

L-Shape Model Switching-Based Precise Motion Tracking of Moving Vehicles Using Laser Scanners

Dongchul Kim, *Member, IEEE*, Kichun Jo^{ID}, *Member, IEEE*, Minchul Lee^{ID}, *Student Member, IEEE*, and Myoungho Sunwoo^{ID}, *Member, IEEE*

Abstract—Detection and tracking of moving objects is one of the most essential functions of autonomous cars. In order to estimate the dynamic information of a moving object accurately, laser scanners are widely used for their highly accurate distance data. However, these data only represent the surface of an object facing the sensor and changes the appearance of an object over time. This change produces unexpected tracking errors of estimated dynamic states. In this paper, in order to minimize the tracking error caused by appearance changes, a tracking algorithm based on L-shaped model switching is proposed. The suggested algorithm is validated in real traffic experiments where position, velocity, and heading angle error were measured by using precise GPS. The L-shape tracking algorithm successfully mitigated the effect of appearance changes and improved estimation performance.

Index Terms—Vehicle tracking, L-shape model, modeling switching filter, laser scanner.

I. INTRODUCTION

RECENTLY, attention on the development of highly safe and convenient vehicles has been drawn not only by customers in the market but also by governmental regulations. These social concerns are a driving force in developing an intelligent vehicle that provides information regarding the driving environment to the drivers, and thus enables assisted safe and convenient driving. Finally, the evolution of intelligent vehicle technology will realize an autonomous car that drives itself to the destination without human intervention, which is the ultimate goal of the intelligent vehicle.

Manuscript received January 14, 2017; revised September 5, 2017; accepted October 19, 2017. Date of publication December 6, 2017; date of current version February 1, 2018. This work was supported in part by the BK21 Plus Program through the Ministry of Education, South Korea, under Grant 22A20130000045, in part by the Industrial Strategy Technology Development Program under Grant 10039673, Grant 10060068, and Grant 10042633, in part by the System Industrial Strategic Technology Development Program under Grant 10079961, in part by the International Collaborative Research and Development Program through the Ministry of Trade, Industry and Energy (MOTIE Korea) under Grant N0001992, and in part by the Energy Resource Research and Development Program through the Ministry of Knowledge Economy, South Korea, under Grant 2006ETR11P091C. The Associate Editor for this paper was P. Zingaretti. (*Corresponding author: Kichun Jo*)

D. Kim, M. Lee, and M. Sunwoo are with the Automotive Control and Electronics Laboratory, Department of Automotive Engineering, Hanyang University, Seoul 133-791, South Korea.

K. Jo is with the Driving Assistance Research, 93012 Bobigny, France (e-mail: kichun.jo@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2017.2771820

In order to realize successful autonomous driving, four basic systems of localization, perception, path planning, and control are required [1], [2]. Each system is operated sequentially as follows. First, the localization system estimates the current position of the autonomous car with high accuracy [3]–[6]. Next, the perception system recognizes the surrounding environment [7], [8]. Then, the planning system generates a path based on the environment information from the perception system [9], [10]. Finally, the control system controls the autonomous car to follow the generated path.

Among these systems, the perception system has been researched using laser scanners to recognize moving objects such as vehicles, pedestrians, and motorcycles. Generally, the object tracking method using laser scanners consists of two procedures; object detection and object tracking. The detection procedure extracts a feature from segmented point groups [11], [12] and generates a bounding box with the center point as the object position. After that, the tracking stage manipulates the detection result with dynamic state estimation and a data association filter. As the estimation method, a Kalman filter (KF) and its variations (EKF, UKF) [13]–[17] are widely used. With the estimation method, one of the data association filters is combined, such as nearest neighbor (NN) [13], [14], probabilistic data association (PDA) [18] and its variations (joint PDA (JPDA) [13], [19], integrated PDA (IPDA) [20], and joint IPDA (JIPDA) [21]). The listed tracking algorithms are generally used for many tracking applications but they suffer in cluttered environments. To improve multi-target tracking performance in cluttered environments, multiple hypothesis tracking (MHT) [22], [23], particle filter (PF) [24], [25], and random finite set (RFS) [26] have been proposed. MHT provides multiple hypotheses for tracking the measured association and combines all probabilities in the hypothesis. PF is a Bayesian filter based on samples named “particles” and can represent uncertainty through the probabilistic distribution of itself. RFS uses the Bayesian framework, taking into account the probability of association with the number of tracks and observations. The algorithms show powerful estimation performance in multi-target tracking problems in cluttered environments but they require massive computation power to achieve this powerful performance.

The laser scanner is an optical device that measures the distance to objects by an active light source. It generates a single or multiple laser beam(s) onto a rotating mirror to scan

a two-dimensional plane or three-dimensional space. Unlike radar, the performance of the laser scanner deteriorates in rainy and foggy situations due to the scattering and refraction of the laser beam. Despite its sensitive performance in these weather conditions, most highly autonomous cars adopt the laser scanner because it provides dense range information with high resolution and accuracy.

However, the laser scanner provides only distance measurements from the surface of the objects facing the sensor. Thus, the laser scanner cannot see the entire shape of the object and the appearance of the point cloud changes over time according to the viewpoint. This appearance change problem can cause unexpected motion estimation errors in the tracking algorithm.

In order to solve this problem, a vehicle tracking method that uses vague line ends [27] was proposed. This method is based on line tracking, but it has an assumption that the line cannot represent an exact size. Thus, the end of the line has the “vague” property for reflecting the uncertainty of the line length. These vague line ends are tracked with large longitudinal position covariance to suppress large rapid motion change. Although this method can suppress longitudinal motion error from parked vehicles successfully, the reduced error is only limited to the longitudinal motion of the track.

Another approach to solving the appearance change problem is by using prior knowledge [28]. In this approach, pre-defined types of shapes (I-shape, IS-shape, L-shape, C-shape, and E-shape) are applied to the tracking algorithm with a different dynamic model. This method is effective in tracking moving objects that have various types of dynamics. However, in the case of misclassification, applying the misclassified dynamic model is not suitable. In addition, tracking performance can become unstable because changes in the shape type can lead to sudden dynamic model changes.

Another tracking algorithm to compensate for appearance changes is geometry model-based vehicle tracking [8]. This algorithm was used by Junior of Stanford University in the DARPA Urban Challenge in 2007. This algorithm introduced a geometry model using anchor point with a measurement model for applying particle filters. The geometry model estimates the length and width of the box and the relative position of the anchor point. By using the geometry model, every point information is evaluated by the measurement model to calculate the likelihood of each particle. Since this calculation should be performed many times, this method requires massive computational power proportional to the number of particles.

The papers [29], [30] describe an L-shape fitting method using laser scanner data. The authors evaluated the method using statistical analysis of point data with computational efficiency. The paper [31] proposed an optimal fitting method for rectangles, with vehicles represented as oriented 2D bounding boxes. These methods only focus on fitting the object and target tracking problems are not considered.

The paper [32] suggests an L-shaped edge model for associating laser scanner data with camera and radar information. The paper [33] describes a 3D geometry model with heterogeneous sensors while the paper [34] presents a detailed shape model capable of representing non-convex object surfaces.

However, these methods show complex structures and require massive computation power.

In this study, we propose a shape invariant tracking algorithm based on L-shape model switching with a simple structure. We adopted an L-shape feature rather than a box model and U-shape feature for the following reasons: (1) The L-shape feature can represent a shape-invariant track point with a simple structure as well as the box model. (2) The U-shape is close to the actual shape of a vehicle with curved bumpers but the expression and fitting process for the U-shape is more complex than L-shape feature extraction.

To mitigate the effect of appearance changes, the proposed algorithm tracks L-shaped features that consist of a corner point and two lines. To reduce unexpected errors from appearance changes, the proposed tracking algorithm tracks the corner point of a target instead of the center point, which has an ambiguous relative position in the box. By using the corner point, the relative position of the tracked point is invariant to shape change, which can lead to successful mitigation of appearance changes. The corner point has two kinds of motion, namely translational and rotational motion. Thus, a constant acceleration model is applied as a prediction model. For reliable tracking performance, the developed algorithm provides a corner point change scheme with uncertainty propagation in L-shape model switching. Furthermore, a size update rule to reduce position error from size changes is suggested.

The proposed vehicle tracking algorithm consists of three parts: L-shape feature extraction, an L-shape tracker, and L-shape to box model conversion (Fig. 1). First, L-shape feature extraction is a raw data abstraction process which extracts L-shape measurements from the point cloud. To extract the features, clustering and line feature extraction techniques are used. Second, the L-shape tracker estimates dynamic states of the track with dynamic and shape models by using the L-shape measurements. For implementing the L-shape tracker, the Kalman filter and nearest neighbor are adopted because the simple structure of the KF-NN allows it to focus on the model switching framework. This simple structure also has the potential to improve the proposed method with advanced tracking algorithms, such as MHT, PF, and RFS. At the same time, an L-shape model switching scheme is provided to detect and compensate for the dynamic states when the tracked corner point is changed. Lastly, the L-shape track is converted to a box track to represent the accurate position and velocity of the target. In this step, the rotational motion of the corner point is eliminated to describe the actual target motion.

This paper is organized as follows. Section II describes the L-shape feature extraction process. Section III explains the L-shape tracker, which is an L-shape model switching and tracking algorithm. Section IV presents L-shape to box model conversion. Section V is an analysis of the evaluation results for the proposed algorithm. Section VI provides a conclusion with a summary of evaluation results and future work.

II. L-SHAPE FEATURE EXTRACTION

The raw data generated from a laser scanner is a set of points representing the distance to obstacles. In order to convert these points to meaningful information, the points

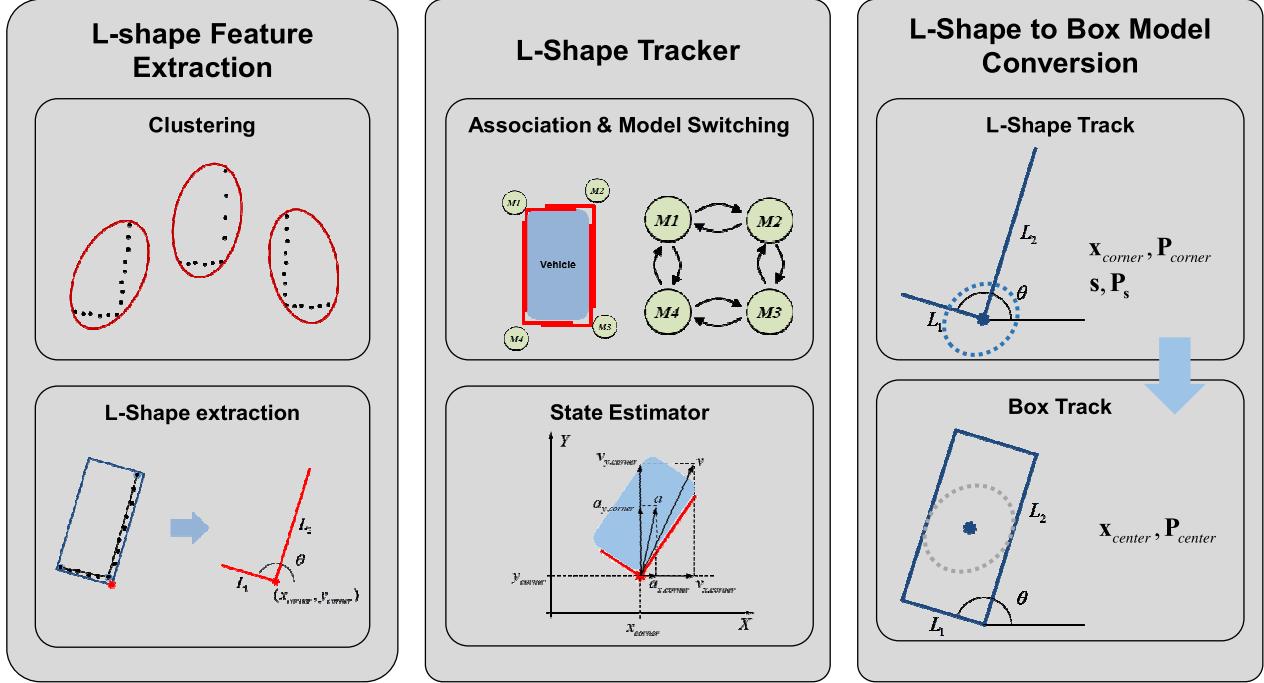


Fig. 1. The overall structure of the L-shape tracking algorithm that consists of L-shape feature extraction, an L-shape tracker, and L-shape to box model conversion.

are clustered and abstracted as an L-shape feature. However, clustered point clouds in driving environments have various shapes. The proposed method mainly targets the vehicle for detection and tracking. Since most vehicles are box shaped, the L-shape feature is efficient for representing a vehicle with a simple structure.

The critical problem of extracting L-shape features from non-L-shaped point clouds is orientation error. Orientation is difficult to extract from a circular object the object lacks an orientation. For vehicles, the point cloud is generally U-shaped because of the curved bumper. This curved bumper can be fitted with an orientation error, which is mostly within 10 degrees because the longest line that determines orientation is extracted from the flat plane of the bumper. This level of orientation error can be suppressed by the L-shape tracking algorithm.

In this study, an adaptive break-point detector is applied to cluster the point cloud and an iterative end-point fit method is adopted to extract lines from each cluster [12]. After that, an L-shape feature is determined based on the extracted lines.

A. Clustering

Clustering is a grouping process to express an object with associated points. In this stage, in order to apply the adaptive breakpoint detector, the points from laser scanners are represented in the polar coordinate system. Then, the points are sorted by the order of the angle. As shown in Fig. 2, if the distance between two consecutive points in angle order is within the threshold distance (D_{th}), then the two points are assigned to the same cluster. The adaptive breakpoint detector provides adaptive threshold distance by

$$D_{th} = \min(r_i, r_{i+1}) \frac{\sin \Delta\theta}{\sin(\lambda - \Delta\theta)} + \sigma_r \quad (1)$$

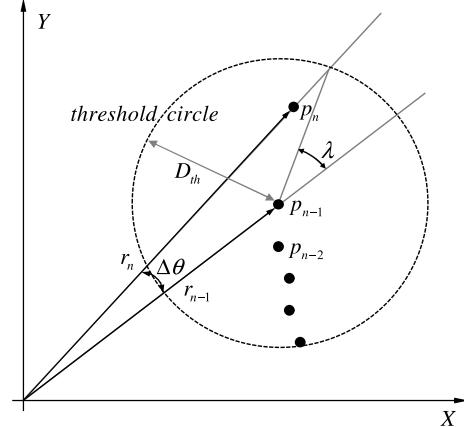


Fig. 2. Adaptive break-point detector compares consecutive two points with a threshold distance (depicted as threshold circle) to assign clusters, in a polar coordinate system.

where r_i and r_{i-1} are the distance information of consecutive points, $\Delta\theta$ is the angle difference between the two points, λ is an acceptable angle for determining the points to be of the same cluster, and σ_r is the standard deviation of the noise of the distance measure.

By using (1), different threshold distance is applied to cluster a dense point cloud in the near distance and the a sparse point cloud in the far distance.

B. Line Extraction

Fig. 3 shows the line extraction method that uses an iterative end-point fit. This method starts with two end-points. Among the clustered points, two points that have a minimum and maximum angle are selected as the initial end-points of the

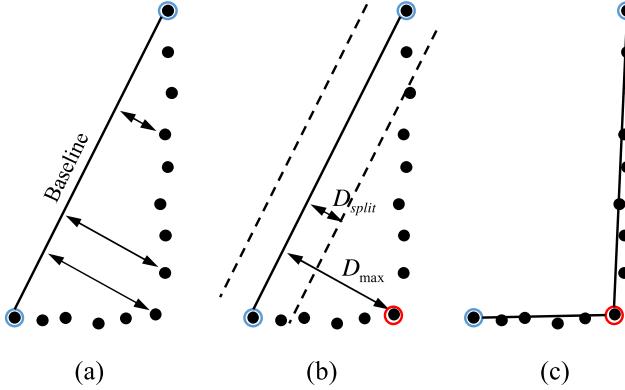


Fig. 3. Line split process of the iterative end-point fit method; (a) baseline from initial end points, (b) determination of break point, and (c) line split result.

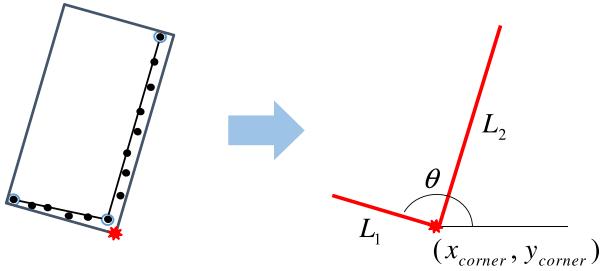


Fig. 4. Among the four corner points of the bounding box, one corner point that is the closest to the laser scanner on the ego-vehicle is selected as the L-shape feature corner point with connected lines.

cluster (Fig. 3 (a), points with blue circles). A baseline is set by the initial end-points, and the distances from the baseline to each point of the cluster are calculated (Fig. 3 (a)). After that, as shown in Fig. 3 (b), a point that has the longest distance (D_{max}) is determined as the break-point (the point with a red circle). By using the break-point, the baseline is divided into two lines, as in Fig. 3 (c). In the same way, each line is divided until the maximum distance is less than the threshold distance (D_{split}). Afterward, the merge process is applied to prevent the creation of too many lines due to line break errors. To reduce the number of line break errors, collinear lines adjacent to each other are represented as one line in this merge process.

C. L-Shape Extraction

By using the extracted lines, the cluster is fitted with an oriented bounding box. Among the extracted lines, the longest line is selected as the representative line to determine the orientation of the bounding box. After determining the orientation, the bounding box is fitted to have a minimum area [35], [36]. By using the bounding box, an L-shape feature is extracted with the closest corner point of the box and two lines connected to the corner point as shown in Fig. 4.

As a result, the L-shape feature contains the position of the corner point (x_{corner}, y_{corner}), the length of two lines (L_1, L_2), and the orientation (θ). L_1 and L_2 of the L-shape feature are assigned clockwise, in order of L_1 , right angle, and L_2 . The orientation θ is defined as the orientation of L_1 . The L-shape feature provides two types of measurements; one is

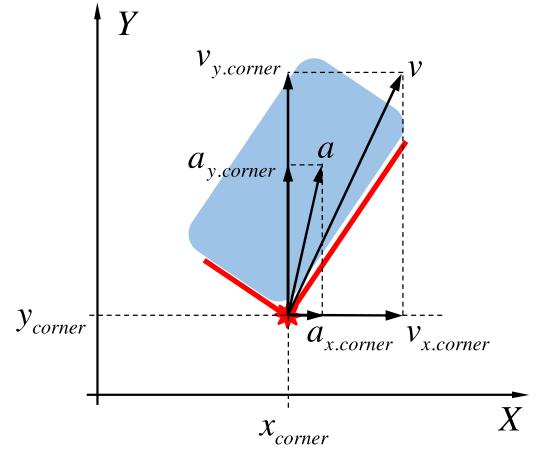


Fig. 5. The dynamic model describes motion information for the corner point.

the dynamic measurement (\mathbf{z}_D) for an update of the dynamic state and the other is the shape measurement (\mathbf{z}_S) for an update of the shape state in the L-shape tracker as

$$\begin{aligned}\mathbf{z}_D &= [x_{corner}, \quad y_{corner}]^T \\ \mathbf{z}_S &= [L_1, \quad L_2, \quad \theta]^T.\end{aligned}\quad (2)$$

III. L-SHAPE TRACKER

The L-shape tracker estimates the dynamic and shape states of the track using a Kalman filter [16] with a nearest neighbor data association. In order to estimate the dynamic state of the track precisely, the L-shape tracker adopts not only a dynamic model, but also a shape model. This estimation algorithm also includes an L-shape model switching scheme to compensate the corner point switching according to appearance change.

A. Track Models

For L-shape tracking, two types of track models are proposed; one is a dynamic model, and the other is a shape model. The dynamic model represents the motion of the tracks and the shape model estimates size, orientation, and yaw rate of the L-shape to compensate for appearance changes.

1) *Dynamic Model*: The dynamic model includes information about the position (x, y), velocity (v_x, v_y), and acceleration (a_x, a_y) of the corner point as shown in Fig. 5. The dynamic information introduced in Fig. 5 is represented as vector \mathbf{x}_D in

$$\mathbf{x}_D = [x \quad y \quad v_x \quad v_y \quad a_x \quad a_y]^T. \quad (3)$$

Since the corner point motion is a combination of translational and rotational motion, movement of the corner point can be described by the constant acceleration model in

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}_D \mathbf{x}_{k-1} + N(\mathbf{0}, \mathbf{Q}_D) \\ \text{where } \mathbf{F}_D &= \begin{bmatrix} 1 & 0 & T_s & 0 & \frac{T_s^2}{2} & 0 \\ 0 & 1 & 0 & T_s & 0 & \frac{T_s^2}{2} \\ 0 & 0 & 1 & 0 & T_s & 0 \\ 0 & 0 & 0 & 1 & 0 & T_s \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},\end{aligned}\quad (4)$$

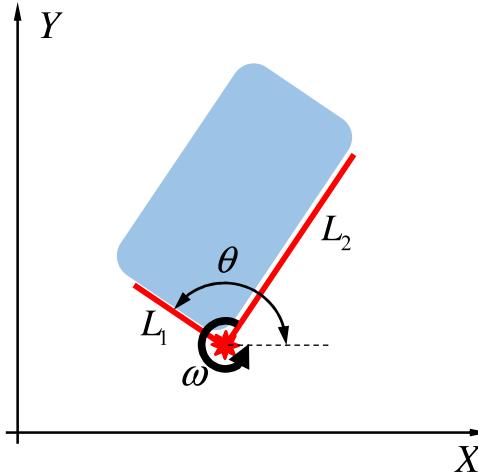


Fig. 6. Shape model presents the size, orientation of the L-shape, and yaw rate of the target.

where subscript k is current time step in the discrete time domain, \mathbf{F}_D is the process matrix representing constant acceleration, T_s is the sampling time, and $N(\mathbf{0}, \mathbf{Q}_D)$ is the Gaussian noise of the dynamic model with zero mean and covariance matrix \mathbf{Q}_D .

$$\mathbf{y}_{D,k} = \mathbf{H}_D \mathbf{x}_k + N(\mathbf{0}, \mathbf{R}_D) \quad (5)$$

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

Measurable data from the sensor is position information and the measurement model of the dynamic state is (5), where \mathbf{H}_D is the measurement matrix and $N(\mathbf{0}, \mathbf{R}_D)$ is the measurement noise represented by a Gaussian distribution with zero mean and covariance matrix \mathbf{R}_D .

2) *Shape Model:* The shape model is composed of line length (L_1, L_2), the orientation of L_1 (θ), and yaw rate (ω) as depicted in Fig. 6. The shape information described in Fig. 6 is reflected in vector \mathbf{s} as

$$\mathbf{s} = [L_1 \ L_2 \ \theta \ \omega]^T. \quad (6)$$

For estimation of shape information, a static model is applied to line length because the target size does not change over time. Since the yaw rate also does not change greatly in a short time, a constant turn rate model is applied to the orientation and yaw rate.

$$\mathbf{s}_k = \mathbf{F}_S \mathbf{s}_{k-1} + N(\mathbf{0}, \mathbf{Q}_S) \quad (7)$$

where $\mathbf{F}_S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

where \mathbf{F}_S is the process matrix containing a static model for line length and a constant turn rate model for orientation and yaw rate, T_s is sampling time, and $N(\mathbf{0}, \mathbf{Q}_S)$ is the process noise that has a Gaussian distribution with zero mean and covariance \mathbf{Q}_S .

Among the states of the shape model, the line length and orientation are measurable data from the sensor. Thus, the

measurement model of the shape model is the following.

$$\mathbf{y}_{S,k} = \mathbf{H}_S \mathbf{s}_k + N(\mathbf{0}, \mathbf{R}_S) \quad (8)$$

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

where \mathbf{H}_S is the measurement matrix for the shape model and $N(\mathbf{0}, \mathbf{R}_S)$ is the measurement noise, which has a Gaussian distribution with zero mean and covariance \mathbf{R}_S .

B. Prediction

According to the track models, the estimated track states are predicted with the sample time. At the same time, the uncertainty of the track is propagated by the prediction equation of the Kalman filter.

1) *Dynamic State Prediction:* The dynamic states of the corner point are predicted according to the dynamic model. From the posterior state of the previous time step ($\mathbf{x}_{k-1|k-1}$), the a priori state of the current time step ($\mathbf{x}_{k|k-1}$) is calculated by

$$\mathbf{x}_{k|k-1} = \mathbf{F}_D \mathbf{x}_{k-1|k-1}. \quad (9)$$

The uncertainty of the previous step ($\mathbf{P}_{D,k-1|k-1}$) is also propagated to the current prior uncertainty ($\mathbf{P}_{D,k|k-1}$) by

$$\mathbf{P}_{D,k|k-1} = \mathbf{F}_D \mathbf{P}_{D,k-1|k-1} \mathbf{F}_D^T + \mathbf{Q}_D. \quad (10)$$

2) *Shape Prediction:* The shape information of the track is also predicted by the shape model. The prior state ($\mathbf{s}_{k|k-1}$) of the shape information is predicted from the previous posterior state ($\mathbf{s}_{k-1|k-1}$) by using the shape model. Similarly to dynamic the uncertainty propagation, the prior uncertainty ($\mathbf{P}_{S,k|k-1}$) can be predicted from the previous posterior uncertainty ($\mathbf{P}_{S,k-1|k-1}$).

$$\mathbf{s}_{k|k-1} = \mathbf{F}_S \mathbf{s}_{k-1|k-1} \quad (11)$$

$$\mathbf{P}_{S,k|k-1} = \mathbf{F}_S \mathbf{P}_{S,k-1|k-1} \mathbf{F}_S^T + \mathbf{Q}_S \quad (12)$$

(11) and (12) represents the prediction equations for the shape information.

C. Model Switching & Data Association

The L-shape model is based on the corner point, which is closest to the laser scanner on the ego-vehicle, and it can be changed due to the pose of the target vehicle. The proposed tracking algorithm provides a scheme to compensate for this change in corner point.

Fig. 7 shows an example of corner point change scenarios. The corner points can be changed to an adjacent corner point, clockwise and counter clockwise. A new track initialized with the first detected corner point starts to track the target vehicle with the model number one (M1). After that, the model number will increase or decrease with corner point switching according to the target pose and viewpoint of the ego-vehicle.

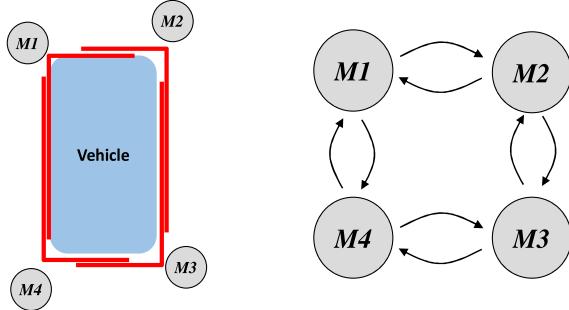


Fig. 7. Each corner point of the track has a model number from M1 to M4. When the detected corner point is changed, the model number is also changed.

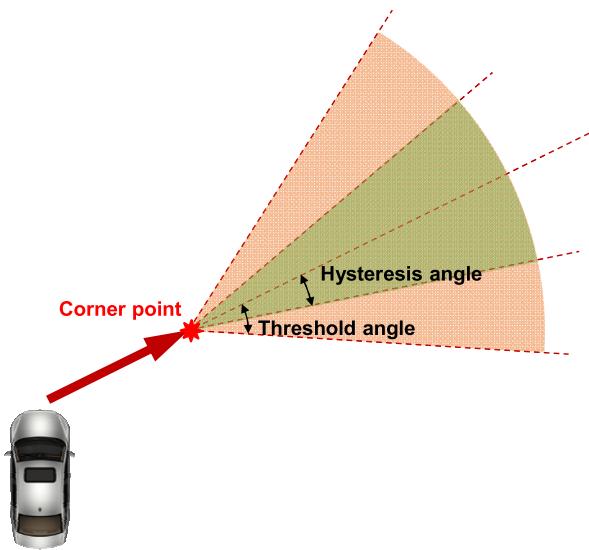


Fig. 8. L-shape model transition detection is performed by comparing the view angle and orientation of the L-shape model.

1) Model Transition Detection: In the model transition detection stage, the L-shape track is evaluated to detect the change in corner point. The evaluation is performed by using the view angle and orientations of L_1 and L_2 . As shown in Fig. 8, if the orientation is within the threshold angle, the model transition is detected.

However, if the heading angle of the target is close to the view angle, the model transitions are frequently detected. In order to prevent frequent transitions, a hysteresis angle is applied, as depicted in Fig. 8. Within the hysteresis angle, the track does not change the model; instead, the measurement is converted to match the track.

a) Clockwise measurement conversion: The clockwise measurement conversion switches the corner point along the line L_1 . The converted corner point position ($x_{corner,cw}$, $y_{corner,cw}$) is calculated by

$$\begin{aligned} x_{corner,cw} &= x_{corner} + L_1 \cos \theta \\ y_{corner,cw} &= y_{corner} + L_1 \sin \theta. \end{aligned} \quad (13)$$

After that, the line length ($L_{1,cw}$, $L_{2,cw}$) is switched, and the orientation is rotated according to the switched $L_{1,cw}$

by

$$\begin{aligned} L_{1,cw} &= L_2 \\ L_{2,cw} &= L_1 \\ \theta_{cw} &= \theta - \pi/2. \end{aligned} \quad (14)$$

b) Counter clockwise measurement conversion: In the same way, the counter-clockwise measurement conversion switches the corner point along the line L_2 . The position of the converted corner point, and the line lengths and orientation are derived by following equations.

$$\begin{aligned} x_{corner,ccw} &= x_{corner} + L_2 \sin \theta \\ y_{corner,ccw} &= y_{corner} - L_2 \cos \theta \end{aligned} \quad (15)$$

$$\begin{aligned} L_{1,ccw} &= L_2 \\ L_{2,ccw} &= L_1 \\ \theta_{ccw} &= \theta + \pi/2 \end{aligned} \quad (16)$$

2) Corner Point Switching: After the model transition detection, the corner point should be switched to another corner point when a model transition is detected. The calculation of the switched dynamic states is based on the shape model. There are two types of corner point switching, clockwise switching and counter clockwise switching.

