Architecture of Vehicle Trajectories Extraction with Roadside LiDAR Serving Connected Vehicles

Jingrong Chen¹, Sheng Tian², Hao Xu³, Rui Yue³, Yuan Sun⁴, and Yuepeng Cui⁵

¹School of Mathematics and Physics, Lanzhou Jiaotong University, Lanzhou 730070, China
²School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, China
³Department of Civil and Environmental Engineering, University of Nevada, Reno, NV 89557, USA
⁴Business School, Harvard University, Boston, NV 02163, USA
⁵Department of Transportation, Fujian University of Technology, Fuzhou 350118, China

Corresponding author: Jingrong Chen (chenjr@mail.lzjtu.cn) and Rui Yue (ruiyueunr@gmail.com)

This paper was typed from Yuan Sun’s doctoral dissertation–3-D Data Processing to Extract Vehicle Trajectories from Roadside LiDAR Data. This work was supported by the Natural Science Foundation of China under Grant 61463026 and Grant 61463027.

ABSTRACT This research developed a data processing procedure for detection and tracking of multi-lane multi-vehicle trajectories with a roadside Light Detection and Ranging (LiDAR) sensor. Different from existing perception methods for the autonomous vehicle system, this procedure was explicitly developed to extract trajectories from a roadside LiDAR sensor. The proposed procedure includes five main steps: region of interest (ROI) selection, ground surface filtering, point clustering, vehicle/non-vehicle classification, and geometrical vehicle tracking. The case study showed that the trajectories of vehicles can be generated with the proposed method. This research is the start of the new-generation connected infrastructures serving connected/autonomous vehicles with the roadside LiDAR sensors. It will accelerate the deployment of connected-vehicle technologies to improve traffic safety, mobility and fuel efficiency.

INDEX TERMS Connected-Vehicle, Vehicle Trajectory, Roadside LiDAR

I. INTRODUCTION

Connected-Vehicles (CV) technology, representing the next generation intelligent transportation systems (ITS), has been promoted by the Federal Highway Administration for several years [1]. In a connected-vehicle (CV) system, the road users can know the traffic information in advance with the help of other road users connected in the same network through various wireless communication methods. The CV network relies on the communication of different roadside and on-board sensors, so all the road users can share their real-time information. However, the current CV application is constrained by the resolution of data input [2]. The CV network requires high-resolution micro traffic data (HRMTD), which means second-by-second real-time traffic data of all individual road users [3, 4]. Since it takes time to build a whole connected system (especially vehicle-to-vehicle communication), supplemental data provided by the roadside infrastructures are required to help the deployment of the CV network [5]. The challenge lies on how to obtain the HRMTD required by the CV network from the roadside infrastructures. The traditional traffic sensors such as loop detectors, video detectors, and radar sensors mainly provide macro traffic data such as traffic flow rates, average speeds, and occupancy. The performance of those traditional traffic sensors can be influenced by different factors, such as light conditions, limited detection range, et al [6]. Therefore, the existing sensors cannot provide the HRMTD needed for connected vehicles. The Light Detection and Ranging (LiDAR)-enhanced roadside infrastructure can be a good option for the HRMTD collection. The traditional LiDAR sensor can transmit and receive electromagnetic radiation at a relatively high frequency (usually 5–20Hz). The new 360-degree LiDAR sensor can detect the objects by creating the point cloud within a specific range, which provides opportunities for the HRMTD collection [7]. The performance of mainstream traffic sensors [8] was summarized in Table 1. It is shown that compared to other sensors, the LiDAR sensor works better in general.

