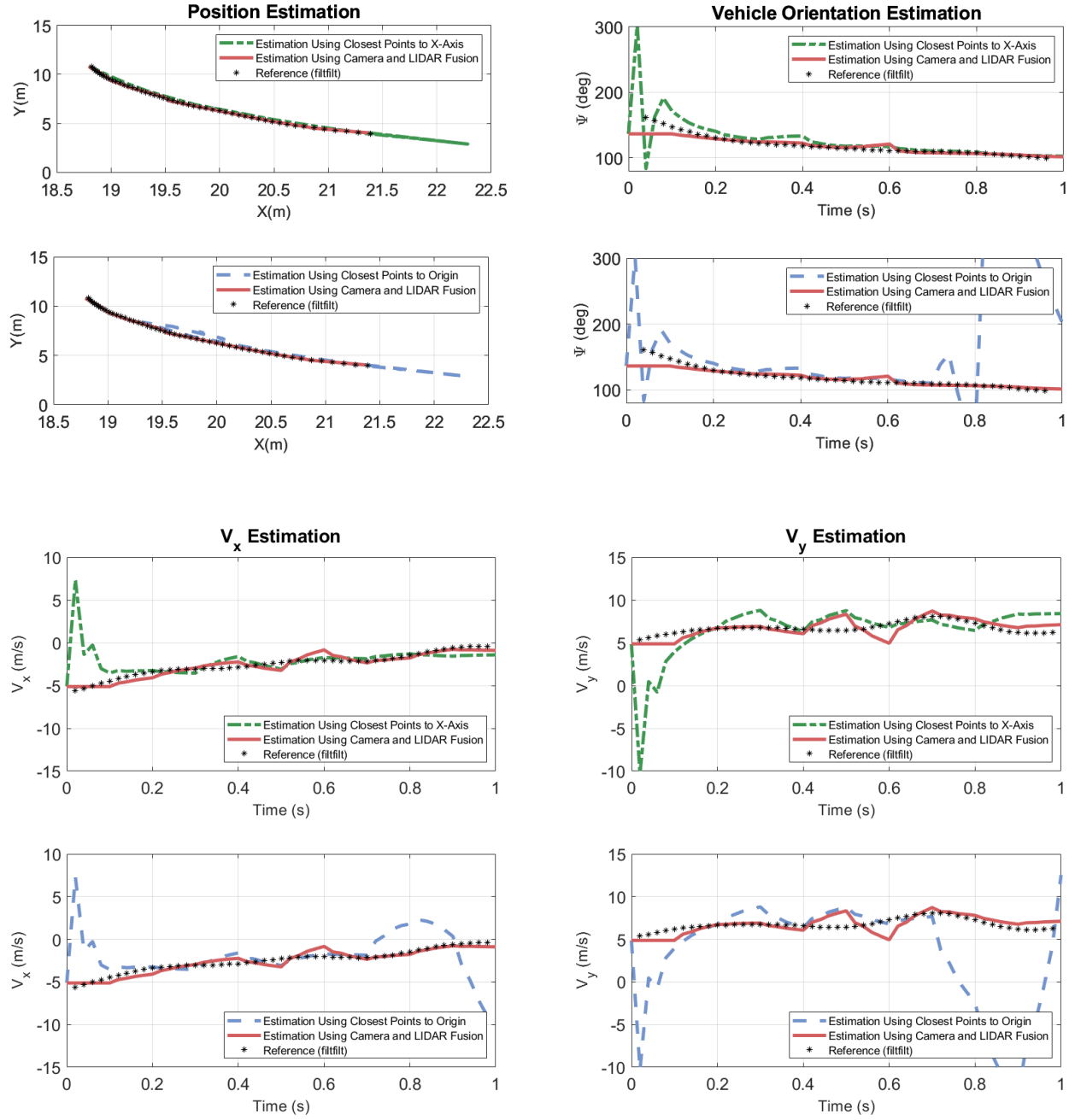


Fig. 20. Comparing various corner detection strategies for the scenario that the vehicle turns right. The images for this scenario are shown in Fig. 15. The vision- LIDAR fusion has better performance than the other two strategies.

