Vehicle Detection, Tracking and Behavior Analysis in Urban Driving Environments using Road Context

Shashwat Verma[1], You Hong Eng[1], Hai Xun Kong[1], Hans Andersen[2], Malika Meghjani[1],
Wei Kang Leong[1], Xiaotong Shen[1], Chen Zhang[1], Marcelo H. Ang Jr.[2], and Daniela Rus[3]

Abstract— **We present a real-time vehicle detection and tracking system to accomplish the complex task of driving behavior analysis in urban environments. We propose a robust fusion system that combines a monocular camera and a 2D Lidar. This system takes advantage of three key components: robust vehicle detection using deep learning techniques, high precision range estimation from Lidar, and road context from the prior map knowledge. The camera and Lidar sensor fusion, data association and track management are all performed in the global map coordinate system by taking into account the sensors' characteristics. Lastly, behavior reasoning is performed by examining the tracked vehicle states in the lane coordinate system in which the road context is encoded. We validated our approach by tracking a leading vehicle while it performed usual urban driving behaviors such as lane keeping, stop-and-go at intersections, lane changing, overtaking and turning. The leading vehicle was tracked consistently throughout the 2.3 km route and its behavior was classified reliably.**

I. INTRODUCTION

Autonomous driving technology for urban environments is at the forefront of academic and industrial research. Unlike autonomous driving on highways, urban environments pose unique challenging scenarios with more complicated road network and traffic signals. In addition, autonomous vehicles need to take into account the uncertainties involved in the interaction with other road users.

A key aspect in dealing with uncertainties in urban environments is to understand the intentions of other vehicles. Any misinterpretation of other vehicle's intentions will not only result in uncomfortable ride due to unnecessary jerks but also cause potential car accidents in the worst case.

The autonomous vehicles, hence, need to accurately analyze and predict the behavior of other vehicles to ensure safety and smooth navigation through the environment. The task of behavior analysis, however, is not independent of the autonomous vehicle's capability to detect and track other vehicles. Vehicle position could be obtained by a single measurement from sensors but its velocity has to be inferred through tracking a sequence of current and past measurements. Only then, behavior analysis can be performed based on the vehicle trajectory and road structures.

In this paper, we introduce and discuss a general framework for multiple vehicles detection, tracking, and behavior analysis based on the road context. The architecture of the proposed system is sub-divided into five subsystems: Vision-based detection and classification, Lidar-based clustering, sensors fusion and tracking, lane coordinate transformation and behavior analysis as shown in Fig. 1.

The vision system detects the vehicles in the image using widely known deep convolutional neural network, YOLO [1] (You Only Look Once). YOLO also provides bounding boxes of detected vehicles in the image frame, which are then utilized to estimate their positions in the map frame by assuming a planar road surface. The Lidar-based clustering system provides a set of potential clusters after extracting the point cloud from the road region. Data association and tracking are performed in the global map frame by the tracking system, where we combine measurements from both vision and Lidar according to the certainty of detection and sensors accuracy. Next, we calculate the detected vehicle position and velocity with respect to the lane coordinate system where behavior analysis could be performed under the context of the road structure.

The main contributions of this paper are twofold:

- A new approach in vehicle detection and tracking via fusing of Lidar and camera where the limitations of each sensor can be overcome. Robust vehicle detection from vision help us remove a lot of false positive if Lidar-based detection system was used. However, Lidar measurement is fused with vision detection to give a better tracking performance in term of a higher sampling rate and a better state estimation accuracy.
- A behavior analysis approach that utilized road geometry information from the prior map. The behavior analysis of the vehicle is performed under its current road context, using measures defined in the lane coordinate system.

II. RELATED WORK

In the past decades, detection and tracking of moving objects (DATMO) has been extensively studied by the mobile robotics community. Several of these techniques are applicable to autonomous driving and they have been widely used. A comprehensive literature review for autonomous driving is presented in [2] and [3]. In this section, we highlight some of the recent work on detection and tracking using sensor fusion for autonomous driving.

[1] Shashwat Verma, You Hong Eng, Hai Xun Kong, Malika Meghjani, Wei Kang Leong , Xiaotong Shen and Chen Zhang are with the Singapore-MIT Alliance for Research and Technology, Singapore { shashwat, youhong, haixun, malika, weikang, xiaotong, zhangchen}@smart.mit.edu

[2] Hans Andersen and Marcelo H. Ang Jr. are with the National University of Singapore, Singapore { hans.andersen}@u.nus.edu, mpeangh@nus.edu.sg

[3] Daniela Rus is with the Massachusetts Institute of Technology, Cambridge, MA, USA rus@csail.mit.edu

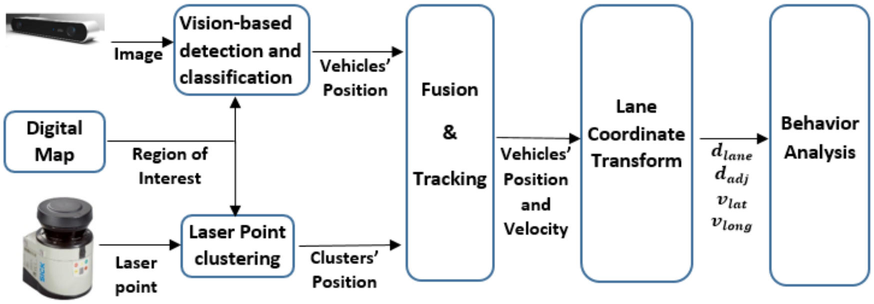

Fig. 1. General architecture of the detection, tracking and behavior analysis.

In [4], the authors used Lidar to detect, track and classify pedestrians and vehicles in the Lidar frame of reference. The positions of detected objects were then transformed into the image frame to find the region of interest for the vision classifier. A sum decision rule was used to combine the result of classification from both Lidar (GMM classifier) and vision (AdaBoost classifier).

Similarly, authors in [5] also considered Lidar as the main sensor in their sensor fusion configuration. The moving objects were first identified using inconsistencies between free and occupied cells within the map. If an occupied cell is detected on a location previously set free, then it belongs to a moving object. The region of interest provided by the Lidar detection system is used as the input for image classification using Histogram of Gradient (HoG) descriptor [6].

Cho et al. [7] proposed a new tracking system consisting of two parts: sensor layer and fusion layer. In the sensor layer, detection and classification were performed for each sensor independently, and their results are combined in the fusion layer which gives the final outcome. This layered system offers a separation between actual sensing hardware and data processing for detection and tracking. In [8], the authors used Lidar for obstacle detection and monocular camera for classification. Sensor fusion is again performed at a higher level of abstraction based on detection certainty and sensors accuracy.

In contrast to previous work discussed above, our sensor fusion approach relies primarily on vision rather than Lidar. We make use of recent advancements in computer vision and deep learning for object detection in real-time and with high accuracy. We perform data association in the global map frame, where we combine high variance vision-based position estimation with the low variance laser point clustering to achieve better position estimation.

Behavior analysis to classify the intention of the surrounding vehicles is a relatively new research area compared to the vehicle detection and tracking. Most of the related works focus on a very specific maneuver like lane changing or overtaking, and the experiments are mainly conducted in highway environments. In [9], the authors proposed a new approach based on Support Vector Machine (SVM) and Bayesian filtering to predict lane change behavior on the highway. Similarly, [10] used one-class SVM to detect dangerous lane change maneuvers in the dataset extracted from 2nd Strategic Highway Research Program (SHRP 2) [11]. An object-oriented Bayesian network was developed in [12] to classify maneuvers including merging and object following in structured highway scenarios.

Furthermore, many previous studies did not utilize road context information available from the digital map. As pointed out by [13], the precise roadway geometry information can improve the performance of the behavior reasoning algorithms by introducing constraints to reduce the complexity of the problem. Our approach is motivated by their research and we conducted our experiments in a real urban driving environment instead of a vehicle test track.

III. Methodology

A. Vision-based object detection and classification

In order to achieve real-time performance for object detection, we use YOLO, a convolution neural network architecture. We trained our network using a hybrid dataset which comprises of images from KITTI dataset [14] and an endemic dataset from Singapore roads. A significant increase in accuracy is obtained by training both datasets concurrently. As our primary interest is to detect other road users, we only detect three different classes of objects, namely pedestrian, bike and vehicle. For the sake of clarity, we only discuss vehicle detection in this paper although our approach is applicable to the remaining classes of objects. Offline network inference on NVIDIA Titan X GPU shows that the system can be executed at 40 fps with mAP of 74.1. However, during real-time execution on the vehicle's PC, equipped with less powerful NVIDIA GTX 1070 GPU, we execute the network inference at 5 fps to compensate for the computational resources required to run other processes concurrently.

We estimate the position of detected vehicles in the global map coordinate system by projecting the coordinates from the image frame using the pin-hole model and flat-ground assumption [15].

B. Point Cloud Clustering

The Lidar range information is first pooled together into a point cloud. These points are then filtered using a binary road mask, which filters out the points that correspond to objects that are not on the road, e.g. vegetations, buildings, etc. The filtered points are then clustered using DBSCAN Algorithm [16] to generate an array of cluster centroids at the rate of 30 Hz. The centroid position is then published in the global map coordinate system.

C. Sensor Fusion and Tracking

1) Data Association: We have implemented a generic data association framework for sensor fusion that is agnostic to both the number of sensors as well as the type of sensor being used.

We define the state of our world at any time instant as a set of tracks Ψ_t. These tracks store the information of each vehicle as vehicle state vector. To correctly track obstacles, it is essential to determine whether there is a need to start a new track or the detected object already exists in the track system. We model this problem of assignment as a bipartite graph G(V, E) where obstacles and tracks are represented as vertices v ∈ V, and edges e ∈ E represent cost between two vertices. This problem is a typical example of a bipartite graph as each vehicle can only be assigned to a single track. The global assignment cost is defined as

$$\sum_{i}^{N}\sum_{j}^{M} c(i,j) \tag{1}$$

where

$$c(i,j) = \left(\frac{x_i - x_j}{\sigma_x}\right)^2 + \left(\frac{y_i - y_j}{\sigma_y}\right)^2 + \log\left(\sigma_x \sigma_y\right) \tag{2}$$

and (x_i, y_i) is the position of the detected vehicle, (x_j, y_j) are the predicted position of the object based on track j and (σ_x, σ_y) are the standard deviation of sensor's detection. Each of these quantities is considered in the global map coordinate system. The cost $c(i,j)$ is essentially a weighted distance by discounting the measurement that has large uncertainty (variance) in a particular dimension. For example, the depth estimation from the vision has a larger uncertainty compared to its lateral estimation, and hence the measurement from the depth projection axis which has a larger variance, are discounted in the cost calculation during data association. There are 6 possible cases for assigning N vehicles to M tracks:

- $N = M$, observed vehicles match exactly the existing tracks
- $N = M$, there is mismatch among the observed vehicle with existing tracks
- $N > M$, all M tracks have an observation
- $N > M$, not all M tracks have an observation
- $M > N$, all N vehicles have an existing track
- $M > N$, not all N vehicles have an existing track

A cost matrix is then populated using (2). Given the cost matrix, Munkres algorithm [17] is used to find the appropriate assignment by minimizing the overall cost of the assignment. Then, each association is checked for its validity by comparing the cost with a minimum cost threshold. If the assignment cost is less than the threshold, the assignment is considered valid. A new track is only generated when the assignment cost is greater than the threshold, and the observation source is the vision-based detection system. We discard Lidar-based observations with assignment cost that is greater than the threshold in order to avoid false positives.

2) Filtering System: Kalman Filter (KF) and its variants have been widely used for vehicle tracking given their low computational cost and practically adequate accuracy. Once a track has been initialized, we use KF to propagate the vehicle states using the constant velocity model and update the predicted state with the incoming new measurements. In the following section, we define states and matrices used in our filtering system, where standard KF equations are used to update the state.

i-th detected vehicle's state vector $\mathbf{X}$ at time t consists of the vehicle's position (x, y) and vehicle's velocity $(\dot{x}, \dot{y})$ defined in the global map coordinate frame.

$$\mathbf{X_t^i} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$$

and the associatied measurement vector $\mathbf{Y}$ consists of the vehicle's position (x, y) estimated from sensors.

$$\mathbf{Y_t^i} = \begin{bmatrix} x & y \end{bmatrix}^T$$

The transition matrix

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is used in predicting vehicle's movement by assuming constant velocity model as given above, where Δt is the time difference between two consecutive states being considered. While the observation matrix can be written as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

because only the vehicle's position (x, y) is observable from our sensors.

The state is initialized using the vehicle's position when it is first detected and has zero velocity, with the measurement error matrix based on the sensor characteristics

$$\mathbf{R_V} = \begin{bmatrix} \sigma_x^v & 0 \\ 0 & \sigma_y^v \end{bmatrix}; \mathbf{R_L} = \begin{bmatrix} \sigma_x^l & 0 \\ 0 & \sigma_y^l \end{bmatrix}$$

with R_V, R_L representing the measurement error matrix for vision and Lidar, respectively. Lidar has a better accuracy in term of position estimate when compared to camera, thus σ^l is smaller than σ^v in order of magnitude. Internally, σ_x^l is equal to σ_y^l as the measurement accuracy for both axes is indifferent for Lidar. On the other hand, σ_x^v is larger than σ_y^v for vision because depth estimation is based on flat ground assumption which has a larger uncertainty compare to the lateral estimation. When applying (2) for the vision measurement update, the difference in distance for the longitudinal axis x and the lateral axis y have to be expressed in the ego vehicle body-fixed coordinate system such that distance differences are discounted correctly. In our experiment, we set $\sigma_x^l = \sigma_y^l = 2$ and $\sigma_x^v = 10, \sigma_y^v = 5$.

3) Track Management System: The environment state at time t with m vehicles being tracked simultaneously, is represented as Ψ_t in (3). Specifically, the track of i^{th} vehicle that contains all the previously detected states of that vehicle up to time t, is given in (4), where X_t^i denotes the state of the vehicle i at time t.

$$\Psi_t = [T_t^1, T_t^2, .., T_t^m] \tag{3}$$

$$T_t^i = [X_t^i, X_{t-1}^i, ..X_0^i] \tag{4}$$

The track management system allows the assignment system to query the state of each obstacle. The system also keeps a count of the number of times a track was marked unassigned consecutively. The track is deleted if this number crosses a particular threshold α and the count is reset to zero if there is an assignment.

The value of α is selected based on the expected duration that a sensor fails to detect the vehicle in the existing track consecutively and also its sampling rate. For example, Lidar might fail to detect the vehicle temporarily while the ego vehicle is traversing over a speed bump, causing pitching motion on the 2D Lidar. Similarly, camera might fail to detect vehicle when there is a sudden change in lighting condition. We set $\alpha = 15$ for Lidar and $\alpha = 10$ for vision.

D. Lane Coordinate System

To predict vehicle motions and behaviors accurately, the current dynamic state and trajectory of the other vehicle should be analyzed in the context of its current road structure because the same vehicle motion under different road structure will mean different intentions. In this section, we introduce the lane coordinate system that enables us to perform behavior reasoning of the other vehicles.

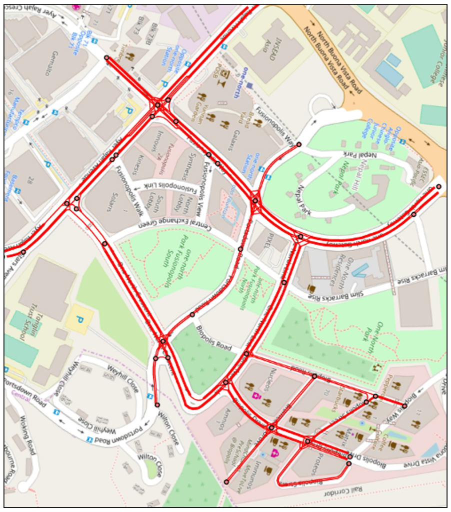

Fig. 2. The roadway geometry information is extracted from the prior digital map. The center of the lanes are marked with a series of connected red line segments. The map shows one section of the one-north region, which is selected as a testbed for autonomous vehicle (AV) technology and deployment in Singapore.

The road geometry information is obtained from SimMobility [18] which is a simulation platform for analyzing transportation systems using real road networks. The road and public transportation network within SimMobility are obtained from NAVTEQ [19]. An overlay of the SimMobilitys road geometry information illustrating the lane center is presented in Fig. 2.

The roadway geometry used in this experiment is described based on a piecewise linear model. The center of the lane is represented by multiple line segments, each line segment consists of two terminal points in the global map coordinate system as shown in Fig. 3. The road curvature is approximated by a connected series of line segments. When the curvature is large, more lane segments can be used to reduce the approximation error to a reasonable level. Given a curve composed of line segments, the Ramer-Douglas-Peucker algorithm is useful in finding a similar curve with fewer points by minimizing the maximum distance between the original curve and the simplified curve.

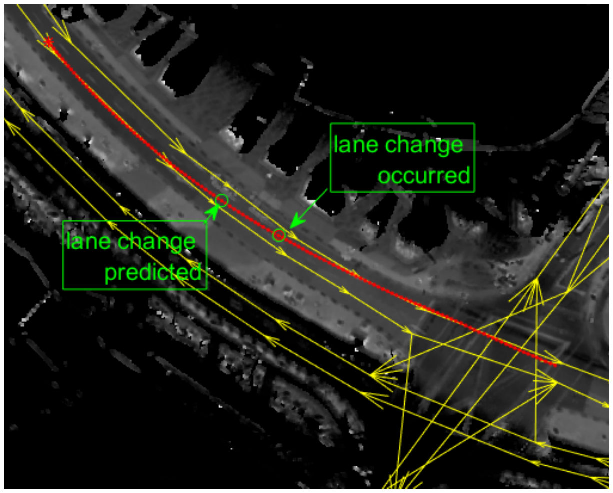

Fig. 3. The center of the lane is represented as a line segment with two points in the global map coordinate system. A series of connected line segments form the road networks which are shown as a yellow arrow pointing from the starting point to the ending point.

Originally, the lane centers are defined in the Universal Transverse Mercator (UTM) coordinate system. We transform the lane segments into our global map coordinate frame using the 2D Helmert-Transformation which require 4 parameters (2 for translation, 1 for rotation and 1 for scaling factor). The parameters are found using least square fitting between a set of UTM points collected using onboard GPS sensor and their corresponding coordinates in the global map coordinate system using ego vehicle localization system.

Ego vehicle localization is achieved using Monte-Carlo based method described in [20]. A SICK LMS 151 LIDAR, which is mounted with 15^o tilted down angle, is fused with wheel odometry and an inertial measurement unit (IMU) to localize the vehicle within the pre-recorded map.

It is important to perform behavior reasoning within the lane coordinate system as this provides the road structure and context to the solution. First, vehicles could only drive within the road boundary and follow the center of the lane most of the time. Second, the lane change behavior only happens at the certain boundary of the road.

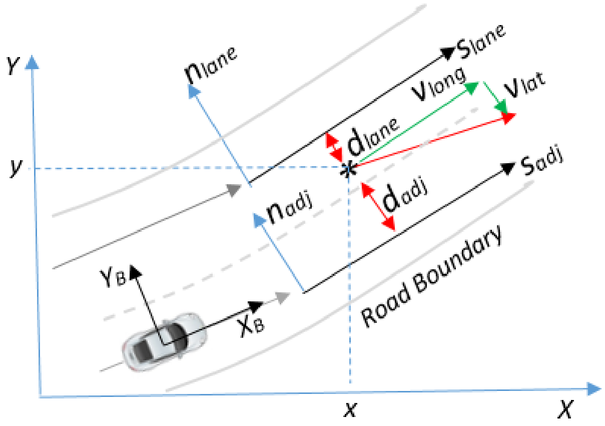

Fig. 4. Lane Coordinate System. The diagram explains the setup of three coordinate systems: global map coordinate system, ego-vehicle body-fixed coordinate system, and lane coordinate system.

Fig. 4 shows three coordinate systems associated with the behavior analysis: global map coordinate system (X, Y), ego vehicle body-fixed coordinate systems (X_B, Y_B) and lane coordinate system (s_{lane}, n_{lane}).

1) Global Map Coordinate System (X, Y): The axes of the global map coordinate system consist of X and Y axis in metric units. The ego vehicle localization system provides its position and heading states in this frame. The lane segments are also defined in this frame.
2) Ego Vehicle Body-Fixed Coordinate System (X_B, Y_B): The position of the target vehicle (x, y) are first detected and measured in the ego vehicle body-fixed coordinates as the sensors are mounted on the vehicle body frame. The axes of the body Cartesian coordinates are composed of the longitudinal axis (X_B) and lateral axis (Y_B) in metric units.
3) Lane Coordinate System (s_{lane}, n_{lane}): Lane coordinate system is not a fixed coordinate frame but change according to the position of the target vehicle. The lane segment with the minimum distance to the target vehicle will be selected.

The lane segment is defined by a starting point and an ending point. It is a directional line segment pointing from the starting point to the ending point which defines the x-axis of the lane coordinate system, denoted as S_{lane}, whereas the orthogonal axis is denoted as n_{lane} as shown in Fig. 4. We will now define three measures in this frame. First, we have the orthogonal distance between the target vehicle and the segment, denoted as d_{lane}. Second, we decompose the vehicle speed v into two orthogonal components, namely longitudinal speed v_{long} and lateral speed v_{lat}.

The distance between a point and a line segment is calculated using the method discussed in [21]. In order to find the segment with the minimum distance to the target point, one needs to iterate through all the segments which is computationally expensive. In order to improve the computational efficiency, we store in the database for every segment, a set of segments that it can transit to. For example, a vehicle can only transit to the segment in front and back and adjacent left or right if there is any. In this way, once a target vehicle is initialized with a particular segment, we will only calculate the distance between the target vehicle with the current segment and those that it can transit too, which is a much smaller subset of segments. A transition matrix with a binary value is used to indicate which segments are the potential transit candidates given the current segment.

Besides the transition matrix, we have also created an adjacent lane matrix which is used to indicate the adjacent lane segment given the current lane segment. One segment could have up to two adjacent segments, which happen to the center lane when there are three or more lanes. However, based on the position of the target vehicle, only one adjacent lane will be associated at one particular instance, it is indicated as s_{adj} and n_{adj}. In this frame, we calculate the minimum distance between the target vehicle and the adjacent lane, denoted as d_{adj}.

The transformation from the ego vehicle body-fixed coordinate to global map coordinates is performed based on the ego vehicle position and heading information from the localization system. Once the target vehicle is transformed into the global map coordinate, d_{lane}, d_{adj}, v_{lat} and v_{long} can be computed.

E. Behavior Analysis

Driving in urban environments include frequent stop-and-go traffic, queuing at traffic signals, and lane changing. Lane change is one of the most dangerous maneuvers compared to others [10], and therefore it is beneficial to be predicted it in advance. In this section, we categorize the behavior of a detected vehicle into 3 categories: stopping, lane keeping, and lane changing.

Fig. 5 illustrates how the d_{adj} and d_{lane} change with time as the lane change occurs at time $t = 0$. As the target vehicle is leaving its current lane, the distance to its initial lane center d_{lane} increases, and the distance to the adjacent lane decreases. Lane change occurs at the instance when the target vehicle's associated lane is changed, i.e. when it is nearer to the adjacent lane than to its initial lane. As the target vehicle merges to the lane, d_{lane} reduces and d_{adj} increases.

To prevent a collision, lane change maneuver of the surrounding vehicles has to be predicted in advance. An example trajectory showing where a lane change is predicted and occurred is shown in Fig. 3. In order to predict lane change, we calculate the time to lane change T_{LC},

$$T_{LC} = \frac{d_{adj} - d_{lane}}{v_{lat}} \tag{5}$$

by assuming a constant lateral speed model when a vehicle changes lane. Fig. 5 (bottom) shows that the T_{LC} is reduced as the vehicle changes lane and it goes to zero when the lane change occurs. We then map the T_{LC} to the probability of lane change as:

$$P_{LC} = e^{-\lambda * T_{LC}} \tag{6}$$

where λ is a positive constant. As shown in Fig. 5, the probability of the lane change P_{LC} increases from zero to almost one just before the lane change occurs and back to zero after that.

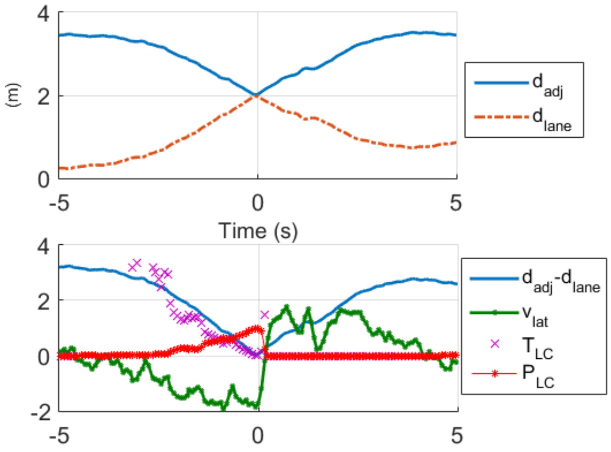

Fig. 5. The lane change occurs at time zero. Top figure shows the "magic" cross created by d_{adj} and d_{lane} which indicates lane change happens at the crossing point; the bottom figure shows v_{lat} is about constant during lane change and the value of T_{LC} and P_{LC}.

IV. Experiments

Tests were performed in real driving conditions at one-north, a science and business park in Singapore. To evaluate the performance of the perception system, the ego vehicle followed and tracked a leading target vehicle that performed typical maneuvers such as lane changing, stop-and-go at intersections, turning at intersections and overtaking another vehicle. The fusion algorithm was able to detect, track and classify the behavior of the leading vehicle for the entire 2.3 km route with a total of 16537 numbers of detection with an average sampling rate of 35Hz. The entire vehicle states and distance between the target vehicle and the ego vehicle are shown in Fig. 6. Detection rate of the target vehicle using Lidar and camera are tabulated in Table I. A video that highlights the capabilities of our method can be viewed at `https://youtu.be/s_rvaHvTn64`. In order to give a better understanding of the result, we will discuss 3 specific sequences of the target vehicle:

- Target vehicle performed lane change twice on a straight two-lanes road.
- Target vehicle overtook a parked vehicle.
- Target vehicle made a 90^o turn.

A. Test 1

In test 1, we show the perception system was able to track the target vehicle when it was performing lane changes. The behavior analysis module was able to predict the lane changing behavior 1.0 s and 0.6 s in advance for the first and second lane changes respectively. Fig. 11:A shows four selected images when the target vehicle is making the first lane change. Lane change behavior is predicted when $P_{LC} > 0.4$, where 0.4 is a threshold selected to balance between early detection and false positive. The condition is satisfied during 148.3-149.3 s and 154.0-154.6 s (see Fig. 7). Fig. 8 shows the trajectory of the target vehicle on the global map coordinate system with emphasis on the lane change maneuvers which are color coded in red. The algorithm classifies the behavior as lane keeping (LK) after the lane change occurred as the vehicle maintain lateral velocity to merge to the target lane. The time to lane change T_{LC} back to the original lane is big and hence the P_{LC} is small.

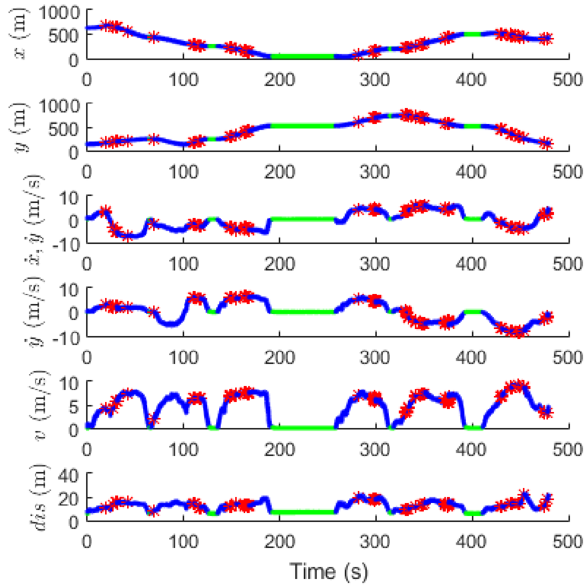

Fig. 6. Tracking results of the target vehicle for the entire 2.3 km route with 16537 numbers of detection in 479 s. The maximum speed of the target vehicle is 9.4 m/s and the maximum distance from the ego vehicle is 22.5 m. The behavior is color coded blue for lane keeping, red for lane changing and green for stopping.

TABLE I
Overall detection rate for Lidar and vision

	Detection (count)	Positive detection (count)	Percentage (%)
Lidar	14332	13846	96.6
Camera	2205	2047	92.8

B. Test 2

In test 2, we show the fusion system was able to track both the target vehicle and a parked vehicle when the target vehicle overtaking the parked vehicle. The behavior analysis module was able to predict the lane changing behavior 0.6 s in advance. Fig. 11:B shows four selected images when the target vehicle is overtaking the parked vehicle. The parked vehicle was detected in the 2nd image although it was partially occluded by the target vehicle. On the 3rd image, the target vehicle was crossing the lane divider. Fig. 9 shows the trajectory of the target vehicle and the parked vehicle on the global map coordinate system with emphasis on the lane change maneuvers which are color coded in red and the stopping behavior which is color coded in green. A vehicle is classified as stopping when its speed is smaller than a minimum speed threshold v_{min}, which is set to 0.1 m/s in this experiment.

C. Test 3

In test 3, we show the fusion system was able to track the target vehicle making a 90^o sharp turn. The behavior analysis module correctly classifies the behavior of the target vehicle as lane keeping (LK) because given the road context, the position of the vehicle was far away from the adjacent lane and the lateral speed in lane coordinate system is relatively small. This shows the importance of the road context in behavior reasoning. Fig. 11:C shows four selected images when the target vehicle is making the sharp turn. The vision system was able to detect the vehicle consistently although the vehicle was almost out of the frame in image C3. Fig. 10 shows the trajectory of the target vehicle on the global map coordinate system. The fusion system did not receive measurement update of the target vehicle from Lidar for a period of time. This is due to the pitching motion of the ego vehicle causing the target vehicle out of the 2D plane of Lidar scanning. However, the vision system was able to detect the vehicle consistently and thus the fusion system was able to maintain the track of the vehicle.

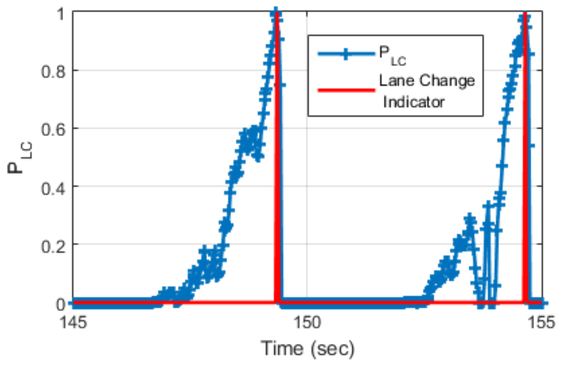

Fig. 7. Probability of lane change versus time in test 1.

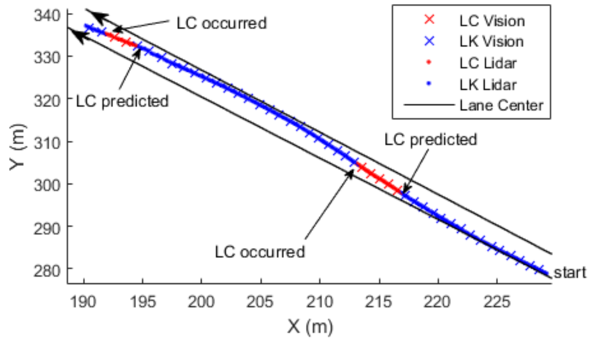

Fig. 8. Trajectory of target vehicle performed Lane Changing (LC) twice in a straight two-lanes road.

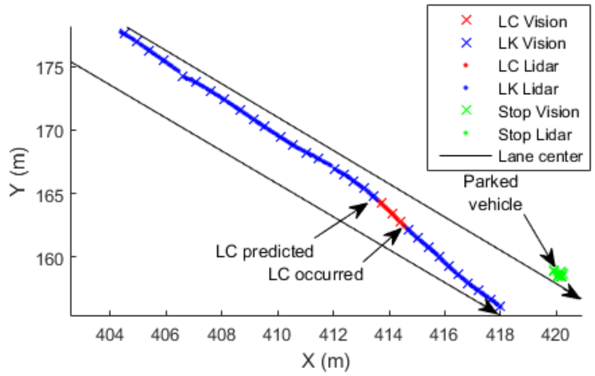

Fig. 9. Trajectory of target vehicle overtaking a parked vehicle.

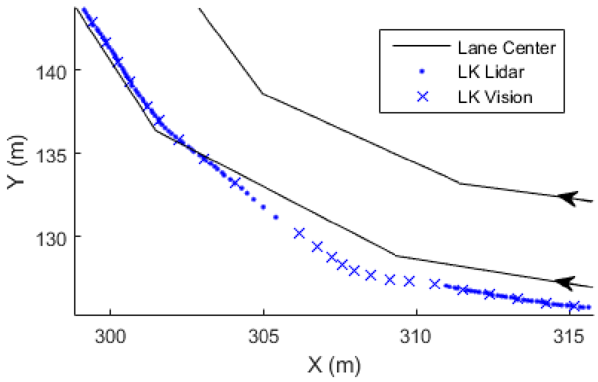

Fig. 10. Trajectory of target vehicle making a 90^o sharp turn.

V. Conclusions

The paper presents a sensor fusion methodology that combines vision and Lidar to robustly detect and track vehicles in the complex urban scenarios. We apply deep learning techniques to detect vehicles from camera image and improve its position estimate by fusing Lidar information. State estimation, data association, and track management are performed in global map coordinate system by considering the characteristics of each sensor. This approach shows its effectiveness in tracking a target vehicle consistently for a 2.3 km route. We have also performed behavior analysis of detected vehicles using road context. By examining the vehicle states in the lane coordinate system, we are able to reliably classify the behavior of the tracked vehicles into stopping, lane keeping and lane changing. This brings us one step closer towards reliable and safe autonomous driving in urban environments.

Acknowledgment

This research was supported by the Future Urban Mobility project of the Singapore-MIT Alliance for Research and Technology (SMART) Center, with funding from Singapore's National Research Foundation (NRF).

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE*

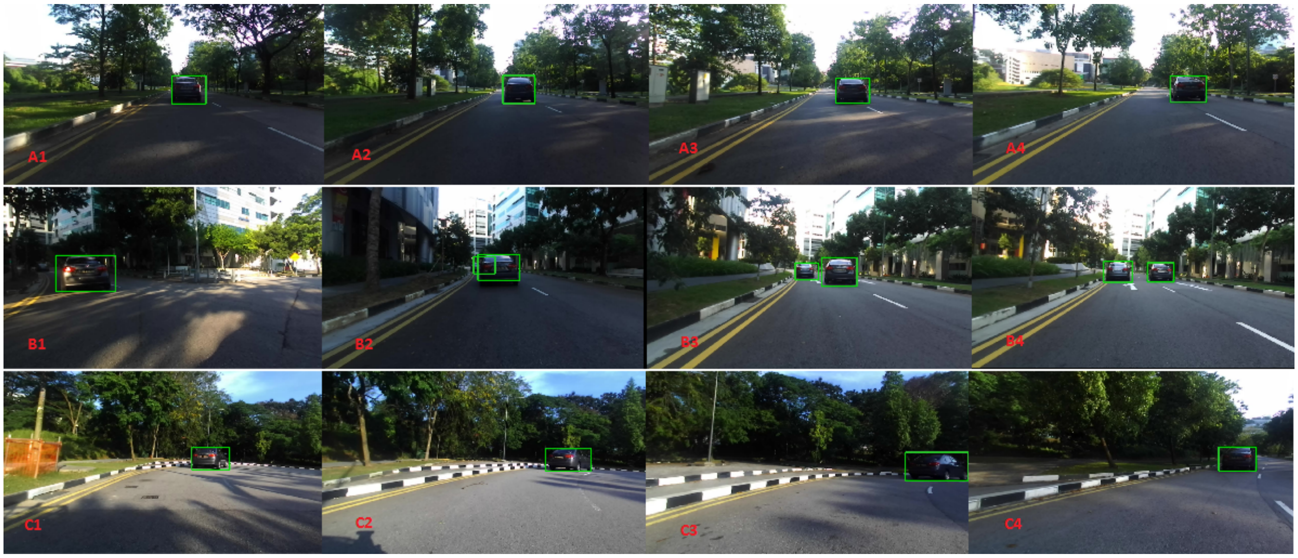

Fig. 11. A: Selected image sequences showing the target vehicle performed lane changing in a straight two-lanes road. B: Selected image sequences showing the target vehicle overtaking a parked vehicle. C: Selected image sequences showing the target vehicle making a 90^o turn.

Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[2] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[3] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[4] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 1044–1049.

[5] R. O. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 525–534, 2016.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[7] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–1843.

[8] F. Garcia, D. Martin, A. de la Escalera, and J. M. Armingol, "Sensor fusion methodology for vehicle detection," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 123–133, 2017.

[9] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797–802.

[10] S. Ramyar, A. Homaifar, A. Karimoddini, and E. Tunstel, "Identification of anomalies in lane change behavior using one-class svm," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 004 405–004 410.

[11] K. L. Campbell, "The shrp 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety," *TR News*, no. 282, 2012.

[12] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 3, pp. 19–31, 2012.

[13] K. Jo, M. Lee, J. Kim, and M. Sunwoo, "Tracking and behavior reasoning of moving vehicles based on roadway geometry constraints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 460–476, 2017.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[15] Y.-C. Kuo, N.-S. Pai, and Y.-F. Li, "Vision-based vehicle detection for a driver assistance system," *Computers & Mathematics with Applications*, vol. 61, no. 8, pp. 2096–2100, 2011.

[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[17] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[18] M. Adnan, F. C. Pereira, C. M. L. Azevedo, K. Basak, M. Lovric, S. Raveau, Y. Zhu, J. Ferreira, C. Zegras, and M. Ben-Akiva, "Simmobility: A multi-scale integrated agent-based simulation platform," in *95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record*, 2016.

[19] C. L. Azevedo, K. Marczuk, S. Raveau, H. Soh, M. Adnan, K. Basak, H. Loganathan, N. Deshmunkh, D.-H. Lee, E. Frazzoli *et al.*, "Microsimulation of demand and supply of autonomous mobility on demand," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2564, pp. 21–30, 2016.

[20] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 1554–1559.

[21] D. H. Eberly, *3D game engine design: a practical approach to real-time computer graphics*. CRC Press, 2006.