The roadside LiDAR-enhanced infrastructure is a new application of the 360-degree (rotating) LiDAR. Considering the massive deployment of roadside LiDAR in the future, the roadside LiDAR has to provide the HRMTD in an extended range compared to the on-board LiDAR serving autonomous vehicles. The challenge here is how to extract the HRMTD from the roadside LiDAR. By now, a lot of algorithms have been conducted for object tracking using the LiDAR, most of which were developed for airborne LiDAR and on-board LiDAR [9]. Asvadi et al. [10] developed a method for object tracking using the videos and LiDAR data. Two parallel mean-shift algorithms were applied for object detection. The object was further tracked using a robust 2D/3D Kalman filter based fusion. However, this method developed for autonomous vehicles can only detect the objects with good shape, which means the
detection range was limited. Wang et al. [11] developed a model-free method for dynamic object detection and tracking using 2D LiDAR. This method lacks the semantic interpretation of the tracking objects. Dewan et al. [12] developed a RANSAC and Bayesian based approach to track the objects with LiDAR. However, the object type, vehicle or pedestrian, was not reported from that algorithm. Wang et al. [13] developed a Support Vector Machine (SVM) to classify pedestrians and vehicles on the road, and used the position information for pedestrian tracking. The algorithm relied on the high-resolution LiDAR sensor. The performance of the algorithm on cost-effective LiDARs (such as VLP-16) was unknown. Hwang et al. [14] used a DBSCAN for object clustering and used a fusing approach for multiple objects detection and tracking using 3D LiDARs and cameras. Qin et al. [15] developed a spatial-temporal approach for object recognition with 2D LiDAR. Pino et al. [16] used a Convolutional Neural Network (CNN) to detect vehicles based on a LiDAR-based system for autonomous vehicles. High accuracy was observed through case studies using real-world data. Those previous efforts provided good references for HRMTD collection using LiDAR. However, those methods in fact could not be directly applied for high-resolution vehicle trajectory using roadside LiDAR for CV network. Firstly, the roadside LiDAR sensor should be able to provide the high-resolution vehicle trajectories without the help of other optical sensors considering the huge maintenance labor in the future. Secondly, the roadside LiDAR sensor needs to detect all the road users in 360-degrees in the horizontal direction, to detect all the HRMTD of all road users. How to extract the vehicle trajectories with high-resolution from the raw roadside LiDAR data is a challenge left for researchers and traffic engineers.

This paper presents a systematic data processing method for HRMTD collection with the roadside LiDAR. The performance of the proposed method was evaluated using the real-world collected data. The following parts of the paper are structured as follows: Section 2 documents the proposed method, Section 3 evaluates the proposed method through different case studies, the discussion of this paper is documented in Section 4, Section 5 summarizes the conclusions of this research.

### TABLE I

**STRENGTHS AND WEAKNESSES OF MAINSTREAM SENSOR TECHNOLOGIES**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inductive Loop</td>
<td>• Flexible design to satisfy a large variety of applications.</td>
<td>• Installation requires pavement cut.</td>
</tr>
<tr>
<td></td>
<td>• Mature, well-understood technology.</td>
<td>• Maintenance requires lane-closure.</td>
</tr>
<tr>
<td></td>
<td>• Large experience base.</td>
<td>• Multiple detectors are usually required for monitoring one location; the detection range is short.</td>
</tr>
<tr>
<td></td>
<td>• Only provides macro traffic parameters (e.g., volume, presence, occupancy, speed, headway, and gap).</td>
<td>• CW Doppler sensors cannot detect stopped vehicles or stopped pedestrians.</td>
</tr>
<tr>
<td></td>
<td>• Insensitive to inclement weather such as rain, fog, and snow.</td>
<td>• Not cover 360 degrees range.</td>
</tr>
<tr>
<td>Microwave Radar</td>
<td>• Typically, Radar is insensitive to inclement weather at the relatively short ranges encountered in traffic management.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Multiple lane operations are available.</td>
<td></td>
</tr>
<tr>
<td>Video Image Sensor</td>
<td>• Monitors multiple lanes and multiple detection zones/lane.</td>
<td>• Installation and maintenance, including periodic lens cleaning; requiring lane closure when the camera is mounted over the roadway.</td>
</tr>
<tr>
<td></td>
<td>• Easy to add and modify detection zones.</td>
<td>• Performance affected by inclement weather such as fog, rain, and snow; vehicle shadows; occlusion; day-to-night transition; vehicle/road contrast.</td>
</tr>
<tr>
<td></td>
<td>• A rich array of data available.</td>
<td>• Some models are susceptible to camera motion caused by strong winds or vibration of camera mounting structure.</td>
</tr>
<tr>
<td></td>
<td>• Provides wide-area detection when information gathered at one camera location can be linked to another.</td>
<td>• Reliable nighttime signal actuation needs excellent street lighting.</td>
</tr>
<tr>
<td>LiDAR</td>
<td>• Able to transmits multiple beams for accurate measurement of vehicle position, speed, and types.</td>
<td>• The operation may be affected by colossal rain or snow.</td>
</tr>
<tr>
<td></td>
<td>• Provide high-resolution data, extremely accurate.</td>
<td>• Performance might be affected by fog.</td>
</tr>
<tr>
<td></td>
<td>• Could cover a large area quickly with 360 degrees view.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Easy to install and maintain.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Work day and night without the influence of light conditions.</td>
<td></td>
</tr>
</tbody>
</table>

### II. METHOD

The data processing procedure proposed in this research includes five modules: region of interest (ROI) selection, ground surface filtering, point clustering, vehicle/non-vehicle classification, and geometrical vehicle tracking. The high-resolution vehicle trajectories can be generated at the end using the proposed approach. The flow chart of the proposed method is shown in Figure 1.

#### A. ROI SELECTION

Filtering of the irrelevant object would decrease the data size for processing. This step can increase the efficiency of the follow-up algorithms by searching fewer points. Since objects out of the road boundary are irrelevant information for extracting vehicle trajectories, those points (i.e. trees, buildings, and other irrelevant objects) out of road boundary were excluded directly by connecting the control points on
the road boundary manually [17]. The road boundary can be identified and visualized by aggregating different LiDAR frames in the space. The road boundary was detected using the density-based method developed by Wu et al. [18]. The brief introduction about the road boundary identification method was introduced as follows. The road boundary identification method starts with a background filtering algorithm. The point cloud created by the roadside LiDAR includes both road users and other irrelevant points called background points in the space. Those irrelevant points may include ground points, trees, buildings, and other noises. It is difficult to identify the road boundary when so many background points exist in the space. An unsupervised machine learning method, 3D-density-statistic-filtering (3D-DSF), was used for background exclusion [19]. The 3D-DSF can be illustrated in four parts: frame overlying, sub-space division, threshold search, and background storage. Frame overlying is to aggregate the points created by the roadside LiDAR in multiple frames. For one frame, the background points and the road users could not be distinguished since they may have a similar density. By aggregating multiple frames, the density of the background points can be higher than the moving road users [20]. The recommended number of frames are between 1500–3500 considering the accuracy and the computational load [21]. The whole space can then be chopped into sub-spaces (cubes) by given a side length. The sum of point density over all frames of each cube can be derived. The assumption of 3D-DSF is that the sum point density of the background cube will be much smaller than that of the cube with road users. An automatic threshold learning based on the point density distribution within different distances from the roadside LiDAR can be found in the reference [19]. With the thresholds, the cube can be identified as background or non-background. Those cubes representing the background can then be stored in a 3D array with their XYZ information. The 3D array can be considered as a background index, any points found in the index are excluded from the space. It should be mentioned that due to the vibration of the LiDAR sensor, the ground points usually could not be fully excluded. After excluding the background, the points representing the road users were aggregated by applying the second-time frame overlying. The boundaries of the aggregated points can then be identified. The objects out of the boundaries were excluded from the space directly since they were out of the ROI. Usually, the data can be reduced to less than 10% of its original size [22].

B. GROUND SURFACE FILTERING
Within the ROI, there are three different types of points: ground points, road users, and outliers. The ground points and outliers are considered as noises and must be excluded from the space. This step is to exclude the ground points from the space. This research applied a hybrid approach for ground surface removal [23]. The hybrid approach can be illustrated in two parts: channel-based clustering and slope-based filtering. For the channel-based clustering, by giving the dynamic thresholds for different LiDAR beams, a rough range of ground surface can be extracted. However, this step may include the root (lower layers) of some road users in the results. The slope-based filtering assumes that the slope of the ground surface is less than the slope of vehicle points in the z-axis direction. Assuming there are two adjacent channels: Ci and Ci+1, the closest point pair between the two channels can be represented as [24].

\[
\{A, B\} = \{a, b\}\min\{|(Xa - Xb)(Ci - Ci + 1) + (Ya - Yb)(Ci - Ci + 1)^2 + (Za - Zb)(Ci + 1)^2\}\]

Where \(Xa, Ya, Za,\) and \(Xb, Yb, Zb\) represent the XYZ coordinates of \(a\) and \(b\), respectively; \(a\) and \(b\) represent two random points in two adjacent channels. The slope between \(A\) and \(B\) can be represented in

\[
\sin(\alpha) = \frac{\sqrt{((Xa - Xb)^2 + (Ya - Yb)^2 + (Za - Zb)^2)}}{Za - Zb}
\]

In this research, the height of the roadside LiDAR placement was 7ft above the ground. 45-degrees is selected as the threshold of the slope for ground surface removal. If \(\alpha\) is higher than 45-degrees, then points in \(Ci+1\) will be removed from the space. Through this step, vehicle points can be distinguished from the ground surface, but the outliers still exist in the space.

C. POINT CLUSTERING
The outliers are usually created due to the mechanical reasons of the LiDAR or the interference from other sensors having the light with the same frequency band as the LiDAR. To further exclude the outliers and group the points belonging to one object, the DBSCAN a density-based spatial clustering of applications with noise method was applied. The reason for selecting DBSCAN for clustering is that the DBSCAN does not need to predefined the number of clusters, which is very useful in transportation applications [17]. The DBSCAN can be described in the code in Figure 2.
FIGURE 2 DBSCAN

Two key parameters, minimum number of points (MinPts) and epsilon (ε), are required for the DBSCAN. MinPts specifies how many points should be included in a cluster and ε defines the searching distance for a cluster. Based on the features of the point density and the mechanical properties of LiDAR, the selected values of MinPts and ε were 1.2m and 10m, respectively [21]. The previous practice [25] showed that the DBSCAN can achieve an accuracy of more than 97% for object clustering.

D. OBJECT CLASSIFICATION

How to differentiate vehicles/non-vehicles is another crucial task in this research. Object classification includes three major steps: feature selection, classifier training, and classification evaluation. Features, including local features and global features, are important for the classification algorithms. For the LiDAR points, it is difficult to use local features or global features to describe the objects due to the continuous change of the shape of cloud points. In this research, both global features and local features are used for the object description. The features were summarized as follows.

a. Average X,Y,Z coordinates of the clustering recorded as X, Y, Z

b. Nearest distance between the points and the LiDAR recorded as ND

c. Object Length (L) and Width (W). Since the LiDAR sensor may only scan part of the vehicle in a frame, the minimum-volume Oriented Bounding Box (OBB) was defined to represent the shape of the object. In this research, a min-max principal component analysis (PCA) based OBB method was used to identify vehicle boundaries. This method first calculates the principle or eigenvectors, and then a reference coordinate system is constructed based on these eigenvectors. The original vehicle cluster is transferred into this new reference coordinate system, and a bounding box is constructed for the vehicle. This bounding box is then translated back to the original coordinate system and used as the optimum bounding box for this vehicle. The front corner point and the back corner point, both of which are closest to the sensor, are identified as a key data pair to represent the location of this vehicle. The length and width of the object can then be calculated.

d. The difference (Diff) between the height (H) of the object and L, given as

\[ \text{Diff} = H - L \] (3)

A database containing a total of 1,056 objects were manually labeled as vehicles or non-vehicles. The type (vehicle or non-vehicle) of the object was determined by checking the corresponding videos. The database was further divided into three parts: training set (500 objects), validation set (241 objects), and testing set (315 objects). This research trained a support vector machine (SVM) as the classifier. A SVM constructs a hyperplane or a set of hyperplanes in a high or infinite dimensional space. A kernel function k can fit the maximum-margin hyperplane in a transformed feature space. k can be represented as

\[ k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j) \] (4)

Where k is the kernel function, \( \mathbf{x}_i, \mathbf{x}_j \) are n dimensional inputs, \( \varphi \) is a map from n-dimension to m-dimension space. \( (\mathbf{x}_i, \mathbf{x}_j) \) denotes the dot product.

The classification vector \( \mathbf{\tilde{a}} \) in the transformed space is

\[ \mathbf{\tilde{a}} = \sum_{i=1}^{n} c_i \varphi(\mathbf{x}_i) \] (5)

Where the coefficients \( c_i \) can be calculated through

\[ \max f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i \left( \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j) \right) \varphi(\mathbf{x}_j)^T c_j y_j = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_i k(\mathbf{x}_i, \mathbf{x}_j) c_j y_j, \]

\[ \sum_{i=1}^{n} c_i y_i = 0, \quad 0 \leq c_i \leq \frac{1}{2m_\Delta} \] (6)

\( \varphi(\mathbf{x}_i) \) lies on the boundary of the margin in the transformed space, and

\[ b = \mathbf{\tilde{a}} \cdot \varphi(\mathbf{x}_i) - y_i = \sum_{j=1}^{n} c_j y_j \varphi(\mathbf{x}_j) \cdot \varphi(\mathbf{x}_i) - y_i \]

\[ = [\sum_{j=1}^{n} c_j y_j k(\mathbf{x}_i, \mathbf{x}_j)] - y_i \] (7)

New points can be classified by computing

\[ \mathbf{\tilde{z}} \mapsto \text{sgn}(\mathbf{\tilde{a}} \cdot \varphi(\mathbf{\tilde{z}}) - b) \]

\[ = \text{sgn}([\sum_{j=1}^{n} c_j y_j k(\mathbf{x}_i, \mathbf{\tilde{z}})] - b) \] (8)

There are four major different kernel types: linear, polynomial, radial, and sigmoid. The SVM classifiers with different kernels were tested using the same dataset. Table 2 contained the test result. All the classifiers investigated in Table 2 can achieve accuracy with more than 92%. It is shown that among the four types of kernels, the polynomial had the best accuracy for the training set, validation set, and testing set. Two major reasons that lead to the classification failure were that pedestrians close to each other could be misclassified as a vehicle and vehicles partially blocked by other objects could be misclassified as pedestrians.
TABLE II
RESULTS OF DIFFERENT SVM CLASSIFIERS

<table>
<thead>
<tr>
<th>Kernel Type</th>
<th>Dataset</th>
<th>Training Set</th>
<th>Validation Set</th>
<th>Testing Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Total Clusters</td>
<td>500</td>
<td>241</td>
<td>315</td>
</tr>
<tr>
<td>/</td>
<td>Vehicle Clusters</td>
<td>389</td>
<td>185</td>
<td>290</td>
</tr>
<tr>
<td>/</td>
<td>Non-Vehicle Clusters</td>
<td>111</td>
<td>56</td>
<td>25</td>
</tr>
<tr>
<td>Radial</td>
<td>Identified Vehicle Clusters</td>
<td>386</td>
<td>183</td>
<td>272</td>
</tr>
<tr>
<td></td>
<td>Identified Non-Vehicle Clusters</td>
<td>114</td>
<td>58</td>
<td>43</td>
</tr>
<tr>
<td>Linear</td>
<td>Identified Vehicle Clusters</td>
<td>384</td>
<td>179</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>Identified Non-Vehicle Clusters</td>
<td>116</td>
<td>62</td>
<td>42</td>
</tr>
<tr>
<td>Polynomial</td>
<td>Identified Vehicle Clusters</td>
<td>115</td>
<td>182</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>Identified Non-Vehicle Clusters</td>
<td>385</td>
<td>59</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>99.0</td>
<td>96.7</td>
<td>94.6</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>Identified Vehicle Clusters</td>
<td>389</td>
<td>190</td>
<td>276</td>
</tr>
<tr>
<td></td>
<td>Identified Non-Vehicle Clusters</td>
<td>111</td>
<td>51</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>97.6</td>
<td>94.6</td>
<td>93.0</td>
</tr>
</tbody>
</table>

E. VEHICLE TRACKING

Finally, an object tracking algorithm was developed to track vehicle trajectories. For a vehicle, a single LiDAR sensor can only scan parts of it [26]. A centroid point of the scanned points may change heavily frame by frame. Due to its large size and the tiny time interval between two frames, the unstable centroid point will cause extreme systematic bias in speed. This tracking algorithm utilizes the geometric location information of vehicle key data pairs to identify key points in different frames belonging to the same vehicles. The algorithm tracks the front key point when the vehicle is approaching the sensor and tracks the back corner key point when the vehicle is leaving the sensor. The global nearest neighborhood (GNN) was applied for data association between different frames [27]. The GNN evaluates each observation in the track gating region and picks up the best one for tracking. One challenge for vehicle tracking is to justify whether the travel distance is shorter than the distance between continuous vehicles in the same lane. Figure 3 shows the comparison of frame-to-frame travel distance and distances between different vehicles in the same lane. It is clearly shown that the frame-to-frame distance is shorter than the distance between different vehicles. This confirms that the GNN can always track the same vehicle in different frames.
The second dataset was collected at the intersection of Evans Street and Enterprise Road in Reno, Nevada which is a two-way-stop-sign intersection with four legs. Speed profiles of two vehicles are shown in Figure 6.

The first vehicle stopped at the eastbound Enterprises Road, waited for a gap and then crossed the intersection. The follow-up vehicle decreased the speed and waited for the leading vehicle to pass the street, as the frame index 4000-4500 indicates. The follow-up vehicle then accelerated and approached to stop bar. When there was a gap, it accelerated sharply and crossed the intersection. The speed profiles at this site showed that the proposed procedure can detect and track vehicles at the stop-go situation.

The processing procedure was then applied to the data collected on Mccarran Blvd@Evans Ave. Figure 7 shows a speed profile of one vehicle without deceleration. It is shown that vehicles with relatively high speed can also be detected and tracked.

B. TRAFFIC VOLUME VALIDATION

Two intersections (N Sierra St@ 11th St; N Virginia St@ 10th St) were selected for traffic volume validation. The continuous frames of LiDAR data were made into a video and were uploaded into YouTube [29]. A 360-degree camera was installed at the two sites. The traffic volume can then be manually checked in the video captured by the camera. Figure 8 shows the google map of the two sites.
The results showed that the vehicle volume counted from the camera and the volume calculated from the LiDAR were close to each other, though not exactly matching. The offsets were mainly caused by the misclassification, between the vehicles and the pedestrians in the LiDAR data [30].

Figure 10 shows the object trajectories extracted at one intersection. In Figure 10, the red points represent vehicle trajectories and the green points represent pedestrian trajectories. Apparently, some points in the traffic lanes should be vehicle points, but they were misclassified as pedestrians (green points) due to the occlusion issue.

IV. DISCUSSION ABOUT THE POSSIBLE APPLICATIONS OF HRMTD

The near future traffic would be mixed traffic between connected vehicles and unconnected vehicles. The unconnected vehicles or pedestrians at the intersection can benefit from the HRMTD generated by this data processing framework directly. First, the roadside unit would detect the pedestrians. Then the detected location, speed, and direction would be sent to the vehicles which might have a collision with the pedestrians. After the alert is received, the possible distracted driver would notice the pedestrians and pay attention to the pedestrians. Hence the pedestrians could be better protected [31]. This communication procedure would not require every vehicle or pedestrian to be connected. Both unconnected vehicles and pedestrians could be perceived and protected.

The Rectangular Rapid Flash Beacon (RRFB) is widely used in Reno, Nevada to reduce crashes between vehicles and pedestrians at un-signalized intersections and mid-block pedestrian crossings. The pedestrians need to push a button to activate the flash beacon, to notify the upcoming traffic their intention to cross the street. An official FHWA-sponsored experimental implementation and evaluation conducted in St. Petersburg, Florida found that RRFBs at pedestrian crosswalks are dramatically more effective at increasing driver yielding rates to pedestrians than traditional overhead beacons. However, some pedestrians are reluctant to push them when they are crossing the street. This behavior is especially dangerous at night time. The reason is that some...
drivers expect to see the flash beacon if pedestrians are crossing the street. If they do not see the flash beacon, they will not slow down in the middle block road segment. The RRFB could be upgraded to detect pedestrians if a LiDAR component was integrated automatically. The developed data processing procedure could also be applied here to improve traffic safety. The wildlife crossings signs could also be upgraded to a system similar to automatically RRFB. First LiDAR will detect the crossing wild animals; then the flashing beacon would notify the upcoming traffic (which is usually at high speed during rural highways), so the drivers would have more response time to take necessary actions.

In most cities, signal timing is revised every few years based on count data collected over two to three days. For communities where traffic patterns are consistent throughout days of the week and months of the year, this limited data set can be acceptable. However, in areas where traffic fluctuates because of school schedules, special events or growing communities where traffic patterns are evolving, that fixed data does not provide the full picture for informed traffic management decisions.

Traffic varies from month to month, week to week even within the same day of the week. A traffic plan that adapts to these intra-day changes would provide better performance. However, like any adaptive scheme, it relies on continuous measurements, and cannot be implemented using non-real-time traffic data collected several times each year. The data processing procedure developed in this research could be applied to this problem. The LiDAR could be installed at the intersection, and automatically perceives the vehicles on the road. The data processing procedure developed in this research has the capabilities to track how many vehicles turn left, how many vehicles go straight, how many vehicles turn right, and how many pedestrians crossing the street by days, hours, minutes even seconds. All of this real-time micro information could be adopted by traffic management agencies to provide a full picture of traffic management decisions.

V. SUMMARY

This research developed a data processing procedure for detecting and tracking vehicle trajectories with a roadside LiDAR sensor. Different from existing perception methods for the autonomous vehicle system, this procedure was explicitly developed to extract trajectories from a roadside LiDAR sensor. The data extraction procedure has been validated by comparing tracking results and speeds logged from a testing vehicle through the onboard diagnostics interface (OBD-I). The validation results suggest that the tracking speed matches real driving speed accurately. This data processing procedure not only could be applied to extract high-resolution trajectories for connected-vehicle applications, but it could also be valuable to practices in traffic safety, traffic mobility, and fuel efficiency estimation. It will be better to test the method performance while considering roads that are well known for high traffic volume and roads that are well known for low traffic volume. This test will be the next step of our work [32]. Future research should investigate how to address rain or snow interference. The heavy rain or snow will introduce noises to the dataset. In extreme situations, it is even possible for them to make sensor detection range much smaller than pleasant weather. How to address this problem would be essential for deploying this system in areas regularly having these weather conditions. Finally, how to deliver the system output in real-time is also a key challenge. This is especially important if the future connected vehicles system requires limited response time.

ACKNOWLEDGMENT

(Jingrong Chen and Sheng Tian contributed equally to this paper.)

REFERENCES

[29] https://youtu.be/3Y5zWVQKuhA. Accessed on June 20, 2019