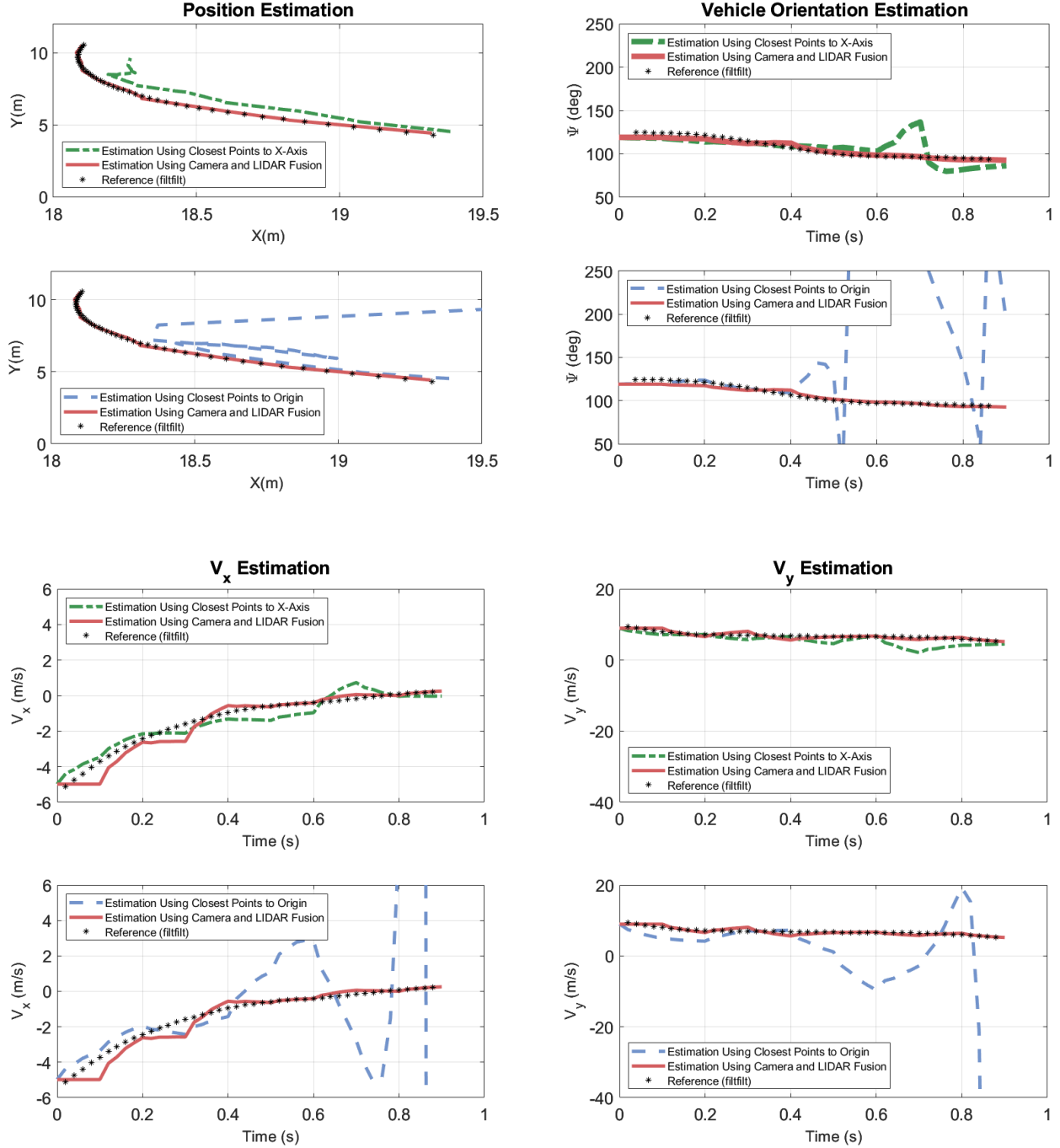


Fig. 21. Comparing various corner detection strategies for the scenario that the vehicle turns right. The similar images for this scenario are shown in Fig. 15. The vision- LIDAR fusion performs better than other strategies.

5. Conclusion

In this paper, a cost-effective vehicle detection and tracking system is developed based on 2-D LIDAR and camera fusion to protect electric micromobility devices, especially e-scooters, by predicting the danger of car- scooter collisions. The cost and size disadvantages of 3-D LIDAR sensors make them an unsuitable choice for micromobility devices. Therefore, a low-cost 2-D LIDAR is used. Although low-cost, the 2-D LIDAR sensor comes with major shortcomings such as a narrow vertical field of view and its low density

of data points. Due to these factors, the sensor does not have a robust output in outdoor applications, and consecutive measurements keep jumping and sliding on the target vehicle surface. To improve the performance of the 2-D LIDAR, a single monocular camera was fused with the LIDAR data not only to detect target vehicles, but also to separately detect the front and side of each target vehicle and to find the corners. It was shown that this corner detection method is more accurate than strategies that are only based on the LIDAR data. The key points and findings of this paper are summarized as follows:

- It is possible to reliably determine whether the readings from a 2D Lidar sensor come from the front or side of a vehicle by using sensor fusion with a monocular camera.
- It is possible to accurately find the location of the front left or front right corner of a car of oncoming traffic by using the sensor fusion algorithm developed in this paper.
- Further, by using a high-gain observer it is possible to estimate other trajectory variables such as vehicle orientation and velocity, in addition to the vehicle corner measurements.
- Finally, the root-mean-square-error is the smallest for the sensor fusion algorithm developed in this paper compared to other methods based purely on the Lidar data for finding the target vehicle's trajectory variables.

Acknowledgments

This research was supported in part by a research grant from the National Science Foundation (NSF Grant CPS 2038403).

6. References

- [1] Grand View Research, "Electric Scooters Market Size, Share & Trends Analysis Report", 2020 – 2030, Report ID: GVR-1-68038-196-2, February 2020.
- [2] C. Pekow, "E-scooter accidents surge as the micromobility devices' popularity grows," *Smart Cities Dive*, Online article, Accessed May 11, 2023, Published November 1, 2022. Available at: <https://www.smartcitiesdive.com/news/e-scooter-accidents-surge-as-the-micromobility-devices-popularity-grows/635478/>
- [3] A. Griswold (2020, Feb.). "At least 29 people have died in electric scooter crashes since 2018". Quartz.com. Retrieved from: <https://qz.com/1793164/at-least-29-people-have-died-in-electric-scooter-crashes/>
- [4] J. Tark, "Micromobility Products-Related Deaths, Injuries, and Hazard Patterns: 2017–2021," United States Consumer Product Safety Commission Report, September 2022.
- [5] H. Alai, W. Jeon, L. Alexander, and R. Rajamani, 2023, May. Rear Vehicle Tracking on a Smart E-Scooter. In 2023 American Control Conference (ACC) (pp. 1735-1740). IEEE.
- [6] S. Smaldone, C. Tonde, V. K. Ananthanarayanan, A. Elgammal, and L. Iftode, "The cyber-physical bike: a step towards safer green transportation," In 12th Workshop on Mobile Computing Systems and Applications, Phoenix, AZ, 2011, pp. 56 – 61.
- [7] Garmin. Varia Rearview Radar. Accessed: Jan. 26, 2019. [Online]. Available: <https://buy.garmin.com/en-US/US/p/518151>

- [8] W. Jeon and R. Rajamani, "Rear vehicle tracking on a bicycle using active sensor orientation control." *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 8, pp. 2638-2649, 2017.
- [9] Z. Meng, X. Xia, R. Xu, W. Liu, and J. Ma. "HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR." *IEEE Transactions on Intelligent Vehicles* (2023).
- [10] A. Mulyanto, R.I. Borman, P. Prasetyawana, and A. Sumarudin, 2020. 2d lidar and camera fusion for object detection and object distance measurement of ADAS using robotic operating system (ROS). *JOIV Int. J. Informatics Vis*, 4(4), pp.231-236.
- [11] H. Zhong, H. Wang, Z. Wu, C. Zhang, Y. Zheng, and T. Tang, 2021. "A survey of LiDAR and camera fusion enhancement". *Procedia Computer Science*, 183, pp.579-588.
- [12] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu. "Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications." *IEEE Sensors Journal* 20, no. 9 (2020): 4901-4913.
- [13] M. Wang, L. Zhao, and Y. Yue. "PA3DNet: 3-D Vehicle Detection with Pseudo Shape Segmentation and Adaptive Camera-LiDAR Fusion." *IEEE Transactions on Industrial Informatics* (2023).
- [14] G. Bradski. "The openCV library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25, no. 11 (2000): 120-123.
- [15] S. Kreiss, L. Bertoni, and A. Alahi, 2021. "Openpipaf: Composite fields for semantic keypoint detection and spatio-temporal association". *IEEE Transactions on Intelligent Transportation Systems*, 23(8), pp.13498-13511.
- [16] C.Y. Wang, A. Bochkovskiy, and H.Y.M. Liao, 2021. "Scaled-yolov4: Scaling cross stage partial network". In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition* (pp. 13029-13038).
- [17] A. Kumar, T. Kashiyaama, H. Maeda, and Y. Sekimoto, 2021, December. "Citywide reconstruction of cross-sectional traffic flow from moving camera videos". In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 1670-1678). IEEE.
- [18] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464-7475. 2023.
- [19] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", *KDD*, vol. 96, no. 34, pp. 49-60, Aug. 1996.
- [20] H. Alai, A. Zemouche, and R. Rajamani, 2023, May. "On Challenges in Coordinate Transformation for Using a High-Gain Multi-Output Nonlinear Observer". In *2023 American Control Conference (ACC)* (pp. 1024-1029). IEEE.
- [21] M.A. Fischler and R.C. Bolles, 1981. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, 24(6), pp.381-395.