# Moving Object Detection and Tracking Based on Interaction of Static Obstacle Map and Geometric Model-Free Approach for Urban Autonomous Driving

Hojoon Lee, Jeongsik Yoon, Yonghwan Jeong, and Kyongsu Yi, *Member, IEEE*

***Abstract*— Detection and tracking of moving objects (DATMO) in an urban environment using Light Detection and Ranging (LiDAR) is a major challenge for autonomous vehicles due to sparse point cloud, multiple moving directions, various traffic participants, and computational load. To address the complexity of this issue, this study presents a novel model-free approach for DATMO using 2D LiDAR implemented on autonomous vehicles. The approach has been used to classify moving points in the point cloud using the predicted Static Obstacle Map (SOM) generated via interaction between Geometric Model-Free Approach (GMFA) and SOM, and estimates the state of each moving object via GMFA. The motion of each point represented by the state of moving objects updates the SOM. The interaction between GMFA and SOM estimates the correspondence between consecutive point clouds in real-time. The proposed approach has been evaluated via RT range and labeled dataset. The accuracy of estimation of the yaw angle and the velocity of a moving vehicle has been quantitatively evaluated using the RT-range. The performance is significantly improved compared with the geometric model-based tracking (MBT). The estimation of the yaw angle, which has a significant effect on the cut-in/cut-out intention of the target vehicle, is shown to be remarkably improved. Based on the evaluation of the labeled dataset, false-positive and false-negative features are suppressed more than MBT.**

***Index Terms*— Autonomous vehicles, DATMO, sparse point cloud, model free tracking, LiDAR.**

## I. Introduction

DUE to the generalization of driver assistance systems and aging-related limitations of drivers [1], autonomous driving has been studied continuously in recent decades. Various sensors such as radar [2], [3], Light Detection and Ranging (LiDAR) [4]–[7], and cameras [8]–[10] have been studied to perceive the surrounding environment. Among these sensors, LiDAR plays the most important role in autonomous driving due to the high resolution and accuracy of distance. Therefore, the sensor has been widely used for high-definition (HD) map construction and map based localization [11], simultaneous localization and mapping (SLAM) [12], detection and tracking of moving objects (DATMO) [13], object perception and classification [14].

The framework to solve DATMO using LiDAR can be classified into two types. First, the *detect before track* framework has been used to identify the object using a feature extraction or deep learning in the current scan, followed by tracking based on the detections. Second, the *track before detect* framework establishes the correspondence between points of consecutive scans and tracks moving objects via the correspondence before detection.

*Detect before track* framework can be divided into two categories. First, the features in the point cloud are extracted via traditional methods to create possible hypotheses based on features and are available to track the hypotheses. After clustering the point cloud, each cluster was represented by lines via recursive line fitting and the features were tracked in [15]. The clusters were converted into edge targets to generate the hypotheses for tracking in [16]. These methods are limited by the reliance on clustering for object detection, and the need to increase the number and type of hypothesis to manage various types of objects.

The second category is used for direct detection of objects from point cloud via deep learning and tracking the detections. Multi-View 3D network for sensory-fusion frameworks proposed by a previous study [17]. The network resulted in an average precision of 0.88 from KITTI benchmark dataset. Despite the breakthrough detection rate based on deep learning, the limitations associated with *Detect before track* framework are obvious. Detection using point cloud of current scan leads to wrong shape via mix with a stationary object, sparse point cloud, and occlusion.

*Track before detect* framework is an effective alternative. The framework is divided into two categories. First, the geometric model-based tracking (MBT) uses a likelihood-based measurement model and update using the likelihood between several particles and point cloud. Bayes filter and measurement

Manuscript received May 29, 2019; revised August 22, 2019, November 14, 2019, and January 30, 2020; accepted March 9, 2020. This work was supported in part by the Brain Korea 21 Plus Project in F14SN02D1310, in part by the Technology Innovