


Vehicle Tracking and Speed Estimation From Roadside Lidar

Jiaxing Zhang, Wen Xiao , Benjamin Coifman, and Jon P. Mills

Abstract—Vehicle speed is a key variable for the calibration, validation, and improvement of traffic emission and air quality models. Lidar technologies have significant potential in vehicle tracking by scanning the surroundings in 3-D frequently, hence can be used as traffic flow monitoring sensors for accurate vehicle counting and speed estimation. However, the characteristics of lidar-based vehicle tracking and speed estimation, such as attainable accuracy, remain as open questions. This research therefore proposes a tracking framework from roadside lidar to detect and track vehicles with the aim of accurate vehicle speed estimation. Within this framework, on-road vehicles are first detected from the observed point clouds, after which a centroid-based tracking flow is implemented to obtain initial vehicle transformations. A tracker, utilizing the unscented Kalman Filter and joint probabilistic data association filter, is adopted in the tracking flow. Finally, vehicle tracking is refined through an image matching process to improve the accuracy of estimated vehicle speeds. The effectiveness of the proposed approach has been evaluated using lidar data obtained from two different panoramic 3-D lidar sensors, a RoboSense RS-LiDAR-32 and a Velodyne VLP-16, at a traffic light and a road intersection, respectively, in order to account for real-world scenarios. Validation against reference data obtained by a test vehicle equipped with accurate positioning systems shows that more than 94% of vehicles could be detected and tracked, with a mean speed accuracy of 0.22 m/s.

Index Terms—Smart city, 3-D lidar, traffic monitoring, urban sensing, vehicle detection.

I. INTRODUCTION

CITIES are facing increasing challenges in traffic management and air pollution induced by heavy traffic. Emissions from on-road vehicles are widely regarded to be the main source of air pollution in urban areas [1]. The key input data source to air quality models is usually generated from vehicle emission models, which is supported by traffic data. Accordingly, using better traffic flow representations is fundamental to improving

emission estimates, and to subsequently improve air quality modeling results. Among the different levels of traffic models, microscopic models operate with more detailed and precise traffic data through variables which include individual vehicle position, speed, acceleration, and deceleration [2]. All such variables can be acquired through vehicle tracking from various monitoring sensors, with accurate tracking of individual vehicles of great importance in creating high-resolution microscopic traffic data to serve traffic modeling and emission studies.

The most commonly used traditional vehicle tracking sensors are inductive loop and infrared detectors, nevertheless, such approaches have drawbacks: Inductive loop detectors are, for example, subject to a high failure rate when installed in poor road surfaces [3], and infrared detectors are very sensitive to extreme weather conditions such as rain, fog and snow [4]. With advanced progress in image processing techniques, vision-based methods have therefore become one of the most promising approaches to vehicle tracking. Video cameras can produce richer visual information than traditional devices without affecting the integrity of the road surface and the observed information can be processed intuitively with modern computer vision technologies [5]. However, whilst cost effective and well developed in terms of data processing, the level of accuracy is limited by image distortion and resolution, and optical cameras are sensitive to adverse illumination and weather conditions.

Panoramic 3-D lidar actively scans its 360° surroundings frequently, capturing a huge volume of data on objects within scanning range. A panoramic 3-D lidar sensor is usually composed of a number of vertically configured laser beams covering a wide vertical field-of-view (FOV), dependent on the number of beams. The instrument rotates around its vertical axis to generate a panoramic view of the surroundings with data recorded in the form of a 3-D point cloud. Operating at a high frequency, vehicles can thereby potentially be detected and tracked directly in 3-D with high spatial accuracy and temporal resolution. Such sensors have been adopted extensively for environment perception in autonomous vehicles [6], however, they are seeing increased use as traffic monitoring sensors due to the ability to capture objects directly in 3-D with a high accuracy [7]. The precision of the obtained individual measurements can be as high as 2–3 cm. Moreover, with the ongoing development of lidar technology and increased ubiquity, the cost of such sensors has dramatically decreased in recent years. Therefore, it is foreseen that such sensors will be widely employed in smart cities for intelligent transportation systems in the near future.

Manuscript received April 30, 2020; revised July 17, 2020 and August 6, 2020; accepted September 13, 2020. Date of publication September 18, 2020; date of current version September 30, 2020. This work was supported by UKCRIC - UK Collaboratorium for Research in Infrastructure & Cities: Newcastle Laboratories (EPSRC award EP/R010102/1). The work of Jiaxing Zhang was supported by the China Scholarship Council Studentship, under Grant 201706370243. (Corresponding author: Wen Xiao.)

Jiaxing Zhang, Wen Xiao, and Jon P. Mills are with the School of Engineering, Newcastle University, NE1 7RU Newcastle Upon Tyne, U.K. (e-mail: j.zhang85@ncl.ac.uk; wen.xiao@ncl.ac.uk; jon.mills@ncl.ac.uk).

Benjamin Coifman is with the Department of Civil, Environmental and Geodetic Engineering, Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: coifman.1@osu.edu).

Digital Object Identifier 10.1109/JSTARS.2020.3024921

According to the literature, most existing roadside lidar-based vehicle tracking methodologies adopt a tracking-by-detection principle. First, the raw lidar data is preprocessed to segment the moving points from the background. Second, the moving points are clustered into small groups representing individual road users. Additionally, vehicles may be extracted either by locating the lanes which vehicles occupy [8], or by vehicle and nonvehicle classification among all determined clusters [9]. In the final stage of vehicle tracking, the commonly used methods are Global Nearest Neighbor (GNN) [8] and Kalman filtering (KF) [10]. It is also noteworthy that these two algorithms are used simultaneously in some works [9], [11], [12]. Normally, the spatial center of a cluster is identified as the location of the vehicle being tracked. Nevertheless, this center shifts, frame by frame, due to the incompleteness of the scanned vehicle from the roadside lidar system. There are several studies that have taken this shift into consideration [11]–[13], and these have largely adopted the same strategy whereby the point in the vehicle cluster closest to the lidar instrument is selected as the tracking point. This means that when the vehicle is approaching the sensor, the front corner of the cluster acts as the tracking point, and when the vehicle is leaving the sensor, the back corner of the cluster is the tracking point. Although such a strategy has indeed increased the accuracy of the estimated vehicle speeds to a certain extent, the issue still warrants further investigation when accurate vehicle speed determination is the end goal.

Intending to address the aforementioned issue, a new integrated vehicle tracking methodology from roadside lidar data is therefore proposed in this article. There are two main tracking stages: centroid-based tracking and tracking refinement. A tracker composed of Unscented Kalman Filter (UKF) [14] and joint probabilistic data association filter (JPDAF) [15] is used in the first stage, which adopts the centroids of clusters as the input to obtain initial vehicle speeds. A tracking refinement module, adopting a different strategy from previous studies, is proposed in the second stage to further improve the accuracy of the calculated vehicle speeds. In this module, vehicle positions are rectified based on the refined transformation between successive vehicle cluster pairings. The implemented tracking framework has been assessed using a test vehicle installed with a speed reference system under different scenarios.

The contributions of this work are threefold.

- 1) A tracking refinement module is proposed and developed to directly determine the transformation relationship between pairs of consecutive vehicle clusters so that the displacement can be calculated simply without the use of any reference points such that the final speed estimation accuracy is improved.
- 2) The speed estimation results are independently validated against an accurate vehicle speed reference system, comprising dual-frequency GNSS, IMU, and Odometer, such that the true achievable accuracy of speed estimation using panoramic lidar can be discovered and the reliability of results assured.
- 3) The developed framework has been demonstrated through different urban scenarios using two different lidar sensors to provide insight on real-world implementations of

panoramic lidar sensors for traffic monitoring applications.

II. RELATED WORK

Tracking-by-detection is widely acknowledged in existing vehicle tracking studies, consequently, a brief review of vehicle detection and tracking is conducted in this section.

A. Vehicle Detection From 3-D Lidar Points

A preponderance of studies on vehicle detection from 3-D lidar points rely on machine learning strategies. Traditional machine learning has proved an efficient approach to vehicle detection from lidar data, with feature selection and classifier training two important factors in its effective implementation. Low-level features based on a small group of points are used in the majority of related studies whereby the selected features to distinguish vehicles and pedestrians mainly comprise shape information [10]. Specifically, the number of points in the object cluster, object length, height profile, difference between height and length, distance to the lidar instrument, and direction of the point distribution in the cluster are typically adopted. There are several classifiers commonly used in lidar-based object detection. For example, support vector machine (SVM) was trained as the classifier and four major different kernel types were compared in [9]; the performance of different classifiers, including naive Bayes, k -nearest neighbor, SVM, and random forest (RF), were compared in [11]; and SVM using the Gaussian radial basis function and RF were experimented with in [16] to detect vehicles from point clouds.

Current state-of-the-art object detection deep learning networks normally follow one of two pipelines: either two-stage or one-stage object detection. In the former pipeline, several object candidates termed regions of interest (ROI) or region proposals (RP) are extracted from a scene. Afterward, these candidates are verified and refined in terms of classification scores and locations. For example, in [17], lidar points were clustered for on-ground obstacles using DBSCAN. These clusters were then fed into a ConvNet for 2-D detection. Chen *et al.* [18] proposed to generate 3-D ROIs from the bird's eye view lidar feature maps by a RP Network, and combine the regional features from the front view lidar feature maps and RGB camera images for 3-D vehicle detection. Another two-stage 3-D object detector, PointRCNN [19], in which stage-one is bottom-up 3-D proposal generation and stage-two refines proposals in the canonical coordinates to obtain the final detection results.

In the one-stage object detection pipeline, single-stage and unified CNN models are used to directly map the input features to the detection outputs. Li *et al.* [20] employed a fully convolutional network on lidar point clouds to produce an objectness map and several bounding box maps. Yang *et al.* [21] proposed a proposal-free, single-stage detector named PIXOR, which outputted oriented 3-D object estimates decoded from pixel-wise neural network predictions. PointPillars [22], another novel object detection encoder, utilized PointNets to learn a representation of point clouds organized in vertical columns

(pillars), then scattered these features back to a 2-D pseudoimage for a convolutional neural network.

It can be seen from this brief review that machine and deep learning technologies have been widely adopted in the field of vehicle detection from raw 3-D point clouds. It is also noteworthy that almost all studies were evaluated using the KITTI benchmark [23]. However, the density of the data collected for the purposes of this research was different from that of the KITTI dataset, which was collected using a Velodyne HDL-64 laser scanner (and which consequently has more lidar beams than the 32 and 16 line scanners used herein). Moreover, as opposed to a common object detection problem, vehicle detection in this research is essentially an issue of straightforward binary classification: after preprocessing operations comprising background removal and moving point clustering, vehicle detection is actually simplified to classification of motor vehicles from other road-going objects, primarily pedestrians and cyclists. Taking the above factors into account, deep learning strategies do not necessarily outperform traditional classification methods, especially when training data is limited. More details can be seen in Section IV.

B. Vehicle Tracking From 3-D Lidar Points

After the vehicle detection stage, the resulting set of identified trackable vehicles is processed for pose and centroid estimation, accompanied by known model fitting [24]. This generates a valid measurement of a vehicle for which the attributes further evolve over time. These measurements are fed to the state estimation filter to predict the kinematic state of each vehicle. The optimal Bayesian filters are utilized to estimate the possible evolution of vehicle states in the presence of uncertainties. Available filters include the Variant Kalman, Particle, and Interacting-Multiple-model (IMM) [25] filters. Later, a data association process is applied to either assign the detected vehicle to an existing track or initiate a new track. Popular data association algorithms used in existing lidar-based vehicle tracking studies include GNN, JPDAF, multiple hypothesis tracking [26], and Hungary [27] algorithms. Finally, a track management module is used to maintain the tracks and to cancel spurious occurrences.

Extended Kalman Filter (EKF) and UKF are two popular state estimation algorithms using nonlinear models for object tracking. UKF is shown to deliver more accurate estimates than EKF in the presence of strong nonlinearity, thus, it is more widely used in lidar-based multiobject tracking, as presented in [28]–[30]. Moreover, there are primarily two classes of data association filter; the deterministic filter and the probabilistic filter. Their representatives are the Nearest Neighborhood Filter (NNF) and probabilistic data association filter (PDAF), respectively. The PDAF outperforms the NNF by avoiding hard, possibly erroneous, association decisions commonly encountered in clutter scenarios where multiple measurements are located close to each other. As an extension of PDAF, the JPDAF can handle multiple target tracking tasks. This background provides the rationale for adopting the UKF and JPDAF as the core algorithms of the vehicle tracker in this work. Similar combinations, such as IMM-UKF-JPDAF, can be found in, e.g., [24] and [30].

C. Tracking Refinement

As mentioned in the Introduction, vehicles can only be partially scanned due to self-occlusion in panoramic lidar systems. This has inevitably been responsible for unsatisfactory tracking results, especially in vehicle speeds, by many current centroid-based tracking methodologies. The few efforts made so far to improve this situation are summarized as follows: in [30], in addition to the position tracker represented by the IMM-UKF-JPDA filter, a bounding box tracker was provided to perform geometrical correction to compensate for occlusion and ego-vehicle perspective change. However, the shifting relationship between the center points in the original occluded box and the newly shifted box was based on simple empirical rules, and judgments (such as the vehicle location relative to the ego-vehicle) had to be made. A novel operation was displayed in [8] where the corner of the bounding box closest to the lidar sensor was chosen to be a reference point of the vehicle. This point replaced the bounding box center to represent the vehicle position and was used as input to the Discrete Kalman Filter tracking procedure. Several other related studies, e.g., [8], [9] found in the existing literature adopt similar tracking refinement strategies to that in [10], implying there has not been any further improvement to solving the issue. A new tracking refinement strategy is therefore proposed and introduced in this article.

III. METHODOLOGY

The proposed methodology consists of vehicle detection, tracking, refinement and, finally, speed validation. Vehicles are detected via a three-step procedure, then tracked by UKF and JPDAF, which takes the centroid of the cluster as the vehicle position, resulting in biases in vehicle speeds due to the incompleteness of the scanned clusters. Accordingly, a tracking refinement module is developed to improve this situation. The core strategy in this module is image matching, in which the vehicle clusters are transformed to 2-D images. Finally, the estimated speeds are validated against a reference from a test vehicle equipped with an independent positioning system. An overview of the proposed framework is presented in Fig. 1.

A. Vehicle Detection

The first step of the framework is to distinguish vehicles from the background and other road objects. The operations include background removal, moving point clustering, and vehicle classification. Fig. 2 provides an illustration of the entire vehicle detection process.

1) *Moving Point Extraction and Clustering*: All points observed can be generally categorized into two types: moving points belonging to objects changing in their spatial location and static points from the background environment. The maximum distance [7] method is applied to construct the background with the principle that the static environment is assumed impenetrable and only the furthest points of each laser beam are considered to be located on the static background. Therefore, the core operation of background removal is to delete the furthest point of each laser beam for all scanning angles. Fig. 2(b) shows an

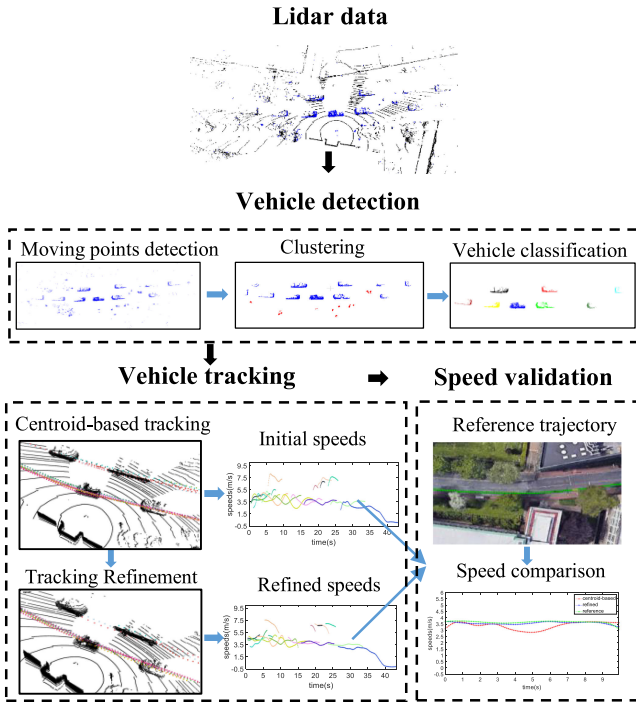


Fig. 1. Overview of the proposed methodology.

example of the moving points extracted from an original point cloud.

The Euclidean Cluster Extraction algorithm is used for clustering and there are three important parameters: the minimum cluster size S_1 , the maximum cluster size S_2 , and the minimum distance d between two clusters. In terms of d , if the value is too small, a real single object can be incorrectly observed as multiple clusters. Conversely, if the value is too large, multiple objects can be regarded as a single cluster. Therefore, empirical testing is suggested to determine the optimal value of d for datasets from different test sites. In our test datasets from the two types of laser scanners, the minimum distance between two vehicles is around 1.5 m, and the minimum distance between a pedestrian and a vehicle is around 1.8 m. Therefore, d is set to be 1 m in the tests. The cluster size is dependent on the sensor's number of beams, thus needs to be adjusted for different sensors. According to the statistics, the largest vehicle cluster contains around 6000 points and the smallest contains around 200 points in datasets from RS-LiDAR-32, so $S_1=150$, and $S_2=6500$. Whilst since the point density is much lower for the VLP-16, the values are smaller for these datasets: $S_1=50$, $S_2=5500$.

2) *Vehicle and Nonvehicle Classification:* The purpose of this stage is to select vehicles out of all the moving clusters. The remaining moving points after background removal belong to either vehicles or nonvehicles including pedestrians, cyclists, motorcyclists, and false alarms (e.g., waving trees and bushes). Therefore, the vehicle detection task is simplified to a binary classification problem, for which, an SVM classifier with radial basis function as the kernel function and an RF classifier with 20 trees are exploited. A 28-D feature set $F=[f_1, f_2, f_3, f_4]$ containing both volumetric and distributional information of the

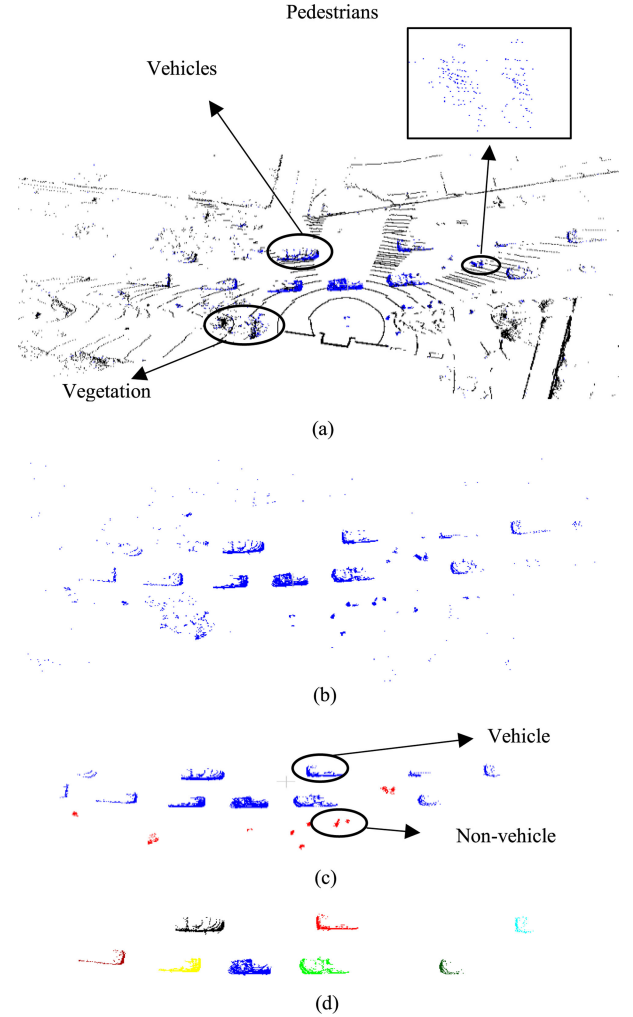


Fig. 2. Vehicle detection process. (a) Original point cloud. (b) Moving points. (c) Clustering and vehicle classification. (d) Detected vehicles.

clusters, shown as follows, is considered as potential choice to train the classifiers.

- 1) The vertical point distribution histogram of the cluster: the proportion of the overall number of points in each vertical section varies among different urban objects [16]. The input cluster is divided into 20 vertical sections from its overall height to the ground

$$f_1 = [v_1, v_2, \dots, v_{20}].$$

- 2) The standard deviation of points in the cluster

$$f_2 = [x_s, y_s, z_s].$$

- 3) The volume size of the cluster

$$f_3 = [\text{Length}, \text{Width}, \text{Max_height}, \text{Min_height}].$$

- 4) The area of the 2-D minimum bounding box of the cluster: $f_4 = a$.

The importance of each feature in F is evaluated using RF, and SVM and RF are trained by different subsets comprised of different number of features. In addition, a simple rule-based

classifier is tested for comparison. The area of the 2-D bounding box of the cluster is used as a threshold to distinguish vehicles and nonvehicles so as to detect as more vehicles as possible. Performance of these three classifiers will be evaluated in the experiment section.

B. Centroid-Based Tracking

After vehicle detection, the clusters belonging to the same vehicle in successive frames are tracked to recover the trajectory.

1) *Tracker*: UKF is used as the initial function in our vehicle tracker. Compared with other filters, UKF gives better performance when the prediction and update functions are highly nonlinear. UKF uses a deterministic sampling technique known as the unscented transformation to pick a minimal set of sample points around the mean. The JPDAF is a statistical approach used to solve the problem of assignment in a tracking algorithm. Instead of choosing the most likely assignment of measurements to a target, the JPDAF takes an expected value, which is the minimum mean square error estimate for the state of each target. At each observation, it maintains its estimate of the target state as the mean and covariance matrix of a multivariate normal distribution. Moreover, JPDAF can handle multiple target tracking scenarios.

2) *Influence of Vehicle Detection*: The success achieved in vehicle detection will have direct influence on vehicle tracking. Missed detections will lead to interruptions in the related trajectories, whilst false alarms will create erroneous trajectories in the tracking results. Two parameters in the JPDAF algorithm are related to the first issue: c , the threshold for assigning detections to tracks, and d , the threshold for track deletion. c is usually set to a 1×2 vector $[c_1, c_2]$, where $c_1 \leq c_2$. Initially, a coarse estimation is performed to verify which combinations of {track, detection} require an accurate normalized distance calculation. Only combinations for which the coarse normalized distance is lower than c_2 are calculated. Detections can only be assigned to a track if their normalized distance from the track is less than c_1 . The values should be increased if there are detections that are not assigned to any tracks and decreased if there are detections that are assigned to wrong tracks. d is usually set to $[p, r]$, where a track will be deleted if it was unassigned at least p times in the last r updates. Two factors are critical in removing nonvehicle trajectories (either pedestrians crossing the road or other false alarms such as waving trees and bushes): orientation and length of the trajectories. If the orientation of a trajectory deviates too far from others, or its length is too short, the trajectory will be removed.

C. Tracking Refinement

In the tracking stage, the centroid of the cluster was adopted as the vehicle position, however the relative position of the centroid changes frame by frame when the vehicle is passing through the roadside lidar sensor. Fig. 3(a) shows an example vehicle, where F, C, and R are the front, center, and rear points of the vehicle, respectively. Fig. 3(b) illustrates the spatial relations between the centroid of the point cloud cluster (C' and C'') and the real centroid C when the vehicle passes the lidar sensor. It can be

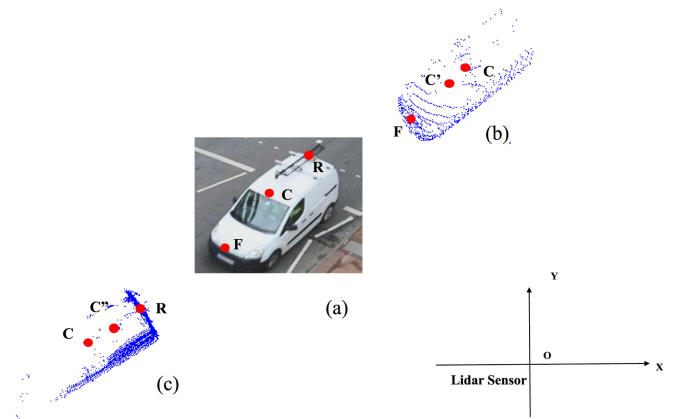


Fig. 3. Vehicle and key tracking points: F, C, R represent the front, center, and rear, respectively (a) and their spatial relations between the centroid of the point cloud clusters (C' and C'') when the vehicle is approaching (b) and leaving (c) the lidar sensor.

seen that C' is between F and C when the vehicle is approaching the lidar sensor as the front of the vehicle is mostly scanned; and C'' is between R and C when it is departing as the rear of the vehicle is mostly scanned. The proposed tracking refinement module is intended to minimize such effects.

After centroid-based tracking, the individual cluster of each vehicle is identified and labeled by the minimum bounding box for subsequent tracking refinement. For each vehicle ID, the optimized tracking refinement solution relies on determining the correct transformation between two successive clusters. The normal strategy for determining the transformation is frame registration. Enlightened by work from [31], 3-D point clouds can be converted to 2-D images to solve the problem by image matching. It is noteworthy that here the conversion is implemented on the previously extracted 3-D vehicle clusters rather than the entire frame, as in [31]. The process of the proposed tracking refinement comprises three steps: conversion from 3-D cluster to 2-D image, image matching, and 2-D to 3-D transformation. Tracking refinement is performed within pairs of successive vehicle clusters in the plan view. Fig. 4 displays the process for one example pair, which is described in the following sections.

1) *Conversion From 3-D Cluster to 2-D Image*: As shown in Fig. 4, Frame m and Frame $m + 1$ are projected to 2-D in the plan view in the first instance, and all the points are in the laser scanner coordinate system XOY . Under this condition, $O_{m+1}(X_0, Y_0)$ is the origin of the minimum bounding box around the vehicle cluster in Frame $m + 1$. Correspondingly, $o_{m+1}(0, 0)$ is the origin of the image which is located in the image coordinate system xoy . (1) shows the conversion from a point on the vehicle cluster to a pixel in the corresponding image, where pixelsize refers to the resolution of the image

$$\begin{aligned} x &= (X - X_0)/\text{pixelsize} \\ y &= -(Y - Y_0)/\text{pixelsize}. \end{aligned} \quad (1)$$

The parameter pixelsize can be decided by testing plausible values in the experiment. In our situation, two case studies in which a test vehicle was tracked have been used to decide the

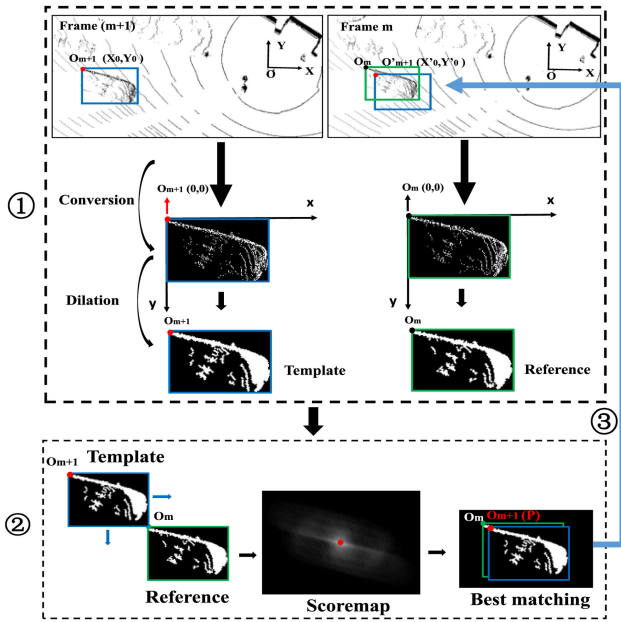


Fig. 4. Workflow of tracking refinement which includes three main steps: (1) Conversion from 3-D cluster to 2-D image. (2) Image matching. (3) 2-D to 3-D transformation.

optimal value in a range from 1 to 10 cm (values outside of this range are considered either too small or too large according to the data density). Root mean square error (RMSE) between the estimated speeds and the reference is calculated for each pixel size. Based on our test, a pixel size of 3 cm generated the lowest RMSE and was therefore chosen as the optimal value.

2) *Image Matching*: Image matching is intended to determine the optimum location of a template within a reference image. The image generated from Frame $m + 1$ is regarded as the template, whereas Frame m is the reference. The template image shifts pixelwise over every possible location in the reference image. Based on the cross-correlation coefficient metric [32], a similarity score $S(x, y)$ is calculated between the template and the corresponding subimage in the reference, accordingly [see (2)]. A score map with each pixel assigned a similarity value is formed after the completion of the search process. The optimum matching location, namely the lightest point (red dot in Fig. 4) in the scoremap, is where the largest score is determined [see (3)]

$$S(x, y) = \frac{\sum_{i=1}^{m_1} \sum_{j=1}^{n_1} T(i, j) R(y+i-1, x+j-1)}{\left[\sum_{i=1}^{m_1} \sum_{j=1}^{n_1} T(i, j) \right]^{1/2} \left[\sum_{i=1}^{m_1} \sum_{j=1}^{n_1} R(y+i-1, x+j-1) \right]^{1/2}} \quad (2)$$

$$P = \arg \max_{x, y} (S(x, y)), \quad (x = 1, \dots, n_2; y = 1, \dots, m_2). \quad (3)$$

In (2), T is the template image with (m_1, n_1) pixels; R is the reference with (m_2, n_2) pixels; and (x, y) is the origin of

the sub-image corresponding to T in R . In (3), P is the optimal matching location in R .

3) *2-D to 3-D Transformation*: Real-world coordinates are reserved for each pixel according to (1). Consequently, P can be located on Frame m , labeled as $O'_{m+1}(X'_0, Y'_0)$ in Fig. 4. Considering its position $O_{m+1}(X_0, Y_0)$ in Frame $m + 1$, the displacement can be calculated and the vehicle speed obtained. The speed values during the entire tracking period will be estimated when a chain of the above operations has been fulfilled amongst all the tracked clusters of the same ID. A Gaussian window with size $s = 20$ is used to smooth the acquired values so as to filter out noises.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Datasets

This study employed two lidar sensors. The first was a RS-LiDAR-32, a panoramic instrument from RoboSense. The sensor has a detection radius of up to 200 m and is designed for various applications such as autonomous vehicles, robotics, and 3-D mapping. It has 32 laser beams and collects data at a speed of 640 000 pts/s. The scanning frequency was set to 10 Hz in our tests. It covers a 360° horizontal field of view and a 40° vertical field of view with $+15^\circ$ upward and -25° downward looking angles. The second sensor was a Velodyne VLP-16, with 16 laser beams and a maximum detection range of 100 m. The vertical field of view of the instrument is 30° with $+15^\circ$ upward and -15° downward. The scanning frequency was also 10 Hz in our experiments.

Two different sites were chosen in Newcastle upon Tyne, UK, to test the proposed method under real-world traffic conditions. At the first site (first row in Fig. 5), the RS-LiDAR-32 was setup along a straight road near a traffic light controlled pedestrian crossing. The lidar sensor was c. 4 m away from the first of two traffic lanes. At the second site (second row in Fig. 5), the VLP-16 was set up at a road intersection. The lidar sensor was c. 4.5 m away from the first of multiple lanes.

A test vehicle equipped with an independent speed reference system was used to validate estimated vehicle speeds. The reference system is composed of two GNSS antennas, an IMU unit and an odometer. The GNSS and IMU unit were mounted on top of the vehicle, while the odometer was installed on one of the rear wheels to improve positional accuracy during GNSS outages. The test vehicle was driven through the scanning area for several rounds at both test sites. The initially obtained reference data were postprocessed using Kinematic software from Advanced Navigation to obtain accurate test vehicle positions. Figures in the middle column of Fig. 5 show examples of the processed trajectories from Test Site 1 and Test Site 2. The uncertainty of reference speeds will be assessed to assure they are accurate enough.

B. Vehicle Detection Results

A dataset containing 316 vehicle clusters and 224 nonvehicle clusters was created manually from lidar data previously collected at several random locations in Newcastle upon Tyne.

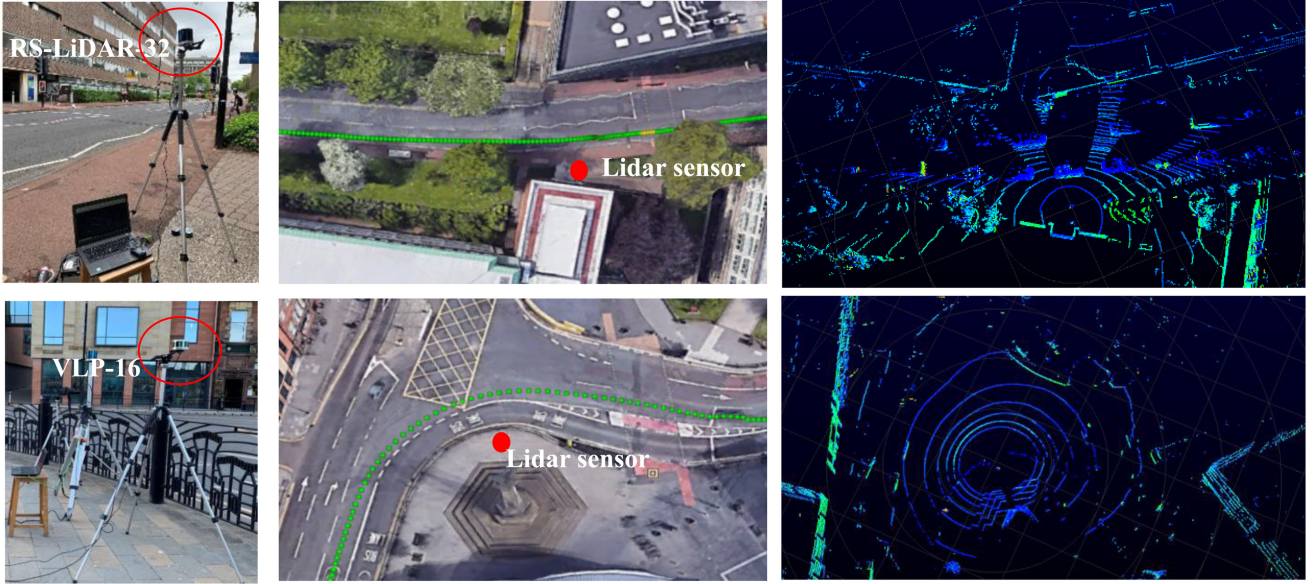


Fig. 5. Test Site 1 and 2: left column: lidar sensor setup; middle column: test scene and postprocessed trajectories of the test vehicle (green dashed lines); right column: lidar data.

TABLE I
PERFORMANCE OF CLASSIFIERS TRAINED BY DIFFERENT FEATURE SETS

	SVM			Random Forest			Rule-based		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
F ₁	0.860	0.960	0.907	0.898	0.924	0.911	0.861	0.993	0.923
F ₃	0.875	0.960	0.916	0.912	0.950	0.931			
F ₅	0.781	0.997	0.876	0.915	0.937	0.926			
F ₂₈	0.956	0.940	0.948	0.902	0.958	0.929			

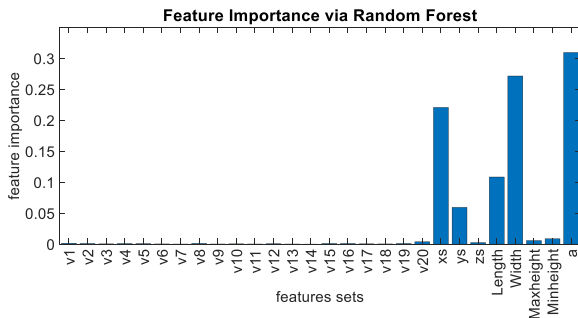


Fig. 6. Estimation of feature importance (28 features).

Around 67% of these were used for training the classifier (210 vehicles and 150 nonvehicle), and the remaining for validation. A test dataset of 697 clusters (300 vehicles, 397 nonvehicles) that was totally new to the classifier was then randomly selected from the lidar observations at Test Site 1. We trained the RF classifier using the 28 features and obtained the weights for each of them, shown in Fig. 6. It can be concluded that a , Width, x_s , Length and y_s are the five features with high importance, among which a is the most important one. As a result, we retrained the SVM and RF classifiers using the following feature sets: $F_1 = a$, $F_3 = [a, \text{Width}, x_s]$, $F_5 = [a, \text{Width}, x_s, \text{Length}, y_s]$,

TABLE II
ANALYSIS OF REFERENCE SPEED IN TWO STATIONARY SECTIONS

	SD(m/s)	Mean(m/s)	RMSE(m/s)	Static period(s)
Section 1	0.006	0.023	0.028	19 to 42
Section 2	0.005	0.019	0.020	10 to 51

$F_{28} = [v_1, v_2, \dots, v_{20}, x_s, y_s, z_s, \text{Length}, \text{Width}, \text{Max_height}, \text{Min_height}, a]$. In addition, a simple rule-based method using the size of the clusters' bounding boxes was implemented for comparison with these two classifiers. Three indices, precision, recall, and F1-score, were used to assess the performance. From comparison results shown in Table I, it is found that the overall performance of the three classifiers on four feature sets can be regarded as relatively indistinguishable. Both SVM and RF performed slightly differently with different number of features, and SVM produced the highest F1-score with all the features. Whereas, the rule-based method performed best in terms of recall with a decent precision. In order to keep as many vehicles as possible to facilitate the tracking process, the rule-based method is adopted even though its overall performance is not the best.

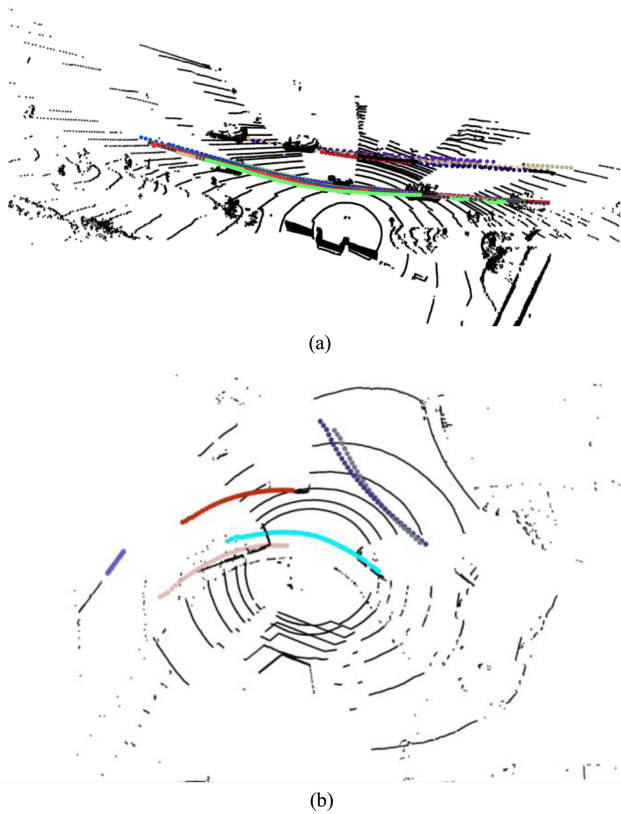
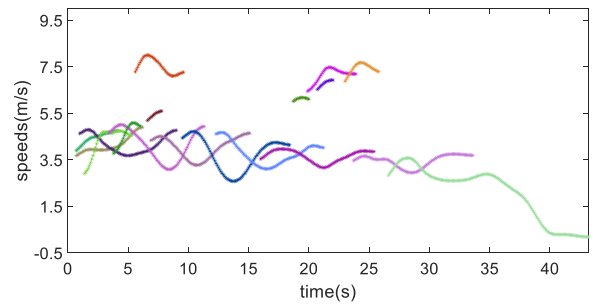


Fig. 7. Trajectories of centroid-based tracked vehicles in case 1 (a) and case 5 (b). Each color represents a vehicle with a unique ID.

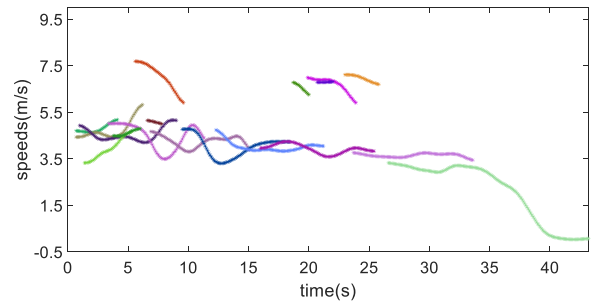
C. Tracking and Speed Estimation

In order to demonstrate the speed estimation under different traffic flow conditions at both test sites, six cases—three from each test site—were selected when the test vehicle passed through the lidar sensor. The first three cases were at Test Site 1. The test vehicle in case 1 and 2 showed different dynamics, with one demonstrating continuous movement and the other showing a pattern of stop-and-go. The lidar sensor was set to a different look angle in case 3, as an aid to understand how best to configure the optimal vertical FOV. The final three cases were from Test Site 2, where the test vehicle displayed three different movement patterns: turning left, turning right, and driving straight forward. In each of the six cases, two sets of vehicle speeds were acquired through the tracking process: initial values from centroid-based tracking and refined values after tracking refinement. Both were compared with the reference datasets in order to assess the accuracy improvement. Case 1 and case 5 are illustrated in detail.

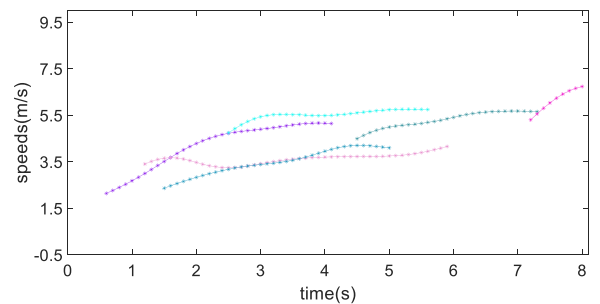
1) *Vehicle Tracking Performance*: Figs. 7 and 8 show the trajectories and speeds of all the tracked vehicles in case study 1 and case study 5. In case study 1, the recording was shown for 43.3 s and tracking continued through the whole period, with all 18 vehicles that appeared in the scanning area during this period successfully tracked. The tracking range of the approach is c. 45 m. Short trajectories were generated from vehicles that were either close to the edge of the scanning area at the beginning of the observation period, or that were occluded by other vehicles



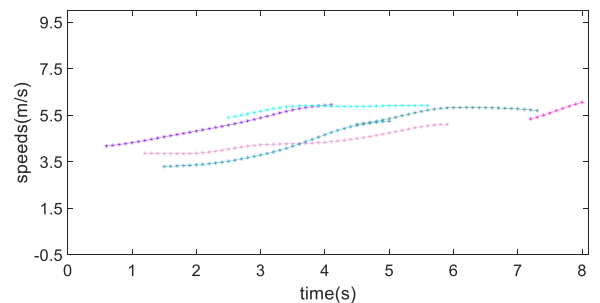
(a) Speeds of all tracked vehicles in case 1 (Centroid-based)



(b) Speeds of all tracked vehicles in case 1 (Refined)



(c) Speeds of all tracked vehicles in case 5 (Centroid-based)



(d) Speeds of all tracked vehicles in case 5 (Refined)

Fig. 8. Speeds of tracked vehicles in case 1 and case 5: (a) and (c) show the centroid-based speeds; (b) and (d) show the refined speeds. Each color represents a vehicle with a unique ID.

during tracking process. In case study 5, the tracking was shown c. 2 s before the test vehicle entered the scanning range and stopped c. 2 s after it left. The whole process lasted for c. 8 s and all six vehicles (including one bus) were tracked. Two vehicles that were turning left were continuously tracked. Tracking of another two vehicles which were turning right, including the test vehicle, were also successively implemented. The trajectories of the remaining vehicles were very short, for example, one in the

TABLE III
EVALUATIONS OF SIX CASE STUDIES

		RMSE(m/s)		MAE(m/s)		Mean speed of test vehicle (m/s)	Vehicle travel direction
		RMSE ₁₃	RMSE ₂₃	MAE ₁₃	MAE ₂₃		
Site 1	Case 1	0.41	0.10	0.31	0.09	3.65	Straight on
	Case 2	0.21	0.07	0.13	0.06	0.70	Straight on
	Case 3	0.38	0.33	0.29	0.27	2.60	Straight on
	Mean	0.33	0.16	0.24	0.14		
Site 2	Case 4	0.53	0.21	0.53	0.17	6.11	Turning left
	Case 5	0.47	0.30	0.42	0.23	5.81	Turning right
	Case 6	0.44	0.33	0.38	0.29	7.29	Straight on
	Mean	0.48	0.28	0.44	0.23		
	Overall mean	0.41	0.22	0.34	0.18		

1: centroid based speeds; 2: refined speeds; 3: reference.

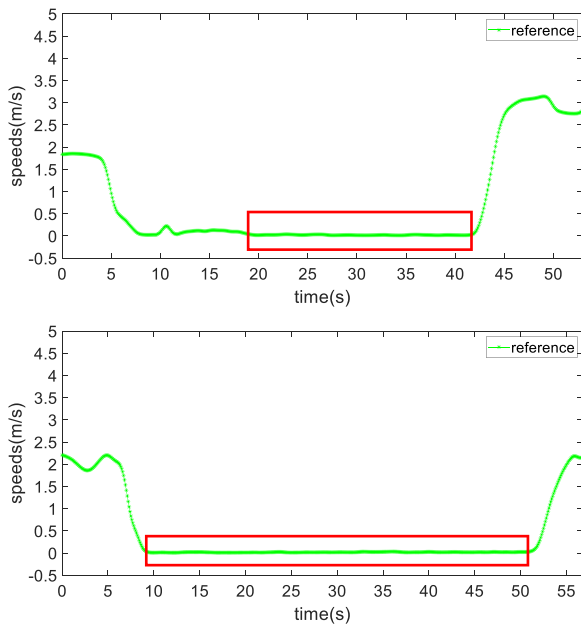


Fig. 9. Two speed estimations including sections (labeled with red box) when the test vehicle was stationary.

second lane. The successful tracking range in this case was c. 18 m.

2) *Evaluation for the Reference Speed Value:* The uncertainty of vehicle speeds from the reference system was evaluated. The evaluation was first implemented when the test vehicle was stationary so that the true speed was known to be 0 m/s. Two sections (shown in Fig. 9) from Test Site 1 were chosen to create the statistics based on the following indices: Standard Deviation (SD), Mean, and RMSE. In the two cases, the durations when the vehicle was stationary were 23 and 41 s, respectively. As shown in Table II, the average RMSE of two cases was 0.024 m/s, which means the displacement deviation is within 3 mm per frame (frame rate 0.1 s).

3) *Comparison Among Three Sets of Vehicle Speeds:* Comparison among three sets of speeds (the centroid-based speeds; the refined speeds; the reference) is conducted for all the six

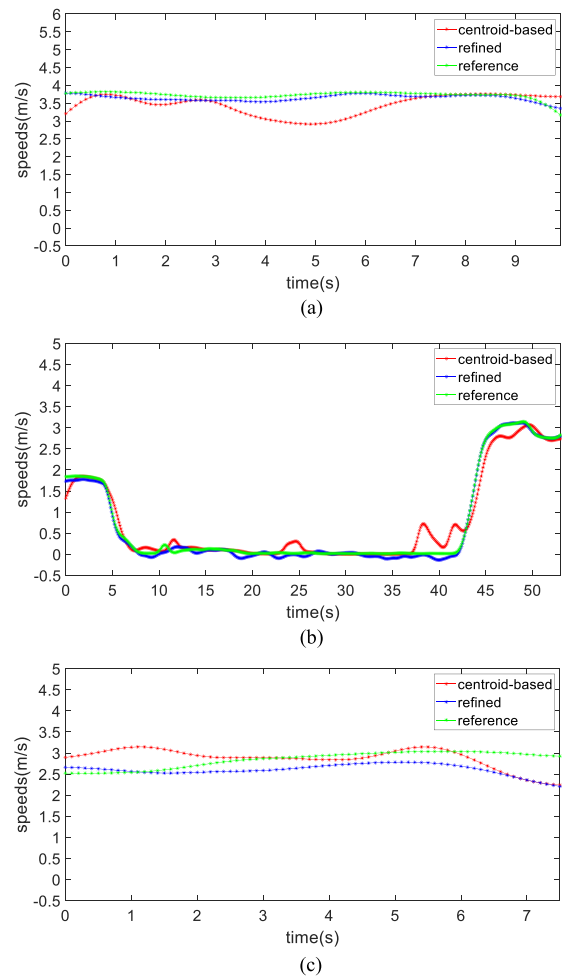


Fig. 10. Comparison results of the test vehicle in three cases from Test Site 1: red: centroid-based speeds; blue: refined speed; green: the reference. (a) Test vehicle speeds in case 1. (b) Test vehicle speeds in case 2. (c) Test vehicle speeds in case 3.

cases. Figs. 10 and 11 illustrate the comparison results of the test vehicle in all the six cases from two test sites. The test vehicle at Test Site 1 was either moving forward with a stable speed or with a pattern of “stop-and-go.” The test vehicle at Test

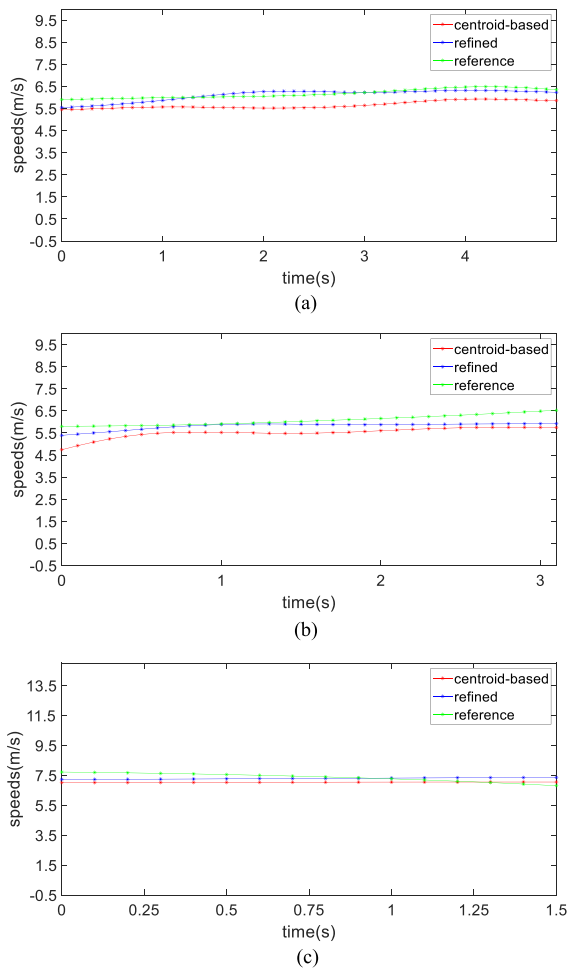


Fig. 11. Comparison results of the test vehicle in three cases from Test Site 2: red: centroid-based speeds; blue: refined speed; green: the reference. (a) Test vehicle speeds in case 4. (b) Test vehicle speeds in case 5. (c) Test vehicle speeds in case 6.

Site 2 was turning left and right in the first two cases, while going forward with a constant speed in the third case. Also it is noteworthy that the test vehicle was going generally faster at Site 2 than at Site 1.

Despite the above movement variability, the refined speeds and the reference are closely in accordance with the roughly estimated moving trends, whereas centroid-based speeds obviously deviate further from the reference. Therefore, it is clear that the tracking refinement step has improved the speed accuracy for all the six cases.

RMSE and mean absolute error (MAE) are used to quantitatively evaluate the three sets of speeds. As seen from Table III, the accuracy of the estimated speeds was improved by the refinement module, under different vehicle dynamics. The mean RMSE value has been decreased from 0.41 to 0.22 m/s and the mean MAE has been reduced by around 47%. Meanwhile, the two different lidar sensors employed at the two test sites produce somewhat different results. The test vehicle was observed at different instrument to vehicle distances and the speed varied depending on real traffic conditions. It is obvious that the performance of the three cases from Test Site 1 (RMSE 0.16 m/s,

MAE 0.14 m/s) is better than those from Test Site 2 (RMSE 0.22 m/s, MAE 0.18 m/s). The test vehicle was approximately 10–15 m from the sensor at Test Site 2; but the vehicle was only around 5 m from the sensor at Test Site 1. Also, the lidar sensor used at Site 2 had only 16 laser beams and accordingly the data density is lower. The above factors have resulted in a particular deficiency in acquiring points on the roof of the vehicle, which means there is insufficient detail in the image matching process during tracking refinement. Moreover, the estimated speeds from all the tracked vehicles range from 0 (stationary) to 13.4 m/s, covering a typical range of speeds range found in an urban city environment, which is limited by the speed restrictions and traffic flow conditions.

V. DISCUSSIONS

In the proposed framework, since vehicle detection has been simplified to a vehicle and nonvehicle classification problem after static point removal, it is realized by a simple rule-based classifier and two machine learning-based classifiers, SVM and RF. The training dataset is sufficient for a classifier intended to distinguish vehicles from other moving objects despite the small number of samples. On one hand, in urban environments, “other moving objects” mainly refer to pedestrians, cyclists, motorcyclists, and a few number of other false positives. Since their features are rather different to that of vehicles, they can be easily distinguished from vehicles. On the other hand, the training samples were selected at various locations to ensure coverage of a wider range so that the samples could be representative enough for the detection problem. Therefore, the resulting precision and recall are relatively high and this step should not be considered as a bottleneck for the workflow. However, the rule-based approach produced higher recall and was hence adopted for the further experiments. Initial trials of deep learning methods did not produce satisfactory results due to limited training data. Nevertheless, given the advancement of deep learning and the increasing number of open-source benchmark data, further attempts to adopt deep learning approaches are to be explored.

Two parameters in the centroid-based tracking stage were important. The first is the threshold that controls the range in which the detections are assigned to tracks, namely, the assignment gate. If the value is too small, some detections that should be assigned to a track might be overlooked. Otherwise, there will be false assignments. In this work, it was empirically set at 4 m, considering both the average vehicle speed and lidar sensor frame rate. The other parameter is the initialization threshold. If the association probability of a detection within the assignment gate is lower than the threshold, a new track will be generated. This parameter is critical to decide if a track should end when severe occlusion appears. For example, in heavy traffic flow, if the vehicle being tracked is occluded completely and consequently reobserved with an association probability lower than the initialization threshold, a new ID will be assigned to the subsequent detections. Thereafter, tracking refinement will resume for the new ID. Whereas, in light occlusion, in which the vehicle being tracked is partially occluded for a short period,

some of the clusters will be incomplete and lower association probability may arise. To keep the tracking continuous in this situation, a small value of 0.1 is assigned to the initialization threshold. In subsequent tracking refinement, if one image of a pair is partly affected, matching can still be conducted between them. Issues may arise that the matching accuracy is low and the estimated speed could be noisy. To tackle this issue, a smoothing algorithm is utilized in the final stage to filter out noise in the speed values.

The performance of the centroid-based tracking mainly relies on the vehicle detection results. In the framework, the tracking accuracy is not quantitatively assessed as all detected vehicles are correctly tracked at both Site 1 and Site 2, based on visual inspection. This mainly results from the high lidar sensor frame rate (0.1 s), which implies that clusters of a vehicle in two successive frames can be easily associated due to the small spatial distance between them. False tracks are mainly caused by false positives (nonvehicle road users, trees, or similar objects) in the detection stage. Tree like objects can be easily filtered by the length of the trajectories since they are almost stationary throughout time. However, some nonvehicle road users, such as pedestrians walking closely together along the road, cannot easily be discarded currently. A potential improvement to enhance the detection accuracy could be integrating semantic constraints, such as extracting road boundaries beforehand, to exclude pedestrians. As shown in the vehicle tracking analysis in Section IV, the tracking ranges of Site 1 and Site 2 are 45 and 18 m, respectively. The 32-beam lidar sensor has a longer maximum scanning range of 200 m, which naturally enables a longer tracking range than the VLP-16 with a maximum scanning range of 100 m. Nevertheless, the tracking range from the experimental data generated by VLP-16 was found to be slightly shorter than the 30 m reported in [10]. Occlusion by buses and longer instrument to vehicle distance are the two main factors accounting for the issue, particularly since the closest lane to the instrument at Site 2 was a bus lane.

The estimated speeds were validated against a reference system that is considered to provide a higher order of accuracy. The RMSE of the reference data was about one-tenth of that of the lidar data. In [10], speed validation was conducted by a test vehicle with an on-board diagnostics logger. The average absolute speed difference between speeds from lidar data and reference data, which is equivalent to MAE in our work, is as high as 0.639 m/s. In comparison, the average MAE of all the cases in the work reported here was 0.18 m/s. A more accurate reference system allowed full exploration of the capacity of lidar speed estimation.

One of the overarching aims of this research is to assess the practicality of using roadside lidar-based vehicle tracking for urban traffic sensing at the individual vehicle level. The following suggestions for large-scale implementations can be obtained based on this study.

- 1) As indicated through the experimental results obtained at both test sites, generally speaking the more laser beams the better and hence a laser scanner with as many beams as possible is recommended.

- 2) The effective detection and tracking range of a lidar sensor is significantly smaller than the maximum scanning range so the instrument should be installed at an appropriate distance to the road to maximize the coverage.
- 3) Considering that the height of common cars is smaller than 1.8 m, the lidar sensor should be installed higher than that to minimize occlusions.
- 4) Occlusions are unavoidable if only one sensor is used, hence multiple lidar sensors are recommended, if available.

This work has also shown that a speed accuracy of c. 0.2 m/s can be expected from high-precision 3-D lidar data and this could benefit more detailed and accurate traffic flow or behavior analysis. The high scanning frequency means that the resolution of the final speed estimation is also high and therefore accurate acceleration and deceleration values can be obtained. A principal drawback, however, is the inherent lack of RGB information, due to which the identification of vehicles is less straightforward than when using video camera data.

VI. CONCLUSION

This work presents an integrated vehicle tracking framework using roadside lidar data. Vehicle clusters were detected from the raw point clouds using a three-step schema in the first instance. Afterward, a centroid-based tracking procedure was applied to identify clusters for each vehicle. A refinement module was then used to improve the accuracy of vehicle speeds from 0.41 to 0.22 m/s, surpassing the accuracy reported in contemporary literature. Results demonstrate that lidar sensors are able to detect and track vehicles at a range of speeds in typical urban environments, proving the capability of lidar sensors for accurate speed monitoring. Moreover, it has shown that instruments with increased laser beams and longer scanning ranges will benefit vehicle detection and tracking, thereby improving estimated speed accuracy. It is concluded that lidar sensors can be employed for accurate vehicle counting and speed tracking in urban traffic sensing, but with limitations relating to short working range and occlusions. In future work, optimization of the algorithms is prioritized in order to realize near real-time traffic monitoring, necessary to open up the approach to further traffic monitoring applications. Moreover, multisensor utilization will be considered to address occlusion issues and deep learning-based vehicle classification will be added to the framework.

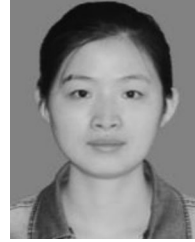
ACKNOWLEDGMENT

The authors would like to thank J. Goodyear, N. Harrap, M. Robertson, D. Bell, and M.-V. Peppia in the Geospatial Engineering group for all their help and guidance in data collection and processing.

REFERENCES

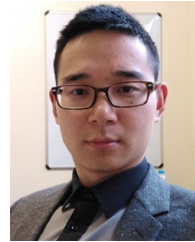
- [1] B. R. Gurjar, A. S. Nagpure, P. Kumar, and N. Sahni, "Pollutant emissions from road vehicle in mega city Kolkata, India: Past and present trends," *Indian J. Air Pollut. Control*, vol. 10, no. 10, pp. 18–30, 2010.
- [2] J. A. Pinto *et al.*, "Traffic data in air quality modeling: A review of key variables, improvements in results, open problems and challenges in current research," *Atmos. Pollut. Res.*, vol. 11, pp. 454–468, Mar. 2020.

- [3] V. Dangi, A. Parab, K. Pawar, and S. S. Rathod, "Image processing based intelligent traffic controller," in *Proc. Int. Conf. Control, Instrum., Commun. Comput. Technol.*, 2012, pp. 13–17.
- [4] H. Haritha and T. S. Kumar, "Survey on various traffic monitoring and reasoning techniques," in *Proc. Comput. Sci. On-line Conf.*, Apr. 2017, pp. 507–516.
- [5] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proc. IEEE 14th Int. Conf. Intell. Transp. Syst.*, 2011, pp. 1103–1108.
- [6] W. Xiao, B. Vallet, M. Brédif, and N. Paparoditis, "Street environment change detection from mobile laser scanning point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 107, pp. 38–49, Apr. 2015.
- [7] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Simultaneous detection and tracking of pedestrian from velodyne laser scanning data," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* vol. III-3, pp. 295–302, 2016.
- [8] J. Wu, "An automatic procedure for vehicle tracking with a roadside LiDAR sensor," *ITE J., Inst. Transp. Eng.*, vol. 88, no. 11, pp. 32–37, Nov. 2018.
- [9] J. Chen, S. Tian, H. Xu, R. Yue, Y. Sun, and Y. Cui, "Architecture of vehicle trajectories extraction with roadside LiDAR serving connected vehicles," *IEEE Access*, vol. 7, pp. 100406–100415, 2019.
- [10] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, "Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 68–87, Mar. 2019.
- [11] B. Lv *et al.*, "LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities," *IEEE Access*, vol. 7, pp. 79895–79907, 2019.
- [12] B. Lv, R. Sun, H. Zhang, H. Xu, and R. Yue, "Automatic vehicle-pedestrian conflict identification with trajectories of road users extracted from roadside lidar sensors using a rule-based method," *IEEE Access*, vol. 7, pp. 161594–161606, 2019.
- [13] J. Chen, S. Tian, H. Xu, R. Yue, Y. Sun, and Y. Cui, "Architecture of vehicle trajectories extraction with roadside LiDAR serving connected vehicles," *IEEE Access*, vol. 7, pp. 100406–100415, 2019.
- [14] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 29, no. 3, pp. 401–422, Mar. 2004.
- [15] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.
- [16] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 114, pp. 166–178, 2016.
- [17] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Depthcn: Vehicle detection using 3D-lidar and convnet," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–6.
- [18] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.
- [19] S. Shi, X. Wang, and H. Li, "Pointcn: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.
- [20] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proc. Robot., Sci. Syst.*, 2016.
- [21] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.
- [22] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.
- [23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.
- [24] M. Sualeh and G.W. Kim, "Dynamic multi-lidar based multiple object detection and tracking," *Sensors*, vol. 19, no. 6, Mar. 2019, Art. no. 1474.
- [25] T. Vu and O. Aycard, "Laser-based detection and tracking moving objects using data-driven markov chain monte carlo," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3800–3806.
- [26] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerasp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
- [27] W. Mei, G. Xiong, J. Gong, Z. Yong, H. Chen, and H. Di, "Multiple moving target tracking with hypothesis trajectory model for autonomous vehicles," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–6.
- [28] M. Allodi, A. Broggi, D. Giaquinto, M. Patander, and A. Prioletti, "Machine learning in tracking associations with stereo vision and lidar observations for an autonomous vehicle," in *Proc. IEEE Intel. Vehicles Symp. (IV)*, 2016, pp. 648–653.
- [29] F. Zhang, D. Clarke, and A. Knoll, "Vehicle detection based on lidar and camera fusion," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst.*, 2014, pp. 1620–1625.
- [30] A. A. Rachman, "3D-LIDAR multi objet tracking for autonomous driving," M.Sc. thesis, Delft Univ. Technol., Delft, The Netherlands, 2017.
- [31] A. Christodoulou, "An image-based method for the pairwise registration of mobile laser scanning point clouds," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 4, pp. 93–100, 2018.
- [32] L. Ding, A. Goshtasby, and M. Satter, "Volume image registration by template matching," *ImageVis. Comput.*, vol. 19, no. 12, pp. 821–832, 2001.



Jiaxing Zhang received the B.S. degree in surveying and mapping engineering from Central South University of Forestry and Technology, Changsha, China, in 2014, and the M.S. degree in photogrammetry and remote sensing from Central South University, Changsha, in 2017. She is currently working toward the Ph.D. degree in geospatial engineering with the School of Engineering, Newcastle University, Newcastle upon Tyne, U.K.

Her research interests include object detection and traffic monitoring based on lidar data and images.



Wen Xiao received the B.S. degree in geodesy and geomatics from Wuhan University, Wuhan, China, in 2010, the M.S. (cum laude) degree in geoinformatics from ITC, University of Twente, Enschede, The Netherlands, in 2012, and the Ph.D. degree in geoinformation science and technology from IGN, Université Paris-EST, Champs-sur-Marne, France, in 2015.

Since 2016, he has been a Lecturer then a University Research Fellow with Newcastle University, Newcastle upon Tyne, U.K. His research interests include 3-D mapping, laser scanning, photogrammetric computer vision, and their applications in the built and natural environment.

Dr. Xiao is currently a Council Member of UK Remote Sensing and Photogrammetry Society.



Benjamin Coifman received the Ph.D. degree in civil and environmental engineering from the University of California at Berkeley, Berkeley, CA, USA, in 1998.

He is an Associate Professor with the Ohio State University, Columbus, OH, USA, with a joint appointment with the Department of Civil, Environmental, and Geodetic Engineering and the Department of Electrical and Computer Engineering. His research emphasizes extracting more information about traffic flow both from conventional vehicle detectors and

emerging sensor technologies, then using this information to gain a better understanding of traffic phenomena.

Dr. Coifman was the recipient of several awards, including the ITS America Award for The Best ITS Research, an NSF CAREER Award, and the TRB Greenfields Award.



Jon P. Mills received the B.Sc. (Hons.) degree in surveying science and the Ph.D. degree in digital photogrammetry from Newcastle University, Newcastle upon Tyne, U.K., in 1993 and 1996, respectively.

He is currently a Professor of geomatics engineering with the School of Engineering, Newcastle University, where he leads the Newcastle Earth Observation laboratory (NEOlabor) in Geospatial Engineering. He is currently the Chair of Commission 1 on data acquisition for EuroSDR (European Spatial Data Research).