As depicted in Fig. 7, there are four corner points, and a model number is assigned for each point from M1 to M4. When the track is initialized, the track model number is assigned as M1. If the corner point is switched clockwise, then the model number increases. Otherwise (counter clockwise switching), the model number decreases.

a) Clockwise switching: The position of the switched corner point is moved along the L_1 .

$$\begin{aligned} x_{corner}^{Mj} &= x_{corner}^{Mi} + L_1^{Mi} \cos \theta^{Mi} \\ y_{corner}^{Mj} &= y_{corner}^{Mi} + L_1^{Mi} \sin \theta^{Mi}, \\ \text{where } &\begin{cases} j = i + 1, & i < 4 \\ j = 1, & i = 4 \end{cases} \end{aligned} \quad (17)$$

Equation (17) represents the switched position of the corner point from the previous corner point position. The superscript Mi and Mj are the model numbers before switching and after switching, respectively.

The velocity and acceleration of the switched corner point can be derived by compensating for rotational motion. The rotational motion of the track is assumed to be a constant turn rate model with current yaw rate. Based on this assumption, the velocity and acceleration of the switched corner point can be obtained by followings.

$$\begin{aligned} v_{x,corner}^{Mj} &= v_{x,corner}^{Mi} - L_1^{Mi} \omega \sin \theta^{Mi} \\ v_{y,corner}^{Mj} &= v_{y,corner}^{Mi} + L_1^{Mi} \omega \cos \theta^{Mi}, \\ \text{where } &\begin{cases} j = i + 1, & i < 4 \\ j = 1, & i = 4 \end{cases} \end{aligned} \quad (18)$$

$$\begin{aligned} a_{x,corner}^{Mj} &= a_{x,corner}^{Mi} - L_1^{Mi} \omega^2 \cos \theta^{Mi} \\ a_{y,corner}^{Mj} &= a_{y,corner}^{Mi} - L_1^{Mi} \omega^2 \sin \theta^{Mi}, \\ \text{where } &\begin{cases} j = i + 1, & i < 4 \\ j = 1, & i = 4 \end{cases} \end{aligned} \quad (19)$$

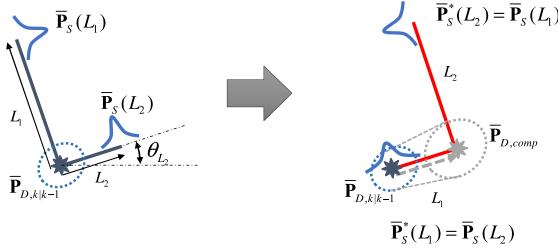


Fig. 9. Uncertainty propagation when the L-shape model switches.

b) *Counter clockwise switching*: in the same way as the clockwise switching, the dynamic information of a corner point that is switched by counter-clockwise switching can be calculated by

$$\begin{aligned} x_{\text{corner}}^{Mj} &= x_{\text{corner}}^{Mi} + L_2^{Mi} \sin \theta^{Mi} \\ y_{\text{corner}}^{Mj} &= y_{\text{corner}}^{Mi} - L_2^{Mi} \cos \theta^{Mi}, \\ \text{where } &\begin{cases} j = i - 1, & i > 1 \\ j = 4, & i = 1, \end{cases} \end{aligned} \quad (20)$$

$$\begin{aligned} v_{x,\text{corner}}^{Mj} &= v_{x,\text{corner}}^{Mi} + L_2^{Mi} \omega \cos \theta^{Mi} \\ v_{y,\text{corner}}^{Mj} &= v_{y,\text{corner}}^{Mi} + L_2^{Mi} \omega \sin \theta^{Mi}, \\ \text{where } &\begin{cases} j = i - 1, & i > 1 \\ j = 4, & i = 1, \end{cases} \end{aligned} \quad (21)$$

$$\begin{aligned} a_{x,\text{corner}}^{Mj} &= a_{x,\text{corner}}^{Mi} - L_2^{Mi} \omega^2 \sin \theta^{Mi} \\ a_{y,\text{corner}}^{Mj} &= a_{y,\text{corner}}^{Mi} + L_2^{Mi} \omega^2 \cos \theta^{Mi} \\ \text{where } &\begin{cases} j = i - 1, & i > 1 \\ j = 4, & i = 1. \end{cases} \end{aligned} \quad (22)$$

3) *Uncertainty Propagation*: The switched corner point is at the line end of the previous shape model. Since the line length and orientation are the estimated values with uncertainty, the uncertainty of the shape affects the dynamic information of the switched corner point. As a result, the uncertainty of the switched corner point should reflect the uncertainty of the shape, as shown in Fig. 9. The propagated covariance matrix is calculated by the linear transformation using the Jacobian matrix of the corner point switching equation.

Each dynamic state of the switched corner point can be represented by

$$\mathbf{x}_{k|k-1}^* = \mathbf{f}(\mathbf{x}_{k|k-1}, \mathbf{s}_{k|k-1}). \quad (23)$$

By using (23), we can obtain a Jacobian matrix, and uncertainty propagation is performed using linear transformation as in the following equation.

$$\begin{aligned} \mathbf{P}_{D,k|k-1}^* &= \mathbf{J} \mathbf{P}_{k|k-1} \mathbf{J}^T, \\ \text{where } \mathbf{J} &= \left[\frac{d\mathbf{f}}{d\mathbf{x}_{k|k-1}}, \frac{d\mathbf{f}}{d\mathbf{s}_{k|k-1}} \right], \quad \mathbf{P}_{k|k-1} = \begin{bmatrix} \mathbf{P}_{D,k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{S,k|k-1} \end{bmatrix} \end{aligned} \quad (24)$$

4) *Nearest Neighbor Association*: In order to solve the multi-target tracking problem, data association is an essential

part of matching between tracks and measurement in a Kalman filter based algorithm. In this study, the nearest neighbor association is adopted as the data association method because a laser scanner can provide a precise point cloud and rarely generates clutter. The nearest neighbor association matches the measurement to track by using the Mahalanobis distance. The Mahalanobis distance (D_M) is determined by the residual (\mathbf{v}_D) and residual covariance (\mathbf{S}_D) as in

$$D_M = \mathbf{v}_D^T \mathbf{S}_D^{-1} \mathbf{v}_D$$

where

$$\mathbf{v}_D = \mathbf{z}_D - \mathbf{H}_D \mathbf{x}_{k|k-1}, \quad \mathbf{S}_D = \mathbf{H}_D \mathbf{P}_{D,k|k-1}^* \mathbf{H}_D^T + \mathbf{R}_D. \quad (25)$$

For each measurement, the Mahalanobis distance is calculated, and the track that has the shortest distance is matched with the measurement.

D. Update

1) *Dynamic State Update*: The dynamic state and uncertainty are updated with associated measurements by using a Kalman filter update equation as follows.

$$\begin{aligned} \mathbf{K}_D &= \mathbf{P}_{D,k|k-1}^* \mathbf{H}_D^T \mathbf{S}_D^{-1} \\ \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_D (\mathbf{z}_D - \mathbf{H}_D \mathbf{x}_{k|k-1}) \\ \mathbf{P}_{D,k|k} &= (\mathbf{I} - \mathbf{K}_D \mathbf{H}_D) \mathbf{P}_{D,k|k-1}^* \end{aligned} \quad (26)$$

where the \mathbf{K}_D is the Kalman gain for the dynamic state update.

2) *Shape Update*: For a shape state update, an additional update rule is applied to the Kalman filter update process. The line length information of the shape state is directly related to the appearance detected by the sensor. Since the sensor always detects a smaller size than the actual size of the real object, size growth should be updated with a larger weighting factor than size shrink. The weighting factors for size update are determined by the ratio of the previous size to the current measured size as shown in

$$\begin{aligned} \mathbf{R}_S^*(L_1) &= w_{L_1} \mathbf{R}_S(L_1), \\ \text{where } w_{L_1} &= L_{1,\text{previous}} / L_{1,\text{measured}} \\ \mathbf{R}_S^*(L_2) &= w_{L_2} \mathbf{R}_S(L_2), \\ \text{where } w_{L_2} &= L_{2,\text{previous}} / L_{2,\text{measured}}. \end{aligned} \quad (27)$$

As a result, the measurement noise covariance (\mathbf{R}_S) is affected by the weighting factor because size growth means that the measured value is approaching the true value.

The compensated measurement noise covariance affects the residual covariance (\mathbf{S}_S) for shape information as shown in

$$\begin{aligned} \mathbf{v}_S &= \mathbf{z}_S - \mathbf{H}_S \mathbf{s}_{k|k-1} \\ \mathbf{S}_S &= \mathbf{H}_S \mathbf{P}_{S,k|k-1} \mathbf{H}_S^T + \mathbf{R}_S^*. \end{aligned} \quad (28)$$

This effect is propagated to the Kalman gain of the shape update equation (\mathbf{K}_S) through the residual covariance (\mathbf{S}_S).

$$\begin{aligned} \mathbf{K}_S &= \mathbf{P}_{S,k|k-1} \mathbf{H}_S^T \mathbf{S}_S^{-1} \\ \mathbf{s}_{k|k} &= \mathbf{s}_{k|k-1} + \mathbf{K}_S (\mathbf{z}_S - \mathbf{H}_S \mathbf{s}_{k|k-1}) \\ \mathbf{P}_{S,k|k} &= (\mathbf{I} - \mathbf{K}_S \mathbf{H}_S) \mathbf{P}_{S,k|k-1} \end{aligned} \quad (29)$$

IV. L-SHAPE TO BOX MODEL CONVERSION

The dynamic state of the corner point includes the rotational motion of the target. The L-shape to box model conversion stage is the process to calculate the dynamic state of the target. The target dynamic state can be calculated from the dynamic state of the corner point by eliminating rotational motion.

A. Dynamic State Conversion

In this stage, the dynamic state is changed from corner point motion to target motion. The position can be converted by using the geometric information of the shape model. The velocity of the corner point is the sum of the target velocity and the tangential velocity of the rotational motion at the corner point. The acceleration of the corner point is also the sum of the target acceleration and rotational acceleration to the geometric center of the target.

1) Position Conversion: The position of the box center can be easily calculated by using geometric information. From the line length and orientation, the position of the center point is obtained by using

$$\begin{aligned} x &= x_{\text{corner}} + \varepsilon_x, \quad \text{where } \varepsilon_x = (L_1 \cos \theta + L_2 \sin \theta)/2 \\ y &= y_{\text{corner}} + \varepsilon_y, \quad \text{where } \varepsilon_y = (L_1 \sin \theta - L_2 \cos \theta)/2. \end{aligned} \quad (30)$$

2) Velocity Conversion: The velocity of the corner point includes both translational velocity and rotational velocity. The rotational motion of the corner point is assumed as a uniform circular motion because we already assumed a constant turn rate model for the shape model. Therefore, we can derive an equation for tangential velocity by using the distance from the center as the radius of the circular motion.

$$\begin{aligned} v_{x,\text{center}} &= v_{x,\text{corner}} - r\omega \cos \alpha \\ v_{y,\text{center}} &= v_{y,\text{corner}} - r\omega \sin \alpha, \\ \text{where } \alpha &= \tan^{-1}(\varepsilon_y/\varepsilon_x) - \pi/2 \end{aligned} \quad (31)$$

3) Acceleration Conversion: In a way similar to the velocity conversion, acceleration information can be obtained by eliminating the rotational acceleration. The rotational acceleration can also be derived from the uniform circular motion of the corner point.

$$\begin{aligned} a_{x,\text{center}} &= a_{x,\text{corner}} - r\omega^2 \cos \beta \\ a_{y,\text{center}} &= a_{y,\text{corner}} - r\omega^2 \sin \beta, \quad \text{where } \beta = \tan^{-1}(\varepsilon_y/\varepsilon_x) \end{aligned} \quad (32)$$

B. Uncertainty Conversion

The uncertainty conversion process is the same as the uncertainty propagation in the model switching stage. In order to find a linear transform relationship for conversion, a Jacobian matrix is calculated by using the dynamic state conversion relationship. As a result, the converted covariance is calculated.

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{s}_{k|k}), \quad (33)$$

$$\mathbf{P}_k = \mathbf{J} \mathbf{P}_{k|k} \mathbf{J}^T,$$

$$\text{where } \mathbf{J} = \left[\frac{d\mathbf{f}}{d\mathbf{x}_{k|k}}, \frac{d\mathbf{f}}{d\mathbf{s}_{k|k}} \right], \quad \mathbf{P}_{k|k} = \left[\begin{array}{c} \mathbf{P}_{D,k|\mathbf{k}} \mathbf{0} \\ \mathbf{0} \mathbf{P}_{S,k|\mathbf{k}} \end{array} \right]. \quad (34)$$

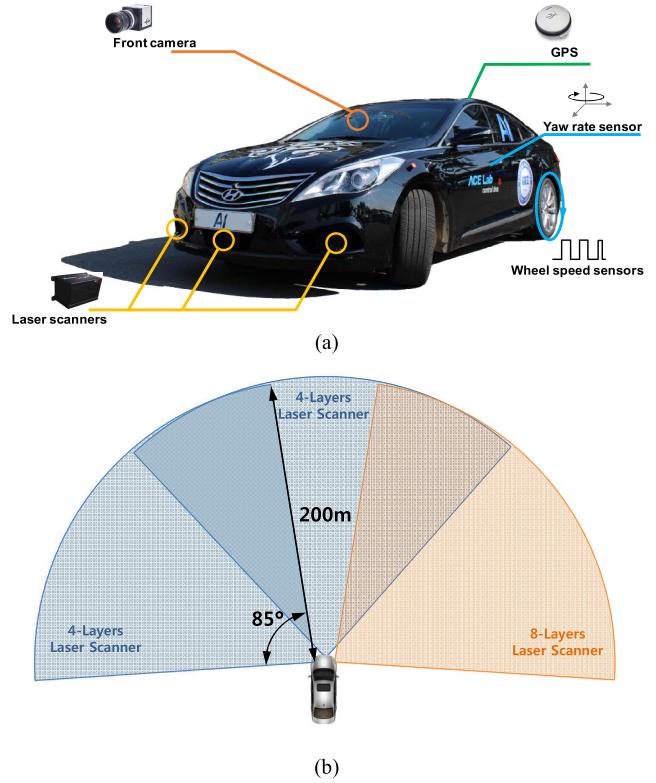


Fig. 10. Sensor configuration of the test vehicle and FOV of the installed laser scanners. (a) the proposed algorithm uses laser scanners, yaw rate sensors, and wheel speed sensors. The GPS and front camera are used for evaluation of the algorithm. (b) among the laser scanners, 4-layer laser scanners are installed at the center and left corner of the front bumper, and an 8-layer laser scanner is equipped on the right corner of the front bumper.

V. EXPERIMENTS

A. Experimental Environments

The proposed tracking system was evaluated with a test vehicle shown in 10 (a). On the front bumper, three laser scanners were installed. Each laser scanner can detect obstacles up to 200 m within 85°. The angular resolution of horizontal scanning is 0.125°, the vertical angular resolution is 0.8°, and the scan frequency is 12.5 Hz. The installed laser scanners can scan multiple layers. Two laser scanners installed on the center and the left of the bumper are 4-layer laser scanners. On the right of the bumper, an 8-layer laser scanner is installed to detect road boundary like curbs. The field of view (FOV) of the sensors is depicted in Fig. 10 (b).

In this study, the L-shape feature is 2D information. However, the installed laser scanners provide 3D position information of each point in a Cartesian coordinate system. In order to apply the proposed algorithm, the 3D information is projected onto an x-y plane (ground plane). After that, the projected points are converted to the polar coordinate system to apply the adaptive break-point detector. The proposed algorithm operates every 80 ms, which is the same as the sampling time for the laser scanner. The tracking algorithm provides multi-target tracking functionality and manages up to 150 tracks. This method was tested on a machine with an i5-4690 CPU with 8 GB RAM. The algorithm was implemented using MS Visual Studio 2013 on Windows OS. During 97.5% of

the operation time, the execution time per time step was within 5 ms.

In order to evaluate the performance of the proposed method, we need the ground truth states of the target vehicle as reference data. Unfortunately, the true states of the target cannot be known. Thus, we set up a system that can measure the true states precisely.

Both the ego-vehicle and target vehicle are equipped with high precision GPS (RTK-GPS). With the GPS equipment, ego-vehicle and target vehicle position and velocity are measured. The synchronization of the measured information is performed by using the global time stamp from each GPS. The measured data is corrected more precisely with post-processing, which is the application of a smoothing algorithm using a vehicle model. Since the error bound of the RTK-GPS is within 10 cm and the position information is corrected by the smoothing algorithm, the post-processed GPS information can be assumed as ground truth. With this ground truth, the proposed algorithm is compared to the box tracking algorithm embedded in the Ibeo Lux laser scanner. Ibeo Lux provides object information with a point cloud (raw data) every 80 ms. The object information includes the position, velocity, heading angle, size (length and width), and class of the object. Additionally, the sensor provides track uncertainty with the standard deviation for the position and velocity, management information consisting of ID and age, and contour information with associated raw data points. Using this sensor, the proposed algorithm can be compared to commercially available automotive applications of laser scanners that can provide object information.

In the experiment, two types of information should be checked. One is about how the algorithm mitigates appearance changes and the other is about estimation performance. In order to check the mitigation of the appearance change problem, the model number, which can show the corner point switching according to appearance change, and L_1 and L_2 , which estimate track size, are monitored. For evaluation of the estimation performance, errors in position, speed, and heading are monitored. This information is directly related to inferred target vehicle behavior, specifically lane keeping and lane change. This inference necessitates a precise position and time to cross the lane. When the lane width is 3 m to 3.6 m and the average passenger car width is 2 m, the lateral position error should be under 0.5 m to determine whether the target vehicle is within the lane or not. The time to lane crossing, which is useful information for determining the lane change motion of the target vehicle, can be calculated using the vehicle speed and heading angle. The estimation performance is compared to that of the box tracking algorithm embedded in a commercial laser scanner.

There are two types of experimental scenarios; one is a circular turn scenario, the other is a driving scenarios on various curvature roads. The circular turn scenario provides an explicit appearance change problem. Thus, we can analyze the performance of mitigating the appearance change problem through the proposed method. The driving scenarios on various curvature roads are for the evaluation of estimation performance in more realistic situations. In these scenarios, the target

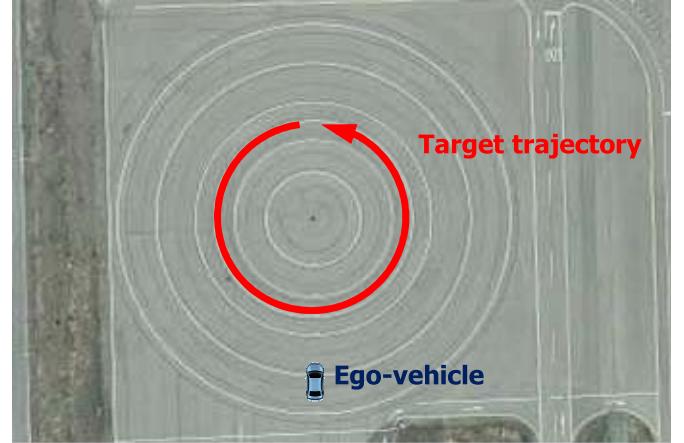


Fig. 11. The circular turn scenario is tested on the circular turn test circuit of KATRI.

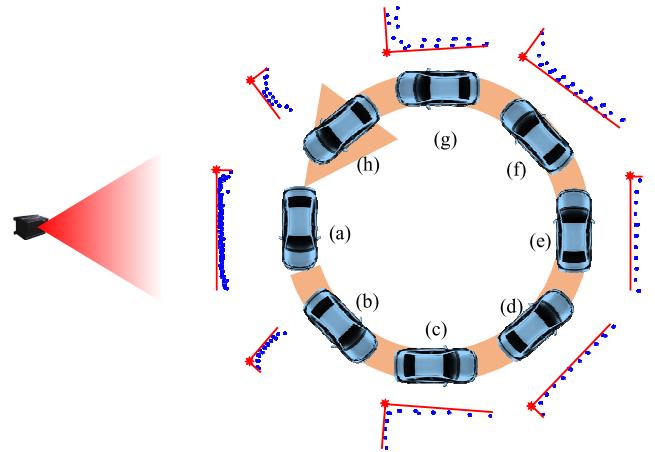


Fig. 12. Appearance changes of the target vehicle and L-shape extraction in the circular turn scenario.

vehicle is driven to make various motions with appearance change problems in a road environment.

B. Circular Turn Scenario

The test circuit for testing the circular turn scenario is the circular turn test track of the Korea Automobile Testing and Research Institute (KATRI). This circuit provides circle lanes of multiple radii to support a circular turn test as shown in Fig. 11. By using these lanes, the target vehicle can be driven on a circular path successfully. In this scenario, we can explicitly evaluate the effect of the appearance changes of the target vehicle on the estimation performance.

Fig. 12 shows different distributions of the point cloud at each target vehicle position during the circular turn scenario. At the position of Fig. 12 (a) and (e), only the side of the target vehicle is detected. Fig. 12 (b) and (h) show that the target vehicle is detected only the bumper. Fig. 12 (c) and (g) depict the detection of the entire L-shape of the target, which is the best case for the tracking algorithm. Fig. 12 (d) and (f) also show detection of an L-shape but not the entire target. At each position, the point cloud shows a different shape representing the relative position between the sensor and the

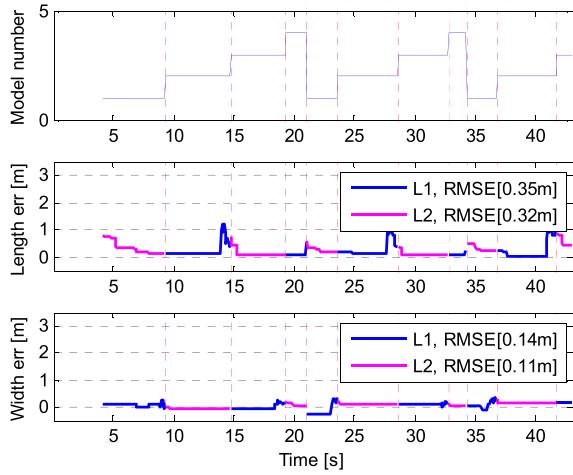


Fig. 13. Model number and estimation result of L_1 and L_2 length in the circular turn scenario.

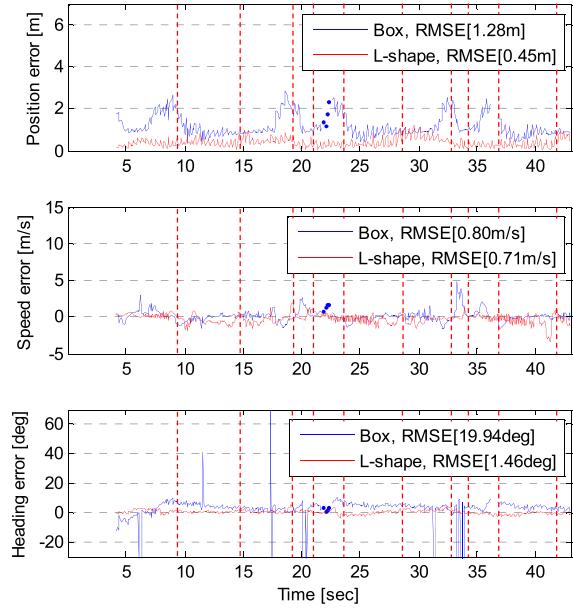


Fig. 14. Estimation errors of position, speed, and heading for the L-shape and box tracking algorithms in the circular turn scenario.

target vehicle. According to this appearance change, different L-shape features are extracted and the L-shape tracking algorithm will adapt these differences using the model switching scheme. By this scenario, the L-shape tracking algorithm can be evaluated for how it mitigates the appearance change effect.

Fig. 13 shows the model number switching and size estimation results for the circular turn scenario. As the corner point is switched, the model number changes from one to four per one circular turn of the target vehicle. In Fig. 13, the red vertical dashed lines represent the time when the model number changed. At that time, the length information of L_1 and L_2 is switched to the other. The horizontal dashed lines of magenta and red are the target vehicle length and width, respectively.

Fig. 14 depicts the estimation errors for the position, speed, and heading from the proposed algorithm and the box tracking

algorithm. The red vertical dashed lines represent the time of model switching as in Fig. 13. At this time, the appearance of the target vehicle is dramatically changed.

The position and speed errors in the box tracking system are significantly affected by appearance changes. The position error increased when the model number changed from one to two and three to four. At these moments, the target appearance change reflects shrinking of the target length information, as in Fig. 12 (b) and (h). In these cases, the position error is increased by up to 2.2 m and returned to the 1 m level by restoring length information. In the other cases (model changes from two to three and four to one), the target width information is suppressed as shown in Fig. 12 (a) and (e), but it causes relatively small errors compared to the previous cases. Each significant increase in the position error is reflected in the speed error. Although it is a small change in the speed error, speed error fluctuation is seen with large position errors. Heading estimation of the box tracking system also provides large errors with high peak error, which occurs when a small appearance changes to an entire L-shape. However, the estimation performance of the proposed algorithm is not affected by appearance changes. The errors of the proposed algorithm do not change with model switching as described in Fig. 14.

The performance of mitigating appearance changes is clearly reflected in the RMS error values for the position, speed, and heading. The RMS error values of the proposed algorithm are 0.45 m for position, 0.71 m/s for speed, and 1.46° for heading. The RMS error values of the box tracking algorithm are 1.28 m, 0.8 m/s, and 19.94° , for position, speed, and heading, respectively. As a result, the proposed method reduced position error by 0.83 m (64.8%), speed error by 0.09 m/s (12.7%), and heading error by 18.48° (92.7%) in RMS values. The proposed algorithm successfully reduces the errors due to the appearance change problem and the error analysis clearly reflects mitigation of the appearance changes by the proposed method.

C. Driving Scenarios on Various Curvature Roads

The test circuit for road driving scenarios is the steering performance test track of the KATRI. This circuit consists of various curvature roads to simulate various real roads, as shown in Fig. 15. The test scenarios on the various curvature roads are composed of four scenarios. The target vehicle was driven along the path (red line) with different dynamic motions, as shown in Fig. 15. Fig. 16 depicts the speed and steering of the target vehicle in each scenario. In order to simulate the real dynamic motions of moving vehicles, the target vehicle motions are combinations of two lateral behaviors and two longitudinal behaviors: lane keeping with constant speed, lane change with constant speed, lane keeping with acceleration & deceleration, and lane change with acceleration & deceleration.

1) *Lane Keeping With Constant Speed*: In this scenario, the target vehicle was driven at a speed of 40 kph with lane keeping. Fig. 17 describes the model number and size estimation for this scenario. The model number is changed



Fig. 15. The experiments were performed on a test circuit with various curvatures to evaluate the performance of the L-shape tracking algorithm.

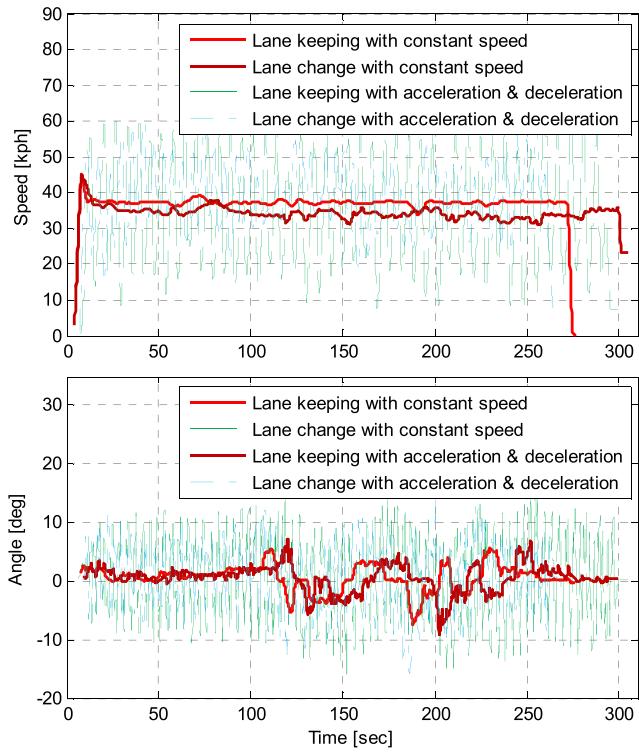


Fig. 16. Target vehicle speed and steering angle in the driving scenarios on various curvature roads.

between one and two because only the two corner points on the rear bumper are tracked in the scenario. Model switching does not occur frequently because corner point switching is rarely performed with lane keeping motions. The estimated line length information is saturated with target length and width.

Fig. 18 shows the estimation errors of the proposed algorithm and the box tracking algorithm. The position error of the box tracking algorithm keeps to about 2 m, but the proposed algorithm estimates the position to within 1 m most of the time in this scenario. Since the rear bumper of the target vehicle is mainly detected in this scenario, the box tracking algorithm, which has no ability to compensate for the length of the target,

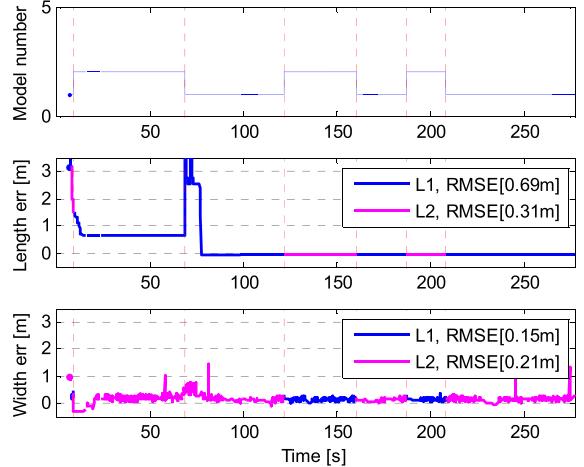


Fig. 17. Model number and estimation results for L_1 and L_2 length in the lane keeping with constant speed scenario.

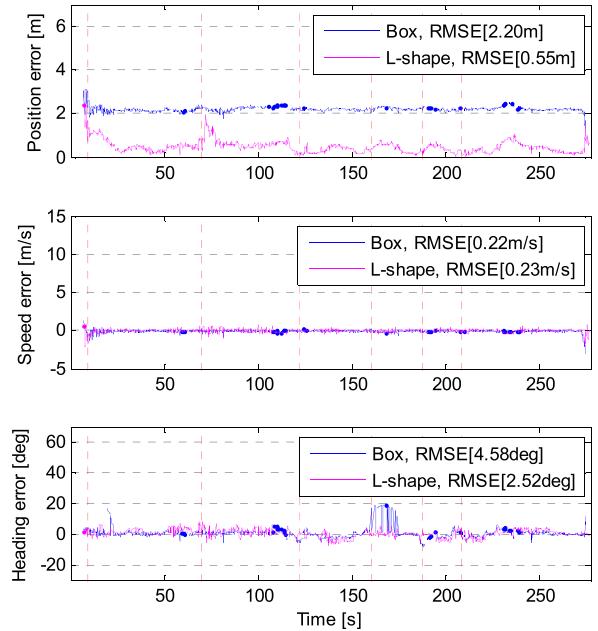


Fig. 18. Estimation errors for the position, speed, and heading of the L-shape and box tracking algorithms in the lane keeping with constant speed scenario.

shows a large error. However, the proposed algorithm compensates for the length information, which can be detected on a curved road. The speed errors of the two algorithms are similar because the target vehicle motion is stable with constant speed. The heading errors for box tracking show peak error when the model is changed, but the proposed algorithm keeps error levels to within 10° . The RMS position error for box tracking is 2.19 m, speed error is 0.24 m/s, and heading error is 4.58° . On the other hand, the RMS position error for the proposed method is 0.55 m, speed error is 0.23 m/s, and heading error is 2.52° . The proposed method reduced position, velocity, and heading errors by 1.65 m (75%), -0.01 m/s (-4.5%), and 2.06° (45%), respectively. The position error and heading error are greatly reduced, but the speed error is increased. However, the difference is only 1 cm/s, and this means that the speed estimation performance in this scenario is almost the same.

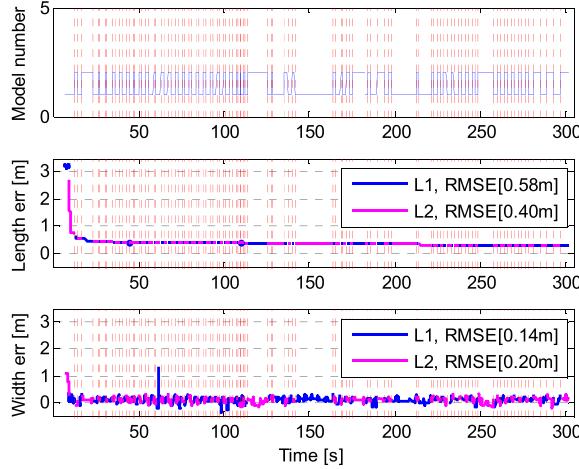


Fig. 19. Model number and estimation results for L_1 and L_2 length in the lane change with constant speed scenario.

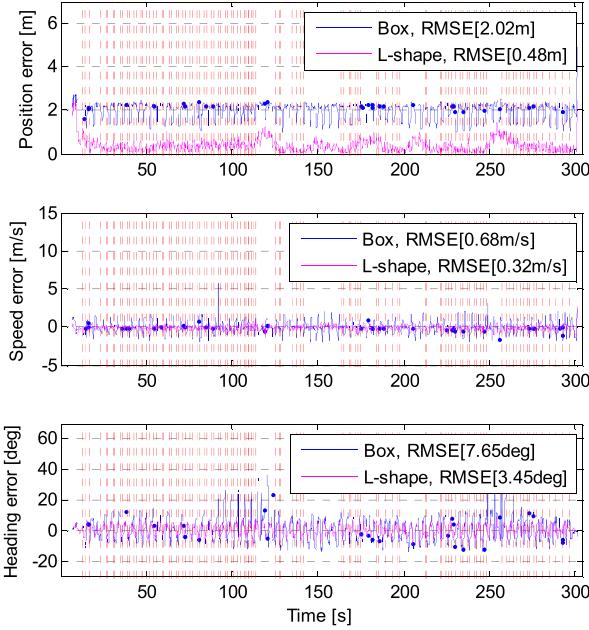


Fig. 20. Estimation errors for the position, speed, and heading of the L-shape and box tracking algorithms in the lane change with constant speed scenario.

2) Lane Change With Constant Speed: In the lane change with constant speed scenario, the target vehicle was driven with lane changes and constant speed (40kph). In this case, the L-shape model was frequently switched because the lane change maneuver causes frequent corner point changes between the two corners of the target rear bumper.

Fig. 19 represents the frequent model changes and switching of line length information in this scenario. The line length information estimated the target width successfully, but was short for the target length. The short target length estimation is caused by frequent self-occlusion of the target side appearance. Despite this frequent model switching, the proposed algorithm keeps the target length information close to the true value.

Fig. 20 shows the estimation performance for the two tracking systems in the lane change with constant speed scenario. The position error of the box tracking algorithm

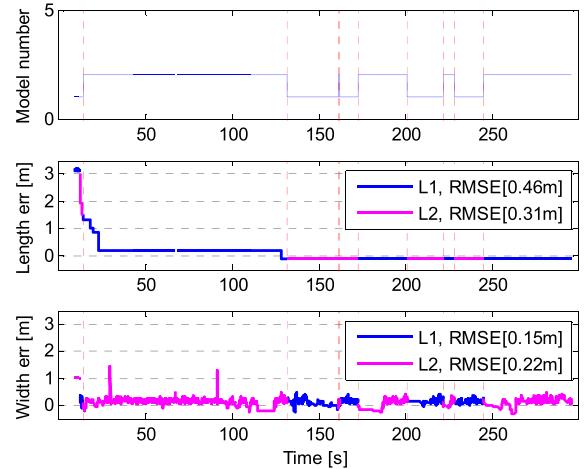


Fig. 21. Model number and estimation results for L_1 and L_2 length in the lane keeping with acceleration & deceleration scenario.

oscillates due to the frequent self-occlusion of the target side appearance by the rear bumper, and this error oscillation is reflected in the speed error. These errors are unexpected motion estimation errors of the box tracking method caused by appearance changes. Since this scenario can describe the lane change of a front vehicle on real roads, the box tracking algorithm might cause a wrong decision by an autonomous car in this type of scenario. The heading angle error of the box tracking method also shows large fluctuations due to lane change motions. In contrast, the proposed algorithm estimates position, speed, and heading with much less error.

The RMS position error for the box tracking algorithm is 2.02 m, the speed error is 0.68 m/s, and the heading error is 7.65°. The proposed algorithm reduced the position error to 0.48 m, speed error to 0.32 m/s, and heading error to 3.45°. The error reduction for position is 1.54 m (76.2%), speed is 0.36 m/s (52.9%), and heading is 4.2° (54.9%).

3) Lane Keeping With Acceleration & Deceleration: This scenario includes acceleration and brake motions by the target vehicle with lane keeping. Fig. 21 depicts the model switching with L_1 and L_2 estimation. Similar to the lane keeping with constant speed scenario, the model number is not changed frequently, and L_1 , L_2 also successfully estimates the target length and width.

Fig. 22 describes the estimation errors for the two algorithms. Since the target vehicle longitudinal motion is extremely changed in this scenario, the position and speed errors by the tracking algorithms increased as compared to the previous road driving scenarios. As acceleration and deceleration repeats, the position and speed errors fluctuate. However, as in the other scenarios, the error fluctuation and position error of the proposed algorithm is much smaller than that of the box tracking method, due to the size estimation.

The RMS error values of the box tracking algorithm for the position is 2.2 m, speed is 1.73 m/s, and heading is 3.33°. The RMS errors of the L-shape tracking algorithm are 0.7m, 1.34 m/s, and 1.99° for position, speed, and heading, respectively. The reduction of errors for the position is

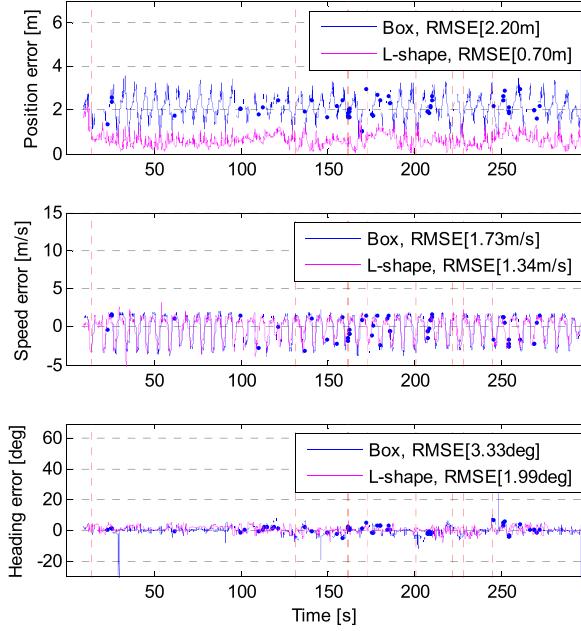


Fig. 22. Estimation errors for the position, speed, and heading of the L-shape and box tracking algorithms in the lane keeping with acceleration & deceleration scenario.

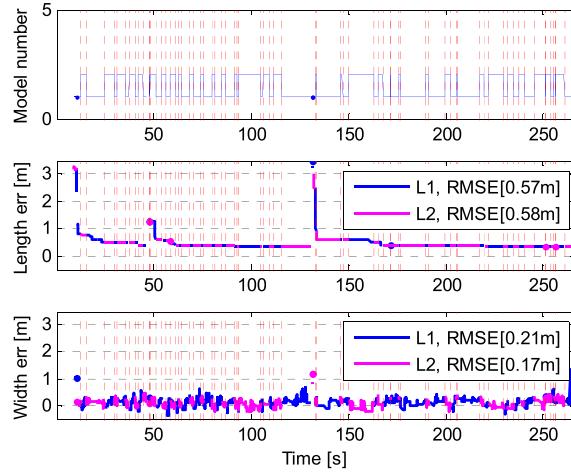


Fig. 23. Model number and estimation results for L_1 and L_2 length in the lane change with the acceleration & deceleration scenario.

1.5 m (68.1%), speed is 0.39 m/s (22.5%), and heading is 1.34° (40.2%).

4) Lane Change With Acceleration & Deceleration: In this scenario, the target vehicle shows the most complex longitudinal and lateral motions. Just as in the lane change with constant speed scenario, the track model frequently switches due to the lane change motion of the target. Fig. 23 represents the model number changes with target size estimations. As another lane change scenario, the model number changes frequently and the size of the target is estimated to be slightly smaller.

Fig. 24 depicts the estimation errors of the two methods. In this scenario, the estimation errors reflect both the lane change characteristics and the extremely dynamic longitudinal motion characteristics of the previous scenarios. First, like the other lane change scenarios, frequent appearance changes

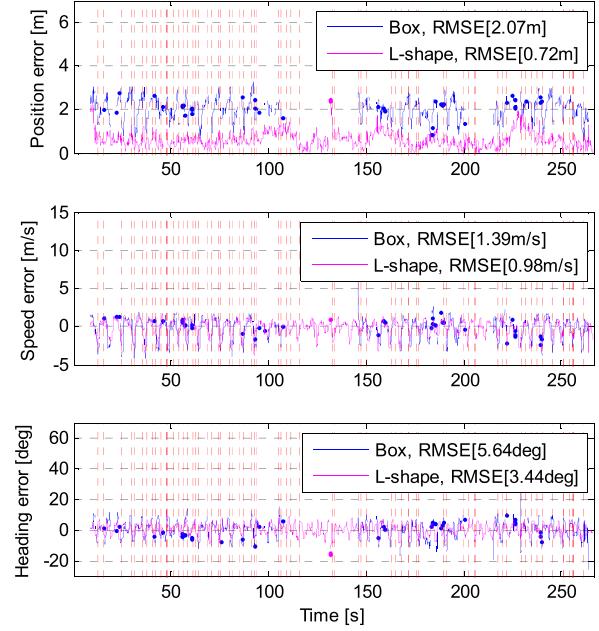


Fig. 24. Estimation errors for the position, speed, and heading of the L-shape and box tracking algorithms in the lane change with acceleration & deceleration scenario.

due to the self-occlusion of the target side degrades the performance of the box tracking algorithm. Second, fluctuating speed errors exist in both algorithms due to the frequent speed changes. However, despite these conditions, the proposed method still provides better estimation performance than the box tracking algorithm.

The RMS errors of the box tracking algorithm are 2.07 m, 1.39 m/s, and 5.64° for position, speed, and heading, respectively. The RMS error values of the proposed algorithm are 0.72 m for position, 0.98 m/s for speed, and 3.44° for heading. As a result, the error reduction by the proposed algorithm is 1.35 m for the position (65.2%), 0.41 m/s for speed (29.5%), and 2.2° for heading (39%).

VI. CONCLUSION

We proposed a vehicle tracking algorithm based on L-shape model switching to overcome the appearance change problem. The proposed algorithm estimates the dynamic states of the corner point and shape information for the L-shape feature independently. Since the position of the corner point is invariant to the size changes of the track, the algorithm can successfully eliminate unexpected position and motion errors by appearance changes. The proposed method was evaluated in several experimental scenarios that provided explicit appearance change problems and real road situations. The experimental results show that the proposed method can significantly reduce errors in position, speed, and heading angle along with mitigating the appearance change problem, as compared to the box tracking algorithm. In particular, the position estimation performance was dramatically improved by reducing the RMS error by about 60 to 70%. The speed error reduction is relatively small in RMS error value, but unexpected errors due to appearance changes are successfully mitigated in lane change scenarios.

However, the performance of this algorithm is directly related to feature extraction results, much like other laser scanner based tracking algorithms. For the best performance of this method, precise corner point position and heading angle extraction is required. Unfortunately, the L-shape model assumes that the vehicle shape is a rectangle, but the actual corner of a vehicle is smoothly curved. Therefore, in future work, a precise corner detecting algorithm for smoothly curved corners should be researched in order to enhance the performance of the proposed algorithm.

REFERENCES

- [1] J. Kichun, K. Junsoo, K. Dongchul, J. Chulhoon, and S. Myoungho, "Development of autonomous car—Part I: Distributed system architecture and development process," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7131–7140, Dec. 2014.
- [2] J. Kichun, K. Junsoo, K. Dongchul, J. Chulhoon, and S. Myoungho, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Aug. 2015.
- [3] J. Kichun, C. Keonyup, K. Junsoo, and S. Myoungho, "Distributed vehicle state estimation system using information fusion of GPS and in-vehicle sensors for vehicle localization," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 2009–2014.
- [4] J. Kichun, C. Keonyup, L. Kangyoon, and S. Myoungho, "Integration of multiple vehicle models with an IMM filter for vehicle localization," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 746–751.
- [5] J. Kichun, C. Keonyup, and S. Myoungho, "GPS-bias correction for precise localization of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 636–641.
- [6] J. Kichun, C. Keonyup, and S. Myoungho, "Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 329–343, Mar. 2012.
- [7] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 475–485, Sep. 2009.
- [8] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Auto. Robots*, vol. 26, nos. 2–3, pp. 123–139, 2009.
- [9] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.
- [10] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
- [11] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 1929–1934.
- [12] C. Premeida and U. Nunes, "Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications," *Robotica*, pp. 17–25, 2005.
- [13] Y. Bar-Shalom, *Tracking and Data Association*. San Diego, CA, USA: Academic, 1987.
- [14] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications*, vol. 1. Norwood, MA, USA: Artech House, 1990, p. 391.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents). Cambridge, MA, USA: MIT Press, 2005.
- [16] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA: Univ. North Carolina Chapel Hill, 1995.
- [17] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.
- [18] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.
- [19] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, vol. 685. Norwood, MA, USA: Artech House, 1999.
- [20] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.
- [21] D. Musicki and R. Evans, "Joint integrated probabilistic data association—JPDA," in *Proc. 5th Int. Conf. Inf. Fusion (FUSION)*, vol. 12, Jul. 2002, pp. 1120–1125.
- [22] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
- [23] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 138–150, Feb. 1996.
- [24] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao–Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [25] D. Schulz, D. Fox, and J. Hightower, "People tracking with anonymous and ID-sensors using Rao–Blackwellised particle filters," in *Proc. IJCAI*, 2003, pp. 921–928.
- [26] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.
- [27] R. MacLachlan, "Tracking moving objects from a moving vehicle using a laser scanner," Robot. Inst., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-05-07, 2005.
- [28] D. Wittmann, F. Chucholowski, and M. Lienkamp, "Improving lidar data evaluation for object detection and tracking using *a priori* knowledge and sensorfusion," in *Proc. 11th Int. Conf. Informat. Control, Autom. Robot. (ICINCO)*, Sep. 2014, pp. 794–801.
- [29] X. Shen, S. Pendleton, and M. H. Ang, "Efficient L-shape fitting of laser scanner data for vehicle pose estimation," in *Proc. IEEE 7th Int. Conf. Cybern. Intell. Syst. (CIS), IEEE Conf. Robot., Autom. Mechatronics (RAM)*, Jul. 2015, pp. 173–178.
- [30] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 54–59.
- [31] P. Broßeit, M. Rapp, N. Appenrodt, and J. Dickmann, "Probabilistic rectangular-shape estimation for extended object tracking," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 279–285.
- [32] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 1836–1843.
- [33] J. Elfring, R. Appeldoorn, and M. Kwakernaak, "Multisensor simultaneous vehicle tracking and shape estimation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 630–635.
- [34] K. Wyffels and M. Campbell, "Precision tracking via joint detailed shape estimation of arbitrary extended objects," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 313–332, Apr. 2017.
- [35] D. Streller and K. Dietmayer, "Multiple hypothesis classification with laser range finders," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2004, pp. 195–200.
- [36] D. Streller and K. Dietmayer, "Object tracking and classification using a multiple hypothesis approach," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 808–812.

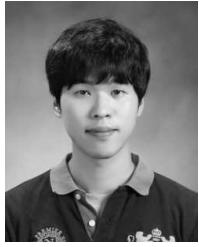


Dongchul Kim (M'14) received the B.S. degree in electronics and computer engineering and the M.S. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree with the Automotive Control and Electronics Laboratory. His current research interests are in detection and tracking of moving object, and information fusion. His research activities include the design of autonomous driving system, distributed control systems, in-vehicle networks, and real-time embedded software development.



Kichun Jo (S'10–M'14) received the B.S. degree in mechanical engineering and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2008 and 2014, respectively. From 2014 to 2015, he was with the Automotive Control and Electronics Laboratory, Department of Automotive Engineering, Hanyang University, where he was involved in system design and implementation of autonomous cars. Since 2015, he has been with the Valeo Driving Assistance Research, Bobigny, France, involved in the highly automated driving.

His main fields of interest are localization and mapping, objects tracking, information fusion, vehicle state estimation, behavior planning, and vehicle motion control for highly automated vehicles. His current research activates include hardware and software platform design of autonomous cars based on distributed real-time embedded systems and in-vehicle networks.



Minchul Lee (S'14) received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, in 2013, where he is currently pursuing the Ph.D. degree with the Automotive Control and Electronics Laboratory. His main fields of interest are precise positioning and localization, vehicle motion assessment, information fusion theories, real-time systems for autonomous car and the system of autonomous vehicle. His current research activities include vehicle motion assessment for an autonomous car.



Myoungho Sunwoo (M'81) received the B.S. degree in electrical engineering from Hanyang University in 1979, the M.S. degree in electrical engineering from The University of Texas at Austin in 1983, and the Ph.D. degree in system engineering from Oakland University in 1990.

He joined General Motors Research (GMR) Laboratories, Warren, MI, in 1985, and involved in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he was involved in the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has been leading research activities as a Professor with the Department of Automotive Engineering, Hanyang University. His research has focused on automotive electronics and controls, such as modeling and control of internal combustion engines, design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs.