



## Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors

Junxuan Zhao<sup>a</sup>, Hao Xu<sup>b</sup>, Hongchao Liu<sup>a,\*</sup>, Jianqing Wu<sup>b</sup>, Yichen Zheng<sup>b</sup>, Dayong Wu<sup>a</sup>

<sup>a</sup> Department of Civil, Environmental, and Construction Engineering, Texas Tech University, Lubbock, TX, USA

<sup>b</sup> Department of Civil and Environmental Engineering, University of Nevada, Reno, Reno, NV, USA



### ARTICLE INFO

**Keywords:**

Infrastructure-based LiDAR  
Pedestrians and Vehicles  
Clustering  
Classification  
Trajectory Extraction

### ABSTRACT

Light Detection and Ranging (LiDAR) is a remote sensing technology widely used in many areas ranging from making precise medical equipment to creating accurate elevation maps of farmlands. In transportation, although it has been used to assist some design and planning works, the application has been predominantly focused on autonomous vehicles, regardless of its great potential in precise detection and tracking of all road users if implemented in the field. This paper explores fundamental concepts, solution algorithms, and application guidance associated with using infrastructure-based LiDAR sensors to accurately detect and track pedestrians and vehicles at intersections. Based on LiDAR data collected in the field, investigations were conducted in the order of **background filtering, object clustering, pedestrian and vehicle classification, and tracking**. The results of the analysis include accurate and real-time information of the presence, position, velocity, and direction of pedestrians and vehicles. By studying the data from infrastructure-mounted LiDAR sensors at intersections, this paper offers insights into some critical techniques that are valuable to both researchers and practitioners toward field implementation of LiDAR sensors.

### 1. Introduction

Surveillance, control, and management of road traffic all rely on effective sensing and detection technologies. Among many commercially available infrastructure-based sensor technologies, inductive loop, microwave radar, and video camera are probably the most popular ones for long- and short-term traffic detection. Although each technology has its inherent strengths and weaknesses, a common problem of these detection means lies in their inability of getting trajectory-level data and low performance in accurate detection and tracking of pedestrians and vehicles. Besides, the number of onboard sensors has been increasing continuously in recent years because of their proved benefits in safety improvement. The ultimate goal along this line is obviously the autonomous vehicles which use a system of accurate sensors, sophisticated algorithms, and powerful computing to take over the driving task in the future.

Although general use of autonomous vehicles is likely years away, there should be no doubt the technology will eventually get matured and sophisticated enough for widespread use on public roads. Future deployment of these vehicles however, raises the question about whether and how future infrastructure-based detection systems should be developed in alignment with the self-driving technology to make the roads and all road users a seamless and cooperative system. Pedestrians for instance - the most

\* Corresponding author.

E-mail addresses: [junxuan.zhao@ttu.edu](mailto:junxuan.zhao@ttu.edu) (J. Zhao), [haox@unr.edu](mailto:haox@unr.edu) (H. Xu), [hongchao.liu@ttu.edu](mailto:hongchao.liu@ttu.edu) (H. Liu), [jianqingwu2015@gmail.com](mailto:jianqingwu2015@gmail.com) (J. Wu), [yichenzheng@unr.edu](mailto:yichenzheng@unr.edu) (Y. Zheng), [jasond.wu@ttu.edu](mailto:jasond.wu@ttu.edu) (D. Wu).

vulnerable road users, are only “passively” protected in an autonomous driving environment because the current vehicle-based sensing system lacks strategic real-time interaction with non-motorized road users, which may cause hazardous situation when vehicle-mounted sensors fail to detect their presence due to malfunction or system failure, as is evidenced by the recent fatal accident, in which a pedestrian was killed by a self-driving Uber vehicle ([The New York Times, 2018](#)).

To fill this gap, a real-time cooperative system will be needed in the long run to entail non-motorized road users receiving situational awareness and taking evasive actions through infrastructure-mounted sensors. In the short run, getting real-time trajectory data of all road users will be a big leap in traffic detection, which will greatly help researchers and practitioners elevate their capabilities in improving highway safety and enhancing traffic operation and control, traffic management, and performance measurement. First and foremost, the initial step towards infrastructure-based human-in-the-loop or all road users’ detection requires a reliable sensing technology. In the past decade, researchers have used LiDAR and vision-based approaches for this purpose with vehicle-mounted sensors and most of the works were [primarily focused on improving detection range and accuracy, as well as reducing computational cost](#) ([Premebida et al., 2007; Premebida et al., 2009; Ismail et al., 2009; Sivaraman and Trivedi, 2013; Ai and Tsai, 2016; Chavez-Garcia and Aycard, 2016; Du et al., 2017](#)). Although both video and LiDAR can fulfill the request, the reason why this research picked LiDAR is two folds: firstly, analyzing infrastructure-based video data requires much more processing and computing power; secondly, it is more expensive to outfit an intersection with cameras at all angles and with full coverage of both short and long distances. Additionally, the illumination condition has a significant effect on video quality – video recorded at night cannot provide as much valid information as video recorded during the daytime ([Mukhtar et al., 2015](#)). Generally speaking, the roadside LiDAR-based and onboard vision-based sensing systems are different in data quality, platform, and expected performance aspects: (1) Data collected from on-board vision sensing systems are mainly images (high-resolution), while data from LiDAR sensors are cloud points (high-accuracy but relative lower density); (2) Onboard vision-based sensing systems can only detect the environment around the vehicle, while roadside LiDAR-based sensing systems can cover a much wider detection range; (3) Onboard vision-based sensing systems need to work with other sensors like LiDAR, and depend on some other supportive data sources, such as high-resolution 3D maps and GPS information. However, roadside LiDAR sensing systems are expected to work individually.

Background filtering, object clustering, object classification, and real-time tracking the movement of an object are the four major steps for processing LiDAR data, regardless of where the sensors are installed. However, [models developed for onboard sensors may not be suitable to infrastructure-based detection](#). Besides the complexity issue, the working environment is quite different. In terms of background filtering for instance, since the background always changes with the movement of vehicles, the onboard sensing systems usually extract an object’s information from raw data using patch segmentation and classification ([Wang et al., 2012](#)). The visual-based methods, such as the mixtures of Gaussians (MoG) method ([Stauffer and Grimson, 2000](#)), statistical background modeling ([Wang et al., 2012](#)) and convolutional neural network deep learning method ([Sakkos et al., 2017; Babaee et al., 2018](#)) cannot be used since the LiDAR data are point clouds instead of pixel information.

Density-based spatial clustering of applications with noise (DBSCAN) and K-means clustering methods and their variations are among the most popular ones for object clustering of LiDAR data ([Morsdorf et al., 2004; Shackleton et al., 2010; Tonini and Abellán, 2014](#)). In this arena, [clustering two nearby objects into a single object and efficient clustering of objects at far distances are the common issues that still need to be addressed](#). In terms of object identification, a major approach is [feature-based machine learning classification](#) ([Wojke and Häselich, 2012; Azim and Aycard, 2012; Lee and Coifman, 2012; Cheng et al., 2014; Zangenehpour et al., 2015](#)). The performance of existing feature-based methods/algorithms heavily depends on the density of cloud points that provide detailed descriptions or specific characteristics of the objects. Onboard detection systems usually use both LiDAR and video sensors with higher resolution (thus higher cost) for perfect accuracy ([Premebida et al., 2007; Himmelsbach et al., 2008; Premebida et al., 2009; Gao et al., 2018](#)). While for infrastructure-based LiDAR sensors, although accuracy is also a critical factor, cost must be considered.

According to [Miyasaka et al. \(2009\)](#), [Azim and Aycard \(2012\)](#) and [Wang et al. \(2017\)](#), the mainstream methods for object tracking with LiDAR data include but are not limited to the [nearest neighbor \(NN\) method](#), [Kalman Filtering \(KF\)](#), [multiple hypothesis tracking \(MHT\)](#), and so on. [Zhao and Shibusaki \(2005\)](#) proposed a novel tracking system aimed at real-time monitoring of pedestrians’ behavior in a shopping mall by scanning the feet of pedestrians using a number of single-row LiDAR sensors. [Less time-consuming and low computational cost \(compared to video detection\)](#) are among their major findings from the study. They have also applied LiDAR to monitoring traffic at an intersection using a network of horizontal LiDAR sensors ([Zhao et al., 2012](#)) and accomplished an excellent tracking ratio of 92.9%. Worthy of note is they manually integrated the data from multiple sensors without considering the scenario when the same object was scanned on different sides in two frames. Additionally, a few critical parameters were estimated based on experience and not presented, which makes the study lack of generality and weak in [adaptability](#).

This article presents a systematic approach to detection and tracking of both pedestrians and vehicles by using infrastructure-based LiDAR sensors. As part of the study, a test site of three intersections were instrumented with 16-laser LiDAR sensors and communication devices. For each of the object clustering, pedestrian/vehicle classification and object tracking process, it investigates critical technical issues associated with roadside LiDAR data and explores solution methods and algorithms. In data clustering, a modified [DBSCAN](#) method was developed, along with an ingenious design of [dividing the detection range into subareas](#) based on the distance from the sensor to improve accuracy and reduce computational cost. In vehicle/pedestrian classification, a backpropagation artificial neural network ([BP-ANN](#)) was applied, and the [direction of the distribution of the clustered points](#) was found to be the key feature to distinguish pedestrian and vehicle clusters when the collected LiDAR data cannot give a fine description of objects. Tracking was conducted by a [discrete Kalman filter model](#) and a [distance-based object association process](#). In the end, the proposed methods and algorithms were integrated into a systematic approach for processing and analyzing infrastructure-based roadside LiDAR data. Using the data collected from the test site, experimental study shows the accuracy of the proposed approach for

detection, classification and tracking is above 95% in average, and the detection range for this level of accuracy is about 30 m (in one direction) for 16-laser LiDAR sensors. The error was mainly caused by occlusion and the outlier's issue.

This pilot study of processing roadside LiDAR data could be a valuable input for various traffic research, especially for the intelligent transportation field. The outputs of the proposed systematic approach are real-time trajectory data of all road users, which can be applied to: (1) Connected/Autonomous vehicles: detect traffic changes and risks around the corners by receiving real-time movement status of each road user in extended distances; (2) Near-crash analysis: study vehicle interactions at multiple scales and extract near-crash events to identify traffic safety issues, as well as recommend countermeasures; (3) Automatic pedestrian-crossing warning signals: pedestrians' crossing detection and potential vehicle-pedestrian crash prediction can be made based on historical trajectory and real-time location/speed/direction.

To the best knowledge of the authors, there are currently only a few efforts in systematically addressing the technical issues as well as the solution algorithms associated with infrastructure-based LiDAR sensors. Although this study was based on 16-laser LiDAR sensors, the proposed approach is applicable to other types of LiDAR sensors with adjustment of some parameters, such as the number of lasers, rotation frequency, vertical resolution of lasers and so on.

The rest of the paper is structured as follows: Section 2 introduces the LiDAR sensors and data. Section 3 presents the technical issues and proposed methods for processing and analyzing data from roadside LiDAR sensors. The case study is presented in Section 4 and Section 5 concludes the article with summary of contributions and limitations, as well as the perspectives on future work.

## 2. LiDAR sensor and data

LiDAR sensors use pulsed laser to detect objects. A LiDAR instrument principally consists of a laser, a scanner, and a specialized GPS receiver (Velodyne LiDAR, 2016). Laser channels, vertical field of view, and vertical resolution of laser beams are the common specifications for model selection. Models with more laser channels provide better coverage and higher accuracy but also come with higher price. Installation also plays an important role in the overall performance of roadside LiDAR applications, a good installation plan can be achieved based on a combined consideration of the technical features of the selected LiDAR model and the objective of the application (Zhao et al., 2019). In this study, LiDAR sensors were installed 6 ft above the ground without inclination.

The rotation frequency (5–20 Hz in general) of the LiDAR sensors can be customized by users. One data frame is generated after the sensor completes a 360°, three-dimensional (3D) scan (with line scan pattern) and the collected point clouds are stored in the packet capture (pcap) format (Velodyne LiDAR, 2016). The size of the data file is determined by the time of data collection and the number and complexity of the surrounding objects, usually about 2–3 megabytes per second per unit in our experiments. The output data from a LiDAR sensor include the location of each point (in X, Y, Z coordinates) and their distance to the sensor, intensity, laser ID, azimuth, adjusted time, and timestamp. Based on the GPS location of the LiDAR sensor and a reference point, all the points can be matched to their exact locations in the real world.

Fig. 1(a) demonstrates a selected frame of raw 3D LiDAR point clouds from the VeloView software; Fig. 1(b) shows one of the three intersections equipped with LiDAR sensors and communication devices for an on-going development of a connected system with all road users, and a snapshot of actual intersection scene. Details about the test site will be presented in the case study section.

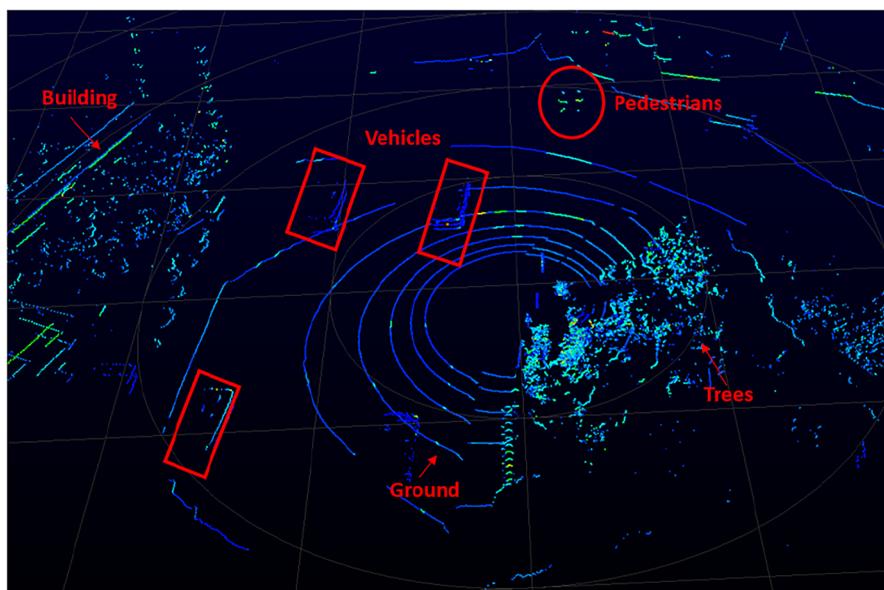


Fig. 1a. Raw LiDAR point clouds from VeloView.



**Fig. 1b.** LiDAR sensor with other equipment and snapshot at one of the three intersections.

### 3. Detection and tracking of pedestrians and vehicles

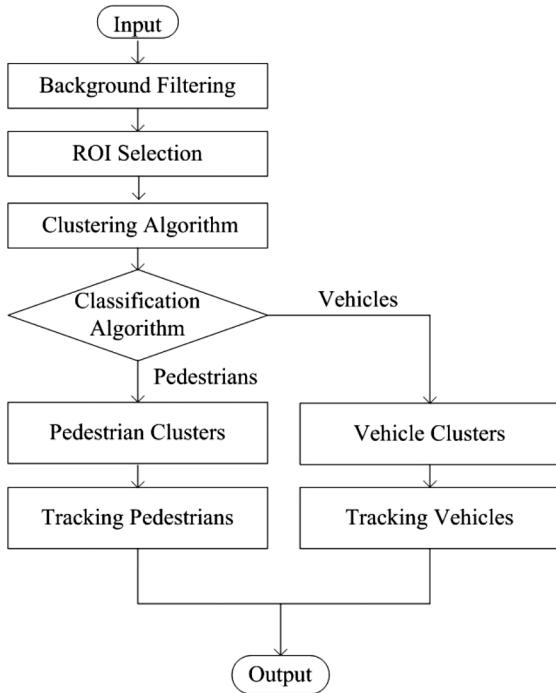
In this study, the LiDAR data collected from the test site were processed in the order of the following steps: background filtering, clustering, object classification, and tracking of movements. Background filtering is a prerequisite but rather independent process, the researchers have developed models prior to reaching the current stage of work. After the background filtering process, an intermediate process was developed to select the region of interest (ROI) so that the focus of the following steps can be placed on the selected ROIs such as a road segment or an intersection. The focus of this study is therefore placed on the clustering algorithm, identification of pedestrians and vehicles, tracking, and the examination of the effectiveness of the integrated approach. The flow chart in Fig. 2 depicts the systematic approach used in this research for processing and analyzing roadside LiDAR data.

#### 3.1. Background filtering

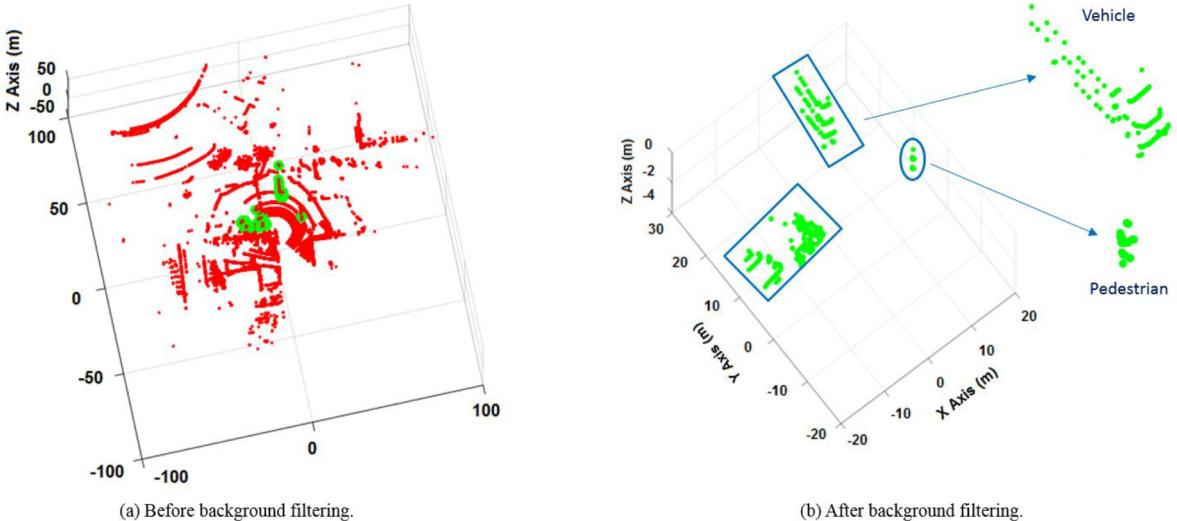
A complete set of raw LiDAR data include everything it detected such as buildings, trees, surface of the ground, and road users. A good background filtering model should keep the data points of road users to the maximum, while at the same time exclude as many as possible the points from the background objects. As part of this research project, an automatic 3D statistic background filtering algorithm (Wu et al., 2017, 2018a) has been developed by a group of researchers from the same team prior to the current stage of the work. The background filtering itself is a rather complicated process, this section briefly introduces their work with an aim to give readers a systematic view of our study without making this paper abundant and duplicative.

The background filtering algorithm involves frame aggregation, points statistics, threshold learning, and real-time filtering. It starts from aggregating multiple LiDAR data frames and divides the 3D space into continuous tiny cubes, subsequently a corresponding 3D matrix is built to store the number of points in each cube in the space. The analysis is then directed to detailed investigation of the cubes: if the number of total laser points in a cube is greater than a threshold that is automatically learned, this cube will be identified as a background space cube, and the identified background cubes are used to exclude background points. In the next interval, those laser points falling in the background cubes are excluded. The number of aggregated data frames for background identification was 3000 and the edge of each cube was 5 cm. The point thresholds were different, depending on the distance of the point from the LiDAR sensor: 19 (0–5 m), 10 (5–10 m), and 7 (> 10 m). Fig. 3(a) and (b) demonstrate the performance of the algorithm by showing the data before and after background filtering. In the experimental study, approximately 99.2% of background data points were successfully excluded from the raw data.

After background filtering, ROI was selected according to the boundary coordinates of the study area, usually the boundary of an intersection or a road segment (Wu et al., 2018b). The background filtering and ROI selection processes are helpful to increase detection/classification accuracy and reduce computational cost.



**Fig. 2.** Flowchart of the LiDAR data process.

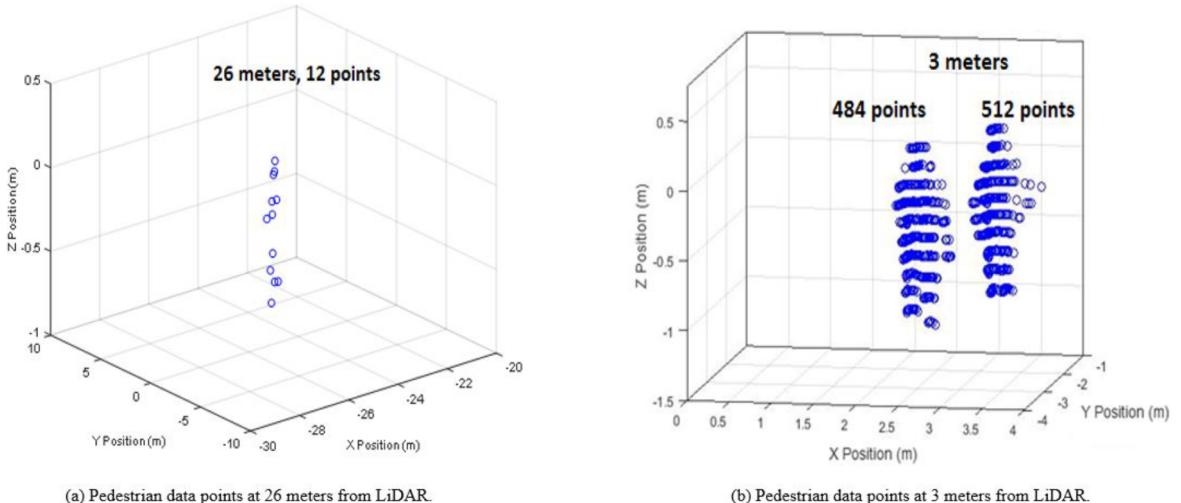


**Fig. 3.** Background filtering.

### 3.2. Object clustering and referencing

#### 3.2.1. Clustering

The principle of object clustering is to categorize various entities into clusters based on their similarities such that the entities in the same cluster present more similar properties than those in other clusters. The internal relationships between points in a cluster can be centroid, density, connectivity, distribution and so on, and what property determines a cluster and how to find clusters efficiently and effectively are the basic problems an algorithm needs to address (Theodoridis and Koutroumbas, 2009). The density-based spatial clustering applications with noise (DBSCAN) is by far one of the most popular density-based methods for clustering (Ester et al., 1996; Aggarwal and Reddy, 2013). It divides a dataset into three categories – core points, border points, and noise points within a predefined searching radius, with core points and border points being clustered points which describe the shape of the objects. Minimal Points (MinPts) and searching radius ( $\epsilon$ ) are the two primary parameters of the traditional DBSCAN algorithm (Ester



**Fig. 4.** Comparison of the numbers of clustered points at different distances.

et al., 1996), which are critical to the performance of the clustering process. If the number of data points within a searching area is greater than or equal to a predefined MinPts value, those data points will be clustered to form an object. A major advantage of the DBSCAN method is that it does not need to predefine the number of clusters or the number of objects, which is especially useful in transportation applications because it is almost impossible to predict the number of pedestrians or vehicles.

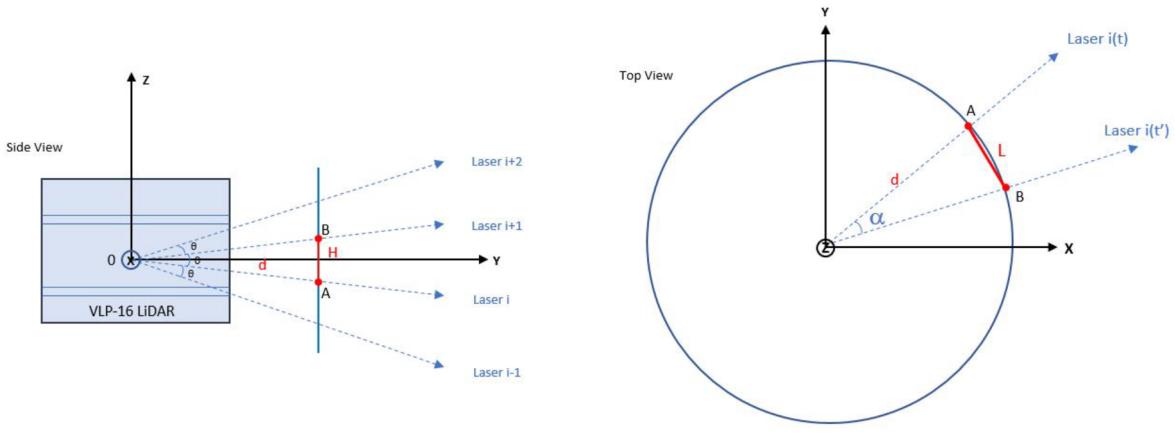
Due to the angled laser beams of LiDAR sensors and the shape/size of pedestrians and vehicles, the numbers of 3D points of pedestrians and vehicles may be different even at the same distance from the sensor. Also, if an object is located near the LiDAR sensor, denser data points can be collected to give a fine description of the object; otherwise, as distance increases, only sparse data points can be collected, especially for pedestrians (Li et al., 2016). To demonstrate, Fig. 4(a) and (b) show the total number of 3D points of pedestrians located at different distances: there are only 12 points of a pedestrian 26 m away from the LiDAR, while two other pedestrians who are only three meters away received 484 and 512 data points, respectively.

Because of the unique features of roadside LiDAR data, it is difficult to obtain accurate clustering results by using fixed MinPts and searching radius as introduced in the traditional DBSCAN algorithms. To effectively adjust the MinPts and searching radius at different locations, a modified DBSCAN clustering algorithm with adaptive MinPts values and searching radii was developed.

Firstly, the searching radius was chosen based on the LiDAR sensor's mechanical structure. As shown in Fig. 5(a), if the 2D distance between point A and the LiDAR sensor is  $d$  meters and B (with same distance  $d$ ) is the nearest adjacent point to point A, then the vertical height  $H$  between point A and point B is expressed in Eq. (1). This height is the minimum searching radius in a vertical direction (elevation) for clustering point A and point B into a same cluster.

$$H = 2d \cdot \tan(\theta/2) \quad (1)$$

where  $H$  is the vertical height between two adjacent points at the same distance  $d$  (meter),  $d$  is the 2D distance between the data point



(a) Vertical height between two adjacent points.

(b) Horizontal height between two adjacent points.

**Fig. 5.** Determine searching radius in DBSCAN algorithm.

and the LiDAR sensor (meter), and  $\theta$  is the vertical angular resolution of LiDAR sensor (For VLP-16,  $\theta = 2^\circ$ ).

Secondly, the horizontal distance between two adjacent points scanned by the same laser beam can also be calculated. As illustrated in Fig. 5(b), the horizontal angular resolution of the LiDAR sensor needs to be determined first, which is:

$$\alpha = \frac{360^\circ * f * n}{N_0} \quad (2)$$

where  $\alpha$  is the horizontal angular resolution of the LiDAR sensor (for VLP-16 with 10 Hz,  $\alpha = 0.2^\circ$ ),  $N_0$  is the data rate (total number of points per second collected by all lasers),  $f$  is the rotation frequency of the LiDAR sensor (Hz), and  $n$  is the number of laser pairs.

Then the horizontal distance between two adjacent points collected by the same laser beam can be obtained by Eq. (3), and this distance is the minimum searching radius in a horizontal direction for clustering point A and point B into the same cluster.

$$L = 2d * \sin(\alpha/2) \quad (3)$$

where  $L$  is the horizontal distance between two adjacent points collected by the same laser beam (meter),  $d$  is the 2D distance between the data point and the LiDAR sensor (meter), and  $\alpha$  is the horizontal angular resolution of the LiDAR sensor (Eq. (2)).

To get reliable detections, the vertical distance between the points scanned by two adjacent laser beams needs to be shorter than the minimum height of a road user, or the object may fall between the two laser beams without being detected. At a 30 m distance, the vertical height between two adjacent points is:  $H = 2d * \tan(\theta/2) = 2 * 30 * \tan(2^\circ/2) = 1.047$  m ( $\approx 1$  m), the pedestrians (an adult higher than 1 m) within 30 m from the LiDAR can be detected successfully. If a searching sphere (e.g.,  $\varepsilon = 1$  m) proposed by the traditional DBSCAN algorithm is used for clustering a crowd of pedestrians, pedestrians standing near each other (e.g., the distance less than 1 m) may be clustered into one group. To solve this problem, the adjusted method uses different radii in both vertical and horizontal directions and generates an ellipsoid searching space. The lengths of the semi-major axis and semi-minor axis are chosen based on the vertical height  $H$  and horizontal distance  $L$  calculated by Eqs. (1) and (3).

In addition, the value of MinPts was estimated based on the maximal number of points collected from the searching ellipsoid. For an ellipsoid with a semi-major axis ( $R_1$ ) and a semi-minor axis ( $R_2$ ), the maximum collectable number of points (TP) is approximated by Eqs. (4)–(6). When the laser beams shoot the ellipsoid perpendicularly, the number of points reaches maximum and the ellipsoid can be treated as an oval.

The oval is first approximated as a rectangle (size:  $2R_1 * 2R_2$ ) and the total number of laser beams shooting on the rectangle is:

$$NL = \left\lfloor \frac{2R_1}{H} \right\rfloor + 1 \quad (4)$$

and the total number of points from each laser within the rectangle area is:

$$NP = \left\lfloor \frac{2R_2}{L} \right\rfloor + 1 \quad (5)$$

and TP can be presented by Eq. (6):

$$\begin{aligned} TP &= \left( \left\lfloor \frac{R_1}{d * \tan(\frac{\theta}{2})} \right\rfloor + 1 \right) * \left( \left\lfloor \frac{R_2}{d * \sin(\frac{180^\circ * f * n}{N_0})} \right\rfloor + 1 \right) * \frac{\pi R_1 R_2}{2R_1 * 2R_2} \\ &= \frac{\pi}{4} \left( \left\lfloor \frac{R_1}{d * \tan(\frac{\theta}{2})} \right\rfloor + 1 \right) * \left( \left\lfloor \frac{R_2}{d * \sin(\frac{180^\circ * f * n}{N_0})} \right\rfloor + 1 \right) \end{aligned} \quad (6)$$

where

TP is the maximum number of points,

$N_0$  is the data rate,

$f$  is the rotation frequency of LiDAR sensor (Hz),

$n$  is the number of laser pairs,

$d$  is the 2D distance between point and LiDAR sensor (meter),

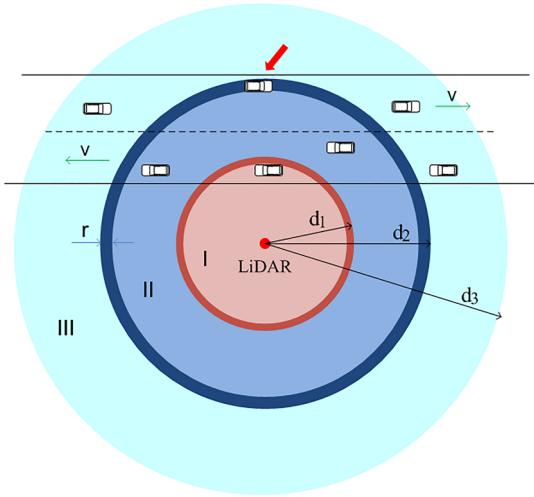
$R_1$  is the semi-major axis of the ellipsoid (meter),

$R_2$  is the semi-minor axis of the ellipsoid (meter), and

$\theta$  is the vertical angular resolution of LiDAR sensor (degree).

Considering the angles between the laser beams and the objects in real situations, the total collectable number of points of an object should be equal to or less than the calculated TP value. The value of MinPts can be determined based on the calculated TP and field data analysis, as to be introduced in the case study section.

When MinPts and searching radii change over distance, calculation of every LiDAR point may lead to heavy computation. To solve this problem, the proposed procedure divides the sensor's detection range into three sub-areas (I, II, III) based on the distances from the sensor (as shown in Fig. 6). Each subarea uses the same searching radius and the MinPts value that are determined by the outer edge (farthest to the sensor) of the subarea. Each subarea is extended to form an overlapped ring, with the ring's width larger than the



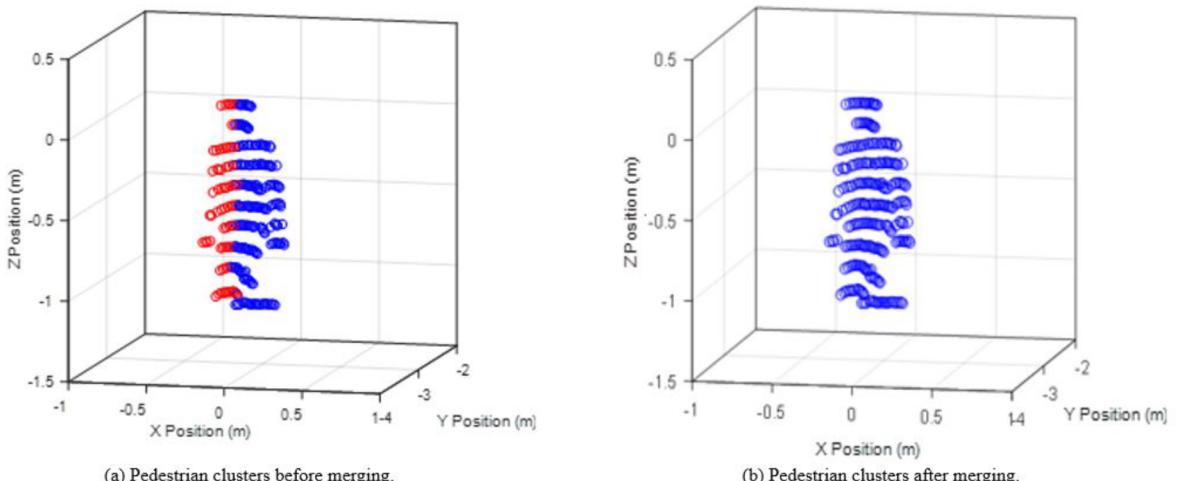
**Fig. 6.** DBSCAN area division.

width of a vehicle or a pedestrian. The overlapped ring method can avoid an object at the border of the subareas being identified as two different targets. Fig. 6 shows an example of three divided areas with two overlapped rings ( $r$ ). After clustering, clusters (e.g., the vehicle pointed by red arrow in Fig. 6) within the overlapped ring areas are compared to determine whether any of them needs to be merged. If the distance of the center points of two clusters is less than a predefined threshold, those two clusters are merged as one. Fig. 7(a) and (b) demonstrate the clustering results of a pedestrian standing in an overlapped area with/without checking the merging criteria. With the modified DBSCAN method, the accuracy of object clustering is greatly improved, especially for the objects away from the sensor and close to each other.

### 3.2.2. Referencing

After clustering, all detected objects are represented by clusters in 3D space. A reference point on the XY plane needs to be found to locate the position of each cluster. This reference point will later be used for tracking clusters and extracting trajectory information. For pedestrian clusters, the mean center of each cluster can be used as its reference point, while for vehicles, bounding boxes need to be determined for better referencing. This method is introduced below.

For each cluster, all clustered points are projected onto the XY plane to find the bounding box, i.e., the minimum rectangle that covers all clustered points based on the minimal and maximal values projected in X and Y direction. In an ideal situation depicted in the upper right corner of Fig. 8, four vertices (A, B, C, and D) of the rectangle can be used to estimate the corresponding corners of a vehicle, and the center point O can be used to indicate the location of a vehicle. In real world however, the data points of a vehicle are usually collected from different angles and the coverage of every corner cannot be guaranteed. For instance, when a vehicle (Veh1) is on the left side of the LiDAR (approaching), A, B, and C points can be detected; when a vehicle (Veh2) is in front of the sensor (crossing), only A and B points may get detected; when a vehicle (Veh3) is on the right side of the sensor (leaving or departing), only



**Fig. 7.** Examples of merging clusters in an overlapped area.

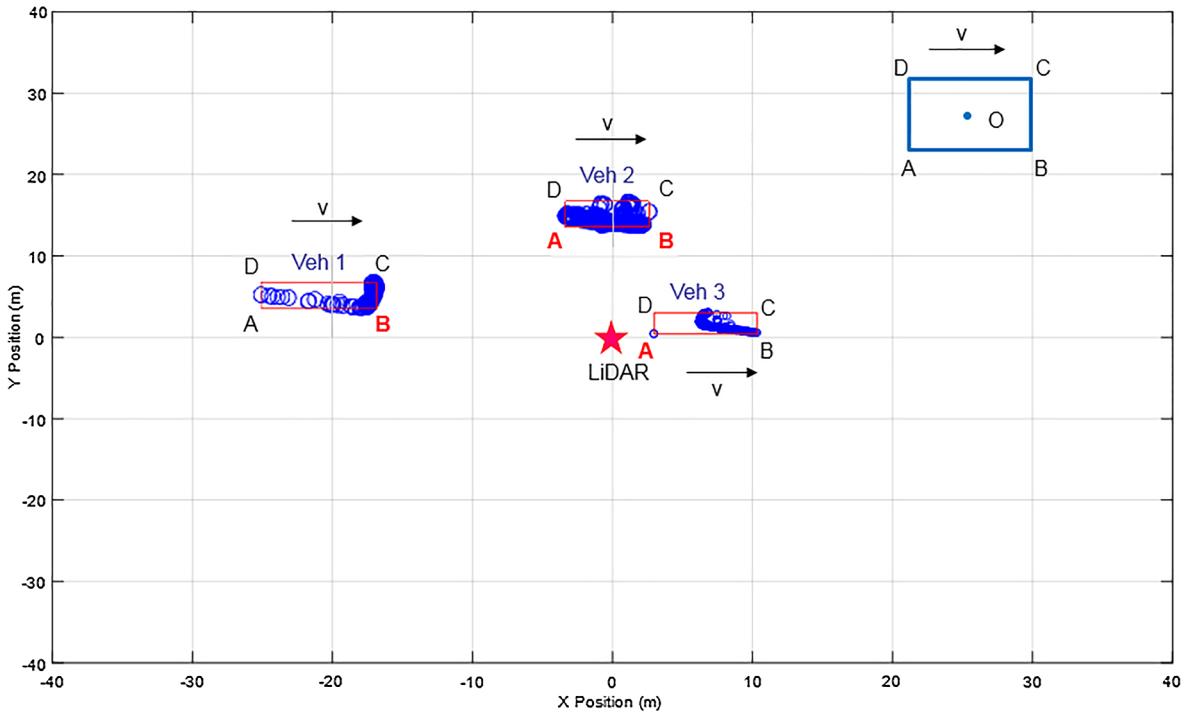


Fig. 8. Reference point of a cluster based on bounding box.

A, B, and D points can be detected. Nonetheless, it is clear that the corner closest to the LiDAR sensor provides the most reliable and accurate information of an object's location. Therefore, for approaching vehicles, we use the inner front corner (i.e., the corner closer to the sensor) of the bounding box as the reference point (Point B for Veh1); for departing vehicles, we use the inner back corner of a bounding box for cluster referencing (Point A for Veh3); for crossing vehicles, we use the inner side of both front and back corners as the reference points (Point A and B for Veh2). If the vehicles have different reference points between two continuous frames, the reference point for trajectory extraction (e.g., speed calculation) will be chosen as the same corner of the associated cluster from the previous or the next frame. In other words, the proposed process continuously tracks inner front and back corners from each frame, and selects inner front or inner back corner based on the relative location of the LiDAR sensor and the vehicle. This position information will then be used in tracking to guarantee the reference point is attached to the object consistently.

The quality of a cluster's reference point highly depends on the accuracy of the bounding box. “Cluster Deficiency” is a common issue caused by occlusion and/or perspective shadow, which affects the integrity of a cluster, and thus the accuracy of a bounding box. Since the bounding box is designed to bound the cluster, the position of the reference point may be shifted if the corresponding corner of a cluster cannot be detected. Fig. 9 demonstrates such an issue: the target vehicle was leaving the LiDAR sensor and the inner back corner was selected as the reference point for positioning; as the side and partial rear portions of the vehicle were not detected from the frame 88, the bounding box for this frame could not be accurate. In this study, we tackled this problem by data

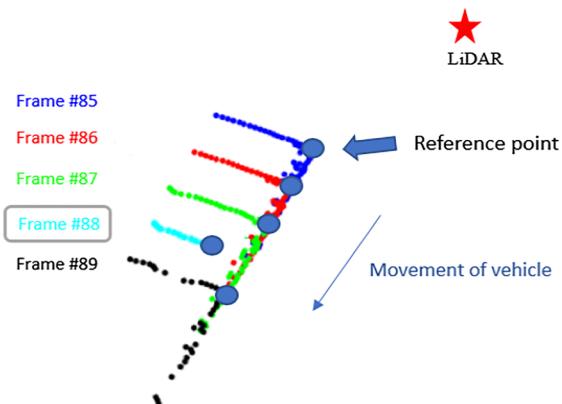
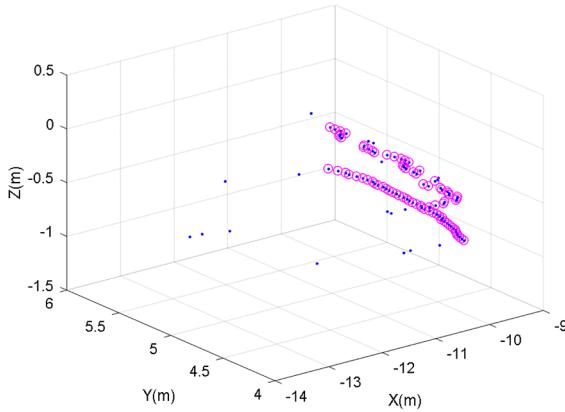


Fig. 9. Reference point and cluster deficiency.



**Fig. 10.** Filtering outliers in a vehicle cluster.

training and by continuously checking the bounding box's length-to-width ratio. Usually, this ratio can only be slightly changed around the ground truth under non-occlusion conditions due to the factors such as the vibration of LiDAR sensors. If the ratio changes abruptly and goes beyond the common range calculated by the training data, this cluster is considered as incomplete and the calculated reference point will not be used in the following tracking procedure.

Another cause for getting inaccurate position referencing point is the outliers (e.g., the point A in Veh3 cluster in Fig. 8) from the clustering process. An outlier filtering method that considers the distribution of the density of clustered points in 3D space is applied to reduce the error caused by outliers. In each cluster, the 3D clustered points are divided into a series of continuous 3D cubes, and the number of 3D points in each cube is counted as the density of that cube. A density threshold is then determined to judge whether a cube is an outlier cube or not. The selection of the threshold is based on the size of the cube and the total number of points in a cluster, in this study, a threshold of 5 points per 50 cm-edge cube was used for outlier filtering. Therefore, if a cube density is less than 5, the points in that cube are considered as outliers and deleted. Fig. 10 shows a vehicle cluster with some sparse points after the clustering process. The selected/reserved points were in small purple circles and the others were considered as outliers. Outlier filtering also enhances the performance for getting accurate reference points for positioning, thus improving the accuracy of classification and tracking in the following process. Both outlier filtering and the treatment for incomplete clusters can improve the accuracy of position reference points to some extent. However, the occlusion issue could not be resolved at 100 percent as this study only uses single LiDAR sensor. Better performance is expected after an ongoing implementation of multiple sensors at the same site.

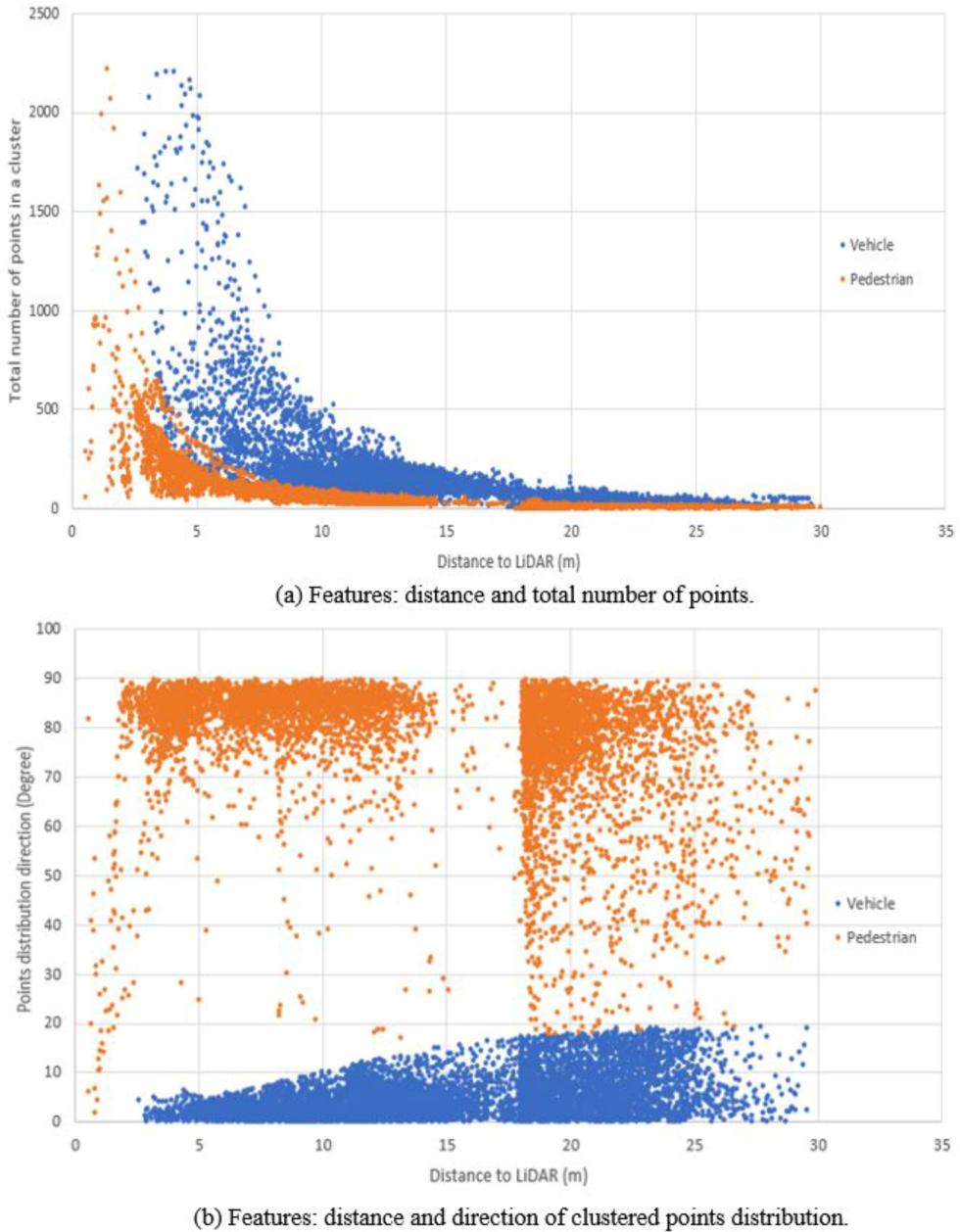
### 3.3. Pedestrian and vehicle classification

With clustering and referencing process described, attention can now be directed to pedestrian and vehicle classification. For this purpose, three features were extracted from the clusters obtained from the previous clustering process:

- (1) Total number of points in a cluster: at the same location, the number of points in a vehicle cluster should be more than that in a pedestrian cluster.
- (2) 2D Distance: the distance of the position reference point of each cluster to the location of the LiDAR sensor can be calculated with the X, Y values.
- (3) Direction of the clustered points' distribution: analysis of the distribution of the clustered points in the 3D space revealed that the distribution of pedestrian clustered points is mainly along the vertical direction (z-axis), while the distribution of vehicle clustered points is primarily along the horizontal direction (parallel to the x-y plane). With the least-square linear regression method, a linear function can be generated to describe the main distribution direction of each cluster.

Fig. 11 shows the relation among the three features. In Fig. 11(a), the number of points in one cluster decreases as the distance between the object and the LiDAR sensor increases. In general, vehicle clusters have more points than pedestrian clusters at the same distance. The difference between pedestrian clusters and vehicle clusters in terms of the direction of the clustered points' distribution can be observed from Fig. 11(b), when the direction angle of a cluster is less than 20°, this cluster has a high possibility to be classified into vehicle category. Among these three features, the total number of points in a cluster is affected by the distance from an object to the LiDAR sensor and occlusion, while the direction of distribution does not have a direct connection with the number of points and distance. Although only three features were identified, it is worthy of note that linear regression might not work in such a case as the boundary lines for classifying vehicles and pedestrians in Fig. 11 are not linear.

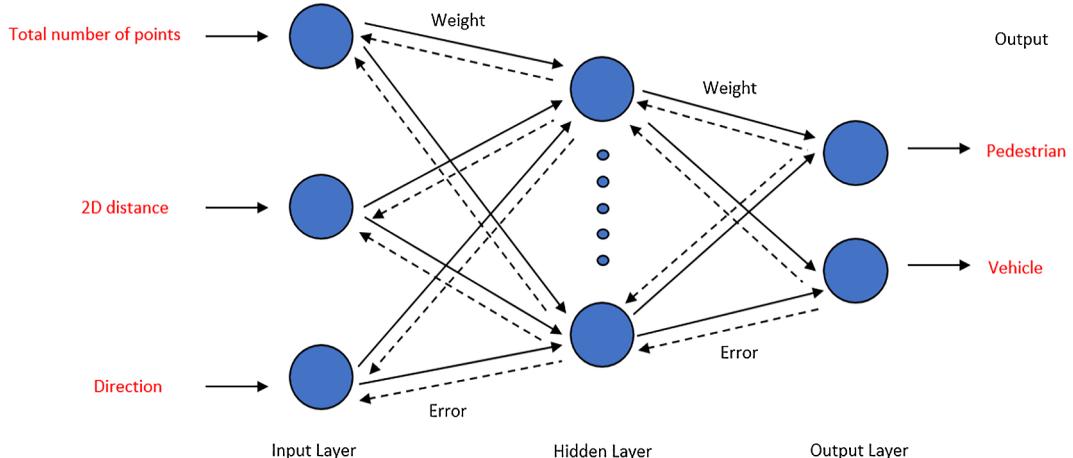
Taking these three features as inputs, a classification model based on backpropagation artificial neural network (BP-ANN) was developed to distinguish pedestrians and vehicles in the detection range. As shown in Fig. 12, the BP-ANN is a multilayer feed-forward neural network composed of an input layer, a hidden layer, an output layer, and neurons in each layer. Input data are fed into the input layer. The activity of each hidden layer is determined by the inputs and the weights that connect the input layer and hidden



**Fig. 11.** Extracted features for classification.

layer. A similar process takes place between the hidden layer and the output layer, and the transmission from one neuron in one layer to another neuron in the next layer is independent. The output layer produces the estimated outcomes. The comparison information (error) between the target outputs and the estimated outputs is given back to the input layer as a guidance to adjust the weights in the next training round. Through each iteration, the neural network gradually learns the inner relationship between the input and the output by adjusting the weights for each neuron in each layer to reach the best accuracy. When the minimum error is reached, or the number of iterations is beyond a predefined range, the training process is terminated with fixed weights.

The activation function, training function, transfer function, performance function all play important roles in model training. Training functions refer to different training algorithms; while the transfer function activates input signals to output signals, learning linear and non-linear relationships between them; and the performance function is used to measure the error. It is not intuitive to find the best combination of the structural parameters and the key functions for a neural network (influencing factors include data size, data complexity, network structure, network functionality, etc.). The case study section provides more information of the training process and details about how the BP-ANN model parameters and functions were determined.



**Fig. 12.** Structure of BP-ANN model.

### 3.4. Tracking

The last step in LiDAR data processing is tracking, in which the speed and trajectory of each object are obtained. Tracking is a process of identifying the same object in continuous data frames (Coifman et al., 1998; Alodi et al., 2016; Wang et al., 2017; Granström et al., 2018), in this research, a discrete Kalman filter tracking method was applied. The state vector  $\mathbf{x}_t$  is composed of four state parameters, which can be expressed as:

$$\mathbf{x}_t = (x_{0t} \ y_{0t} \ v_x \ v_y)' \quad (7)$$

where  $x_{0t}$  and  $y_{0t}$  are the horizontal and vertical coordinates of a cluster's reference point;  $v_x$  and  $v_y$  represent the horizontal and vertical speeds of the cluster's reference point. The measurement vector  $\mathbf{z}_t$  extracted from the clusters is summarized in the following form:

$$\mathbf{z}_t = (x_{0t} \ y_{0t})' \quad (8)$$

Besides, the transition matrix  $\mathbf{A}$  and measurement matrix  $\mathbf{H}$  can be described as:

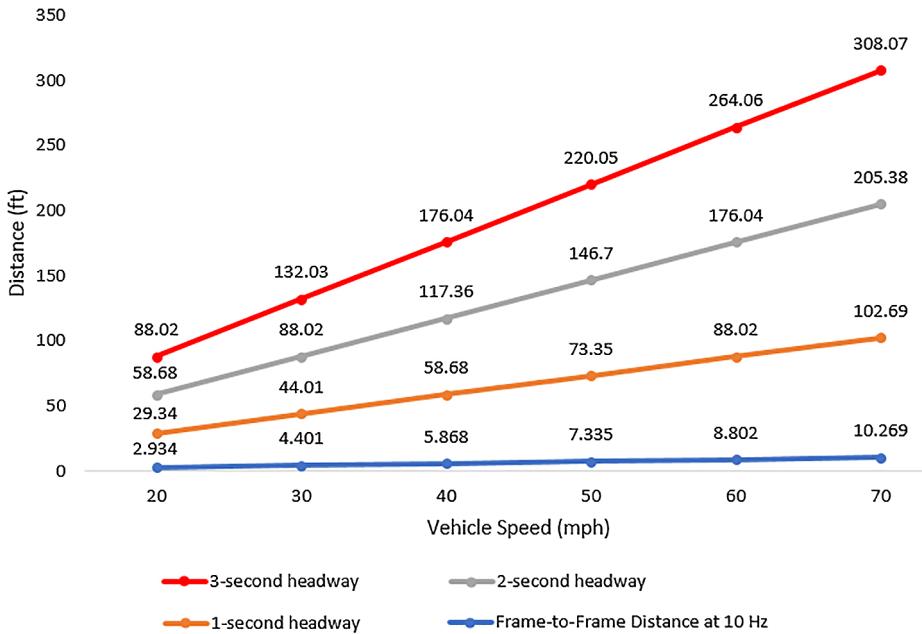
$$\mathbf{A} = \begin{pmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9)$$

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (10)$$

where  $t$  is the time interval between two adjacent frames ( $t \approx 0.1$  or  $0.2$  s in this research). For the Gaussian process noise and measurement noise, their corresponding uncertainty matrices ( $\mathbf{Q}$  and  $\mathbf{R}$ ) need to be adjusted empirically.

In the model, the position and speed of an object in the current frame are estimated based on its information from the previous frame. Then, an object association process is conducted within a searching area to find the associated objects from two consecutive frames, the size of the searching area is determined by the estimated position of the object. In the case when there are multiple candidate objects in the searching region, the object with the shortest distance will be used to match the object in the previous frame and update the measurement. Otherwise, object matching will depend on the time when next qualified candidate objects will be found, if no object can be found within the searching region after a certain time interval (e.g.,  $1.5$  s = 15 frames for a 10 Hz LiDAR sensor), the tracking of this target object stops. If the target object cannot be detected from a frame due to factors such as occlusion or failure of clustering, the predicted status is used as the position of the missing object to update the state vector and Kalman gain.

The shortest distance-based object association process is robust in that the actual distance a vehicle traveled in real world is always larger than that traveled along the frames. Fig. 13 shows the comparison of the frame-to-frame travel distance and the actual distance in the same lane. Assuming under different speeds there are three time-headways of 1, 2, 3 s respectively, the frame-to-frame distance at 10 Hz presents the travel distance of the same vehicle between the adjacent frames (recorded at 10 Hz) at different travel speeds. For example, for a vehicle traveling at 20 mph, the travel distance during one second headway is 29.34 ft, while the same vehicle only travels 2.934 ft ( $20 \text{ mph} * 0.1 \text{ s} = 2.934 \text{ ft}$ ) during one frame. The comparison shows that the travel distance between adjacent frames (at 10 Hz) is much shorter than the distance between vehicles in the same lane, which indicates that the object association based on the shortest distance works in this frame-to-frame mechanism. Although the chart is the comparison of the distance of vehicles in the same lane, similar vehicle headways are needed when vehicles change lanes. Therefore, this tracking method can be used to track the same vehicle efficiently in different frames.



**Fig. 13.** Comparison of the frame-to-frame travel distance and the distance between adjacent vehicles in the same lane.

With the trajectory of each road user successfully identified, their position, velocity, and direction can be obtained. Speed can be calculated by using the distance and time information of two consecutive frames, and the maximum allowable speed (speed limit in this study) was used to filter the outliers of the calculated velocity. Then, the accuracy of speed reporting was evaluated by a testing vehicle. To get the direction information in a polar coordinate system, the direction and distance from the cluster's position reference point to the LiDAR sensor (whose location is at the origin) were calculated and summarized in directional polar plots.

#### 4. Case study

With the help of the City of Reno and the Regional Transportation Commission (RTC) of Washoe, Nevada, three 16-channel LiDAR sensors (VLP-16) were installed at three intersections located at (39.543 N, -119.818 W), (39.538 N, -119.818 W), and (39.537 N, -119.816 W), respectively in the city of Reno, Nevada. [Table 1](#) summarizes the sites and the information associated with data collection, [Fig. 14](#) shows the map of the three data collection sites. The research team uses the data collected from the three sites extensively to investigate the properties of roadside LiDAR data, as well as develop and evaluate the methods and algorithms described in the previous sections.

##### 4.1. Evaluation of the clustering process

The modified DBSCAN method was evaluated against a wealth of data obtained from the three sites. For illustrative purpose, a set of data with four close-by pedestrians were used in this section. [Fig. 15](#) shows an example frame of the clustered points with three vehicles and nine pedestrians. As can be seen, the four close-by pedestrians (presented by red, green, black, and blue dots) were clearly clustered, although the space between pedestrians in green and black was only about 50 cm. In the analysis, the detection range was divided into three subareas:  $d_1 = 0\text{--}10\text{ m}$ ,  $d_2 = 8\text{--}25\text{ m}$ , and  $d_3 = 23\text{--}40\text{ m}$ , with an overlapped area of two meters in width. The MinPts values for the three subareas were tested with 30–60% (with a 1% interval) of the calculated TP values using field data. It was found that approximately 40% of the TP values (i.e., 40, 25, and 10 for the corresponding three subareas) could achieve the best clustering accuracy (i.e., 98.0%, 96.6%, and 94.8%, respectively). [Table 2](#) shows another example of the evaluation results from three sites, which makes comparison between the number of extracted clusters from LiDAR data and the recorded videos for 100 data sample frames. In average, the accuracy of the proposed clustering process achieves 96% from the data obtained from the three

**Table 1**  
Data collection site information.

Location	LiDAR rotation frequency (Hz)	Data collection period	Speed limit (mph)
N Virginia St@15th St (Site 1)	10	19,860 frames (33 min)	25
Sierra St@11th St (Site 2)	10	18,000 frames (30 min)	25
N Virginia St@10th St (Site 3)	5	18,000 frames (60 min)	25



**Fig. 14.** Google map of data collection sites.

sites, as discussed earlier, the error was mainly caused by distance and occlusion.

#### 4.2. Evaluation of the classification process

After clustering, the next step is to classify clusters into either vehicle or pedestrian datasets using the BP-ANN model. To train the ANN model, LiDAR data were recorded and reviewed, then they were divided into three groups for training (70%), validation (15%), and testing (15%). A testing dataset that has not been seen by the trained model was used to validate and evaluate the trained network.

Use the Site 1 data as an example: a total of 6800 clusters were randomly selected from a total of 19,860 frames data and divided into three sub-datasets, with 4800 clusters for training, 1000 clusters for validation, and 1000 clusters for testing. For the training data, the optimal number of neurons in the hidden layer needs to be determined first. An experiment was designed for this purpose, in which the number of neurons was changed from 1 to 20 and meanwhile the classification accuracy was recorded to find the best match. The results in Table 3 show that seven hidden neurons could give the best accuracy rate for both training and testing sessions (i.e., 97.2% and 96.5%, respectively). Once the optimal number of neurons was determined, the authors tested different training functions and transfer functions. It was found that the performance of the ANN model did not change significantly.

The scaled conjugate gradient backpropagation for training function, the sigmoid for activation function, the soft max transfer function, and the mean squared error performance function were selected as the functions of the ANN model. In general, the performance of the model will improve with more epochs/iterations of training, however, the error might start to increase on the validation set after a period of training due to the overfitting issue of the training set. Fig. 16 shows that the best performance for validation was achieved at epoch 14 with a 0.018 mean squared error. To summarize, the structure and functions of the ANN model are listed as follows:

- *feature selection*: total number of points, 2D distance, and direction
- *the number of hidden layers*: 1
- *the number of hidden neurons*: 7
- *training function*: scaled conjugate gradient backpropagation
- *activation function*: sigmoid
- *transfer function*: soft max transfer function

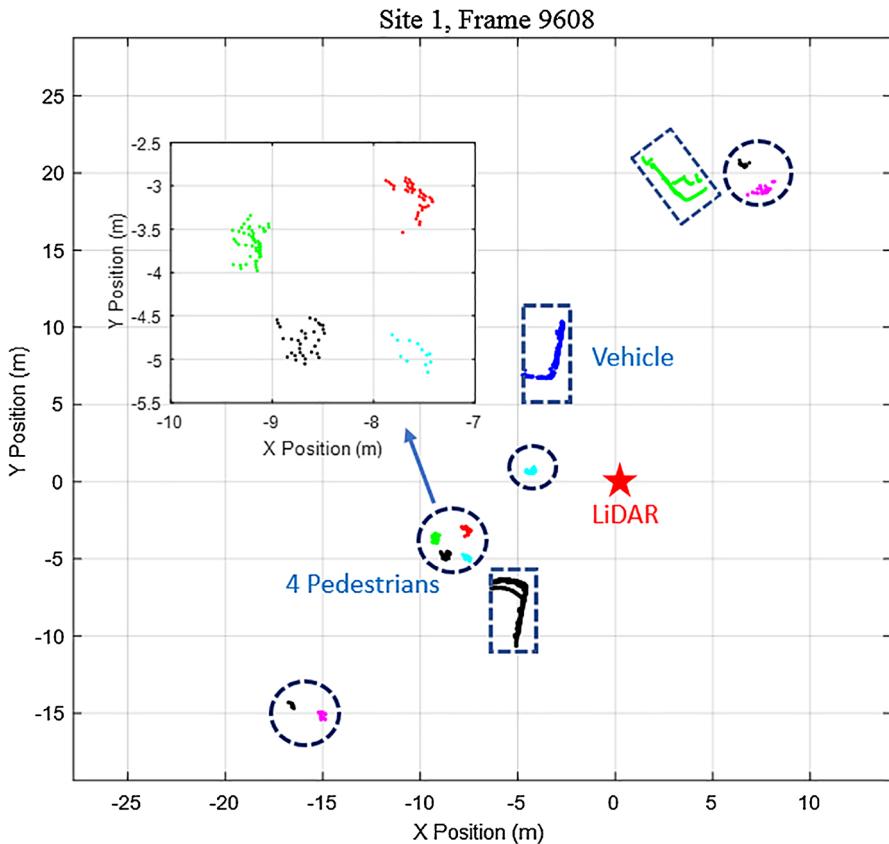


Fig. 15. Clustering results.

**Table 2**

Evaluation of the clustering process.

Location	Total frames	Clusters from the proposed algorithms	Clusters from the video	Detection accuracy
N Virginia St@15th St	100	315	305	96.8%
Sierra St@11th St	100	356	338	95.0%
N Virginia St@10th St	100	243	234	96.2%

- *performance function:* mean squared error performance function

**Table 4** demonstrates the performance of the classification process at the three sites. In general, an approximately 96% classification accuracy can be guaranteed within about 30 m detection range (in one direction) from the LiDAR sensor, at this range, failure was mainly caused by vehicle occlusion.

#### 4.3. Evaluation of the tracking process

For tracking, a total of 1023 vehicles and 48 pedestrians were detected and tracked from 19,860 frames data collected at Site 1. Fig. 17 presents part of the extracted vehicle and pedestrian trajectories, in which the red and blue dots represent the trajectories of vehicles and pedestrians at a 10 Hz frequency. As can be seen, all movements including the through/left-turn/right-turn movements of vehicles, and the crossing movement of pedestrians were clearly tracked. Table 5 shows the tracking results from 1000 data sample frames at three sites, approximately an accuracy of 95% can be reached within about 30 m detection range (in one direction) from the LiDAR sensor, and failure was caused by the same issues such as occlusion and distance.

#### 4.4. Evaluation of speed

Speed validation was conducted by a testing vehicle with an onboard diagnostics (OBD) logger. Worthy of note is the frequency of LiDAR data is 10 Hz, while the OBD speed can only be measured at 2 Hz, indicating the speed can be calculated every 0.1 s by LiDAR

**Table 3**  
Determine the number of neurons for ANN model.

Number of neurons	Training accuracy	Testing accuracy
1	95.2%	93.8%
2	95.5%	93.5%
3	94.0%	94.0%
4	96.7%	96.2%
5	94.8%	93.7%
6	95.8%	94.5%
7	<b>97.2%</b>	<b>96.5%</b>
8	96.8%	95.3%
9	96.3%	94.7%
10	96.5%	94.8%
11	96.7%	94.8%
12	94.5%	93.7%
13	94.3%	94.2%
14	96.3%	94.5%
15	96.0%	95.3%
16	94.7%	95.8%
17	94.3%	94.3%
18	95.8%	93.3%
19	96.8%	94.7%
20	96.2%	94.7%

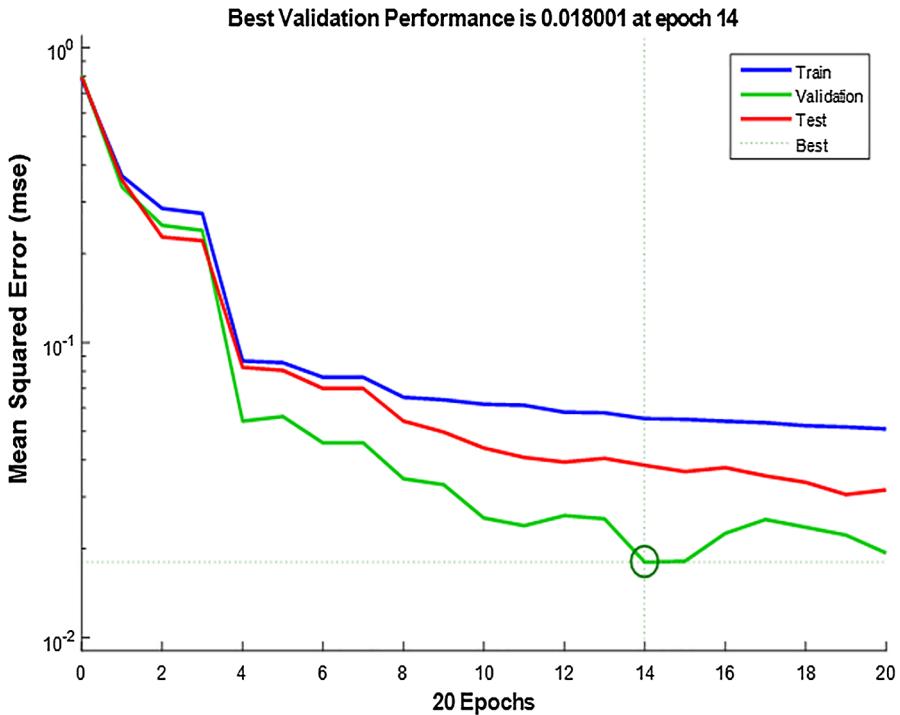


Fig. 16. Error convergence curve for BP-ANN model.

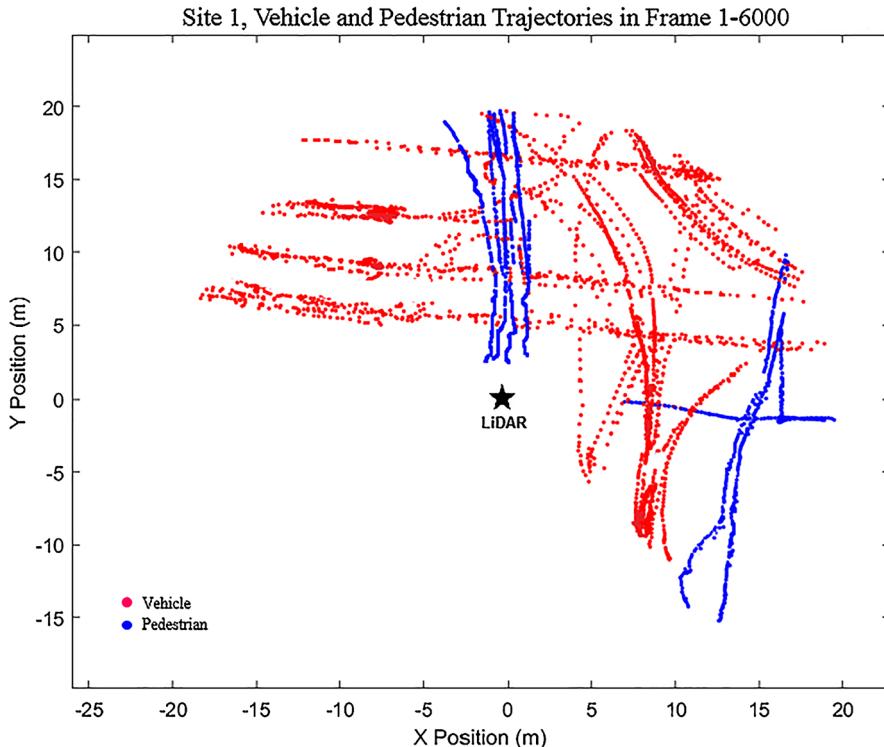
data and the OBD logger could only update every 0.5 s. Therefore, validation was conducted by comparing the space mean speed calculated from the LiDAR data (in every 0.5 s) with the speed measured by the OBD logger. Fig. 18 shows the basic statistics of speed validation: the testing car was running at a relatively stable speed between 25 and 35 mph, the orange bars show the speeds obtained from the OBD logger, and the blue points depict the calculated speeds from the LiDAR data. Our investigation found the error was mainly caused by: (1) the onboard logger cannot provide accurate speed information at a high frequency; (2) speed calculated from the LiDAR sensor may not be accurate due to the timestamp and position errors between frames.

Occlusion issue is always a problem exists in the real situations. The LiDAR location is higher than most light-weight vehicles (passenger cars), so sensors can at least scan the top of vehicles if they are blocked by passenger cars. Complete occlusion normally happens with trucks and buses. Our solution is to install multiple LiDAR sensors at different corners of intersections and along both sides of roads, thus providing better quality LiDAR data of all road users. Fig. 19 shows an example of partial and full occlusion cases.

**Table 4**

Evaluation of object classification.

Location	N Virginia St@15th St	Sierra St@11th St	N Virginia St@10th St
Total clusters	1000	1000	1000
Pedestrians (P)	403	389	284
Vehicles (N)	597	611	716
Identified Pedestrians (TP)	385	370	275
Identified Vehicles (TN)	581	593	690
Unidentified Pedestrians (FP)	16	18	26
Unidentified Vehicles (FN)	18	19	9
Pedestrian Identification Rate (TP/TP + FN)	95.5%	95.1%	96.8%
Vehicle Identification Rate (TN/TN + FP)	97.3%	97.1%	96.3%
Classification Accuracy (TP + TN/P + N)	96.6%	96.3%	96.5%

**Fig. 17.** Example of extracted vehicle and pedestrian trajectories.**Table 5**

Evaluation of tracking.

Location	Total frames	Trajectories from the proposed algorithm	Trajectories from the video	Tracking accuracy
N Virginia St@15th	1000	40	42	95.2%
Sierra St@11th St	1000	35	37	94.6%
N Virginia St@10th St	1000	34	35	97.1%

After data registration, the second pedestrian in Fig. 19(a) can be seen more clearly. In Fig. 19(b), the second vehicle was totally blocked by the first vehicle when using one LiDAR sensor, but it can be detected using two LiDAR sensors at different locations. Therefore, the LiDAR sensor network will extend the detection range and reduce the possibility of objects being blocked.

## 5. Conclusion and perspective

This paper presents a systematic approach to the detection and tracking of both pedestrians and vehicles at intersections by using infrastructure-based LiDAR sensors. The roadside LiDAR data processing procedure includes background filtering, object clustering, pedestrian/vehicle classification, object tracking, and finally obtaining the presence, position, velocity, and direction information of

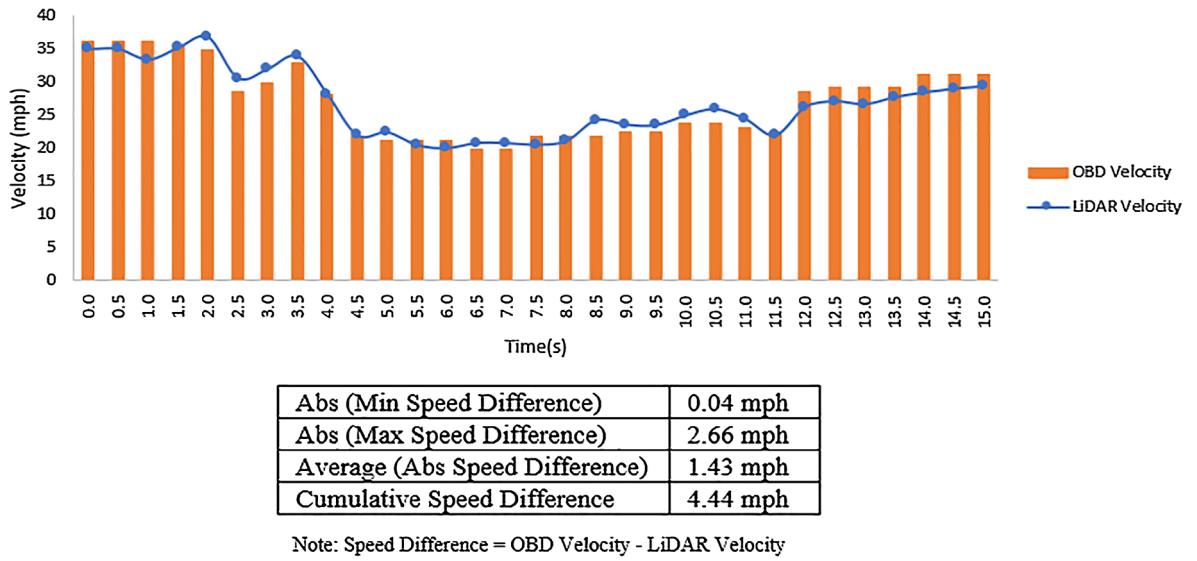


Fig. 18. Speed validation.

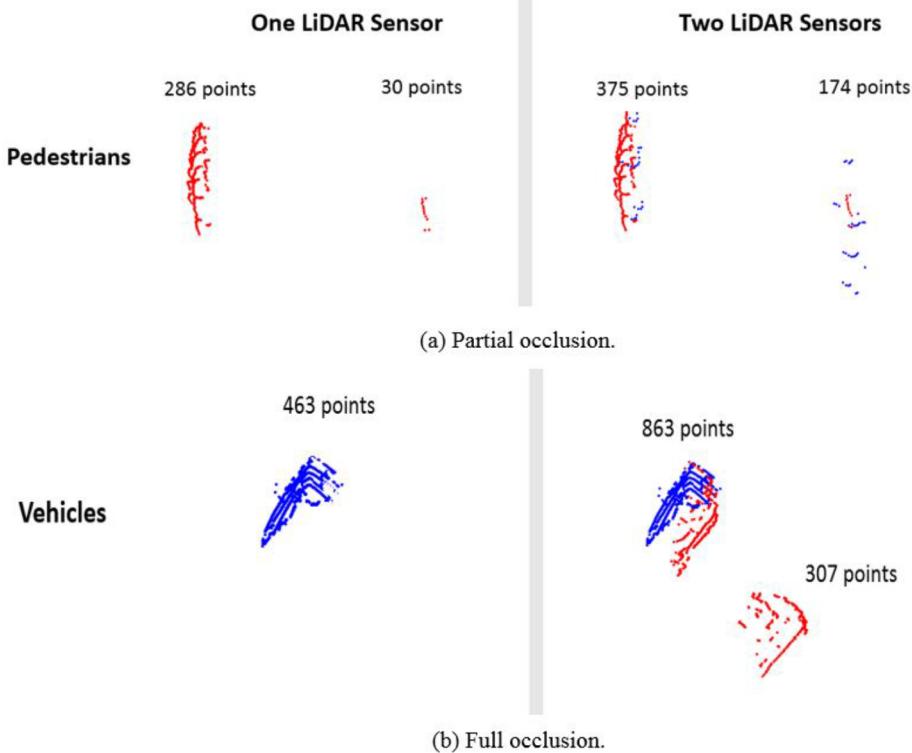


Fig. 19. Occlusion issue.

pedestrians and vehicles. Modified DBSCAN method and ingenious design of area division improve the accuracy of object clustering with LiDAR data, as well as reduce computational expense. Three straightforward but effective features are extracted for classification. Direction of clustered points' distribution is the key feature to distinguish pedestrian and vehicle clusters when the collected LiDAR data cannot give a fine description of objects. A comprehensive case study at three sites showed the effectiveness of the proposed algorithms using the real data collected by a 16-laser LiDAR sensor deployed at roadside. The detection and tracking rates of the proposed roadside LiDAR data processing procedure are all above 95%, and valid detection range is about 30 m (in one direction). Although the procedure was tested with a 16-laser LiDAR, it can be extended to other types of LiDAR sensors. For example, the developed clustering method can be modified with new sensor's configuration such as the number of lasers, the vertical resolution

of laser beams, and sensor's rotation frequency to determine the parameters. Using more advanced LiDAR sensors (e.g., 64- or 128-laser), the performance of background filtering, clustering, and classification can be enhanced since the density of point clouds is increased, thus the tracking accuracy and detection range can be improved to some extent. For basic LiDAR sensors (e.g., 1- or 8-laser), the proposed procedure can also apply to their data. But the effectiveness and accuracy of object detection and tracking may be reduced.

Due to the limited number of lasers of 16-laser LiDAR sensors, the density of point clouds is lower compared with data collected by advanced LiDAR sensors. Features that represent specific characteristics of objects are difficult to extract, so that the types of vehicles cannot be distinguished using the proposed methods. The accuracy of position reference points is also affected by incomplete clusters due to occlusion, perspective shadow, etc. Discrete Kalman filtering tracking is a kind of ad-hoc solution, which needs to be improved in the future. There is no calibration involved in the data processing procedure. In addition, without high-accuracy GPS receivers, in-depth analysis such as statistic test of speed validation and precise position validation cannot be carried out at current stage.

For future work, multiple LiDAR sensors will be deployed at intersections and along roads in order to cover a larger area and reduce the possibility of occlusion. The proposed procedure and algorithms will be further validated and improved for accuracy and reliability of various traffic scenarios, as well as tested using LiDAR sensors with different resolutions. Methods of object detection and tracking can be extended to bicycles in urban areas and wildlife in rural areas.

## Acknowledgement

This work was supported by the SOLARIS Institute, a Tier 1 University Transportation Center (UTC) [Grant No. DTRT13-G-UTC55] and matching funds by the Nevada Department of Transportation (NDOT) [Grant No. P224-14-803/TO #13]. The authors gratefully acknowledge this financial support. This research was also supported by engineers with the Nevada Department of Transportation, the Regional Transportation Commission of Washoe County, Nevada, and the City of Reno. We are very thankful to the reviewers for their time and efforts, their comments and suggestions have greatly improved the quality of this paper.

## References

- Aggarwal, C.C., Reddy, C.K., 2013. *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC.
- Ai, C., Tsai, Y.J., 2016. An automated sign retroreflectivity condition evaluation methodology using mobile LiDAR and computer vision. *Transport. Res. Part C: Emerg. Technol.* 63, 96–113. <https://doi.org/10.1016/j.trc.2015.12.002>.
- Allodi, M., Broggi, A., Giaquinto, D., Patander, M., Piroletti, 2016. A Machine learning in tracking associations with stereo vision and lidar observations for an autonomous vehicle. In: Intelligent Vehicles Symposium (IV), pp. 648–653. <https://doi.org/10.1109/IVS.2016.7535456>.
- Azim, A., Aycard, O., 2012. Detection, classification and tracking of moving objects in a 3D environment. In: Intelligent Vehicles Symposium (IV), pp. 802–807. <https://doi.org/10.1109/IVS.2012.6232303>.
- Babaei, M., Dinh, D.T., Rigoll, G., 2018. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognit.* 76, 635–649. <https://doi.org/10.1016/j.patcog.2017.09.040>.
- Chavez-Garcia, R.O., Aycard, O., 2016. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Trans. Intell. Transp. Syst.* 17 (2), 525–534.
- Cheng, J., Xiang, Z., Cao, T., Liu, J., 2014. Robust vehicle detection using 3D Lidar under complex urban environment. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 691–696. <https://doi.org/10.1109/ICRA.2014.6906929>.
- Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transport. Res. Part C: Emerg. Technol.* 6 (4), 271–288. [https://doi.org/10.1016/S0968-090X\(98\)00019-9](https://doi.org/10.1016/S0968-090X(98)00019-9).
- Du, X., Ang, M.H., Rus, D., 2017. Car detection for autonomous vehicle: LiDAR and vision fusion approach through deep learning framework. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 749–754. <https://doi.org/10.1109/IROS.2017.8202234>.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, pp. 226–231.
- Gao, H., Cheng, B., Wang, J., Li, K., Zhao, J., Li, D., 2018. Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment. *IEEE Trans. Ind. Inf.* <https://doi.org/10.1109/TII.2018.2822828>.
- Granström, K., Svensson, L., Reuter, S., Xia, Y., Fatemi, M., 2018. Likelihood-based data association for extended object tracking using sampling methods. *IEEE Trans. Intell. Veh.* 3 (1), 30–45. <https://doi.org/10.1109/TIV.2017.2788184>.
- Himmelsbach, M., Müller, A., Luettel, T., Wünsche, H.-J., 2008. LiDAR-based 3D object perception. *Proceedings of 1st International Workshop on Cognition for Technical Systems*, vol. 1.
- The New York Times, 2018. How a self-driving Uber killed a pedestrian in Arizona. Available at: <https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestrian-killed.html>.
- Ismail, K., Sayed, T., Saunier, N., Lim, C., 2009. Automated analysis of pedestrian-vehicle conflicts using video data. *Transport. Res. Rec.: J. Transport. Res. Board* 2140, 44–54. <https://doi.org/10.3141/2140-05>.
- Lee, H., Coifman, B., 2012. Side-fire lidar-based vehicle classification. *Transport. Res. Rec.: J. Transport. Res. Board* 2308, 173–183. <https://doi.org/10.3141/2308-19>.
- Li, K., Wang, X., Yu, X., Wang, J., 2016. Density enhancement-based long-range pedestrian detection using 3-D range data. *IEEE Trans. Intell. Transport. Syst.* 17 (5), 1368–1380.
- LiDAR, Velodyne, 2016. *VLP-16. VLP-16 Manual: User's Manual and Programming Guide*. Velodyne LiDAR, Inc., San Jose, CA, USA.
- Miyasaka, T., Ohama, Y., Ninomiya, Y., 2009. Ego-motion estimation and moving object tracking using multi-layer lidar. In: Intelligent Vehicles Symposium, pp. 151–156. <https://doi.org/10.1109/IVS.2009.5164269>.
- Morsdorf, F., Meier, E., Kötz, B., Itten, K.I., Dobbertin, M., Allgöwer, B., 2004. LiDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sens. Environ.* 92 (3), 353–362. <https://doi.org/10.1016/j.rse.2004.05.013>.
- Mukhtar, A., Xia, L., Tang, T.B., 2015. Vehicle detection techniques for collision avoidance systems: a review. *IEEE Trans. Intell. Transp. Syst.* 16 (5), 2318–2338.
- Premebida, C., Ludwig, O., Nunes, U., 2009. Exploiting lidar-based features on pedestrian detection in urban scenarios. In: Intelligent Transportation Systems Conference (ITSC), pp. 1–6. <https://doi.org/10.1109/ITSC.2009.5309697>.
- Premebida, C., Monteiro, G., Nunes, U., Peixoto, P., 2007. A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In: Intelligent Transportation Systems Conference (ITSC), pp. 1044–1049.
- Sakkos, D., Liu, H., Han, J., Shao, L., 2017. End-to-end video background subtraction with 3d convolutional neural networks. *Multimedia Tools Appl.* 1–19. <https://doi.org/10.1007/s11042-017-5460-9>.
- Shackleton, J., VanVoorst, B., Hesch, J., 2010. Tracking people with a 360-degree lidar. In: IEEE International Conference on Advanced Video and Signal Based

- Surveillance (AVSS), pp. 420–426. <https://doi.org/10.1109/AVSS.2010.52>.
- Sivaraman, S., Trivedi, M.M., 2013. Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intell. Transp. Syst.* 14 (4), 1773–1795.
- Stauffer, C., Grimson, W.E.L., 2000. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8), 747–757. <https://doi.org/10.1109/34.868677>.
- Theodoridis, S., Koutroumbas, K., 2009. *Pattern Recognition*, fourth ed. Academic Press.
- Tonini, M., Abellan, A., 2014. Rockfall detection from terrestrial LiDAR point clouds: a clustering approach using R. *J. Spatial Inf. Sci.* 2014 (8), 95–110. <https://doi.org/10.5311/JOSIS.2014.8.123>.
- Wang, D.Z., Posner, I., Newman, P., 2012. What could move? Finding cars, pedestrians and bicyclists in 3d laser data. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4038–4044. <https://doi.org/10.1109/ICRA.2012.6224734>.
- Wang, H., Wang, B., Liu, B., Meng, X., Yang, G., 2017. Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robot. Autonom. Syst.* 71–78. <https://doi.org/10.1016/j.robot.2016.11.014>.
- Wojke, N., Häselich, M., 2012. Moving vehicle detection and tracking in unstructured environments. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3082–3087. <https://doi.org/10.1109/ICRA.2012.6224636>.
- Wu, J., Xu, H., Sun, Y., Zheng, J., Yue, R., 2018a. Automatic background filtering method for roadside LiDAR data. *Transport. Res. Rec.: J. Transport. Res. Board* 2018. <https://doi.org/10.1177/0361198118775841>.
- Wu, J., Xu, H., Zhao, J., 2018b. Automatic lane identification using the roadside LiDAR sensors. *IEEE Intell. Transp. Syst. Mag.* <https://doi.org/10.1109/MITS.2018.2876559>.
- Wu, J., Xu, H., Zheng, J., 2017. Automatic background filtering and land identification with roadside LiDAR data. In: *20th International Conference on Intelligent Transportation (ITSC)*, pp. 1–6. <https://doi.org/10.1109/ITSC.2017.8317723>.
- Zangenehpour, S., Miranda-Moreno, L.F., Saunier, N., 2015. Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic: methodology and application. *Transport. Res. Part C: Emerg. Technol.* 56, 161–176. <https://doi.org/10.1016/j.trc.2015.04.003>.
- Zhao, H., Sha, J., Zhao, Y., Xi, J., Cui, J., Zha, H., Shibasaki, R., 2012. Detection and tracking of moving objects at intersections using a network of laser scanners. *IEEE Trans. Intell. Transp. Syst.* 13 (2), 655–670. <https://doi.org/10.1109/TITS.2011.2175218>.
- Zhao, H., Shibasaki, R., 2005. A novel system for tracking pedestrians using multiple single-row laser-range scanners. *IEEE Trans. Syst. Man Cybernet.–Part A: Syst. Hum.* 35 (2), 283–291.
- Zhao, J., Xu, H., Wu, J., Lin, C., Liu, H., 2019. Detection range analysis of roadside LiDAR sensors. *Proceedings of the 98th Transportation Research Board Annual Meeting*, January, Washington, D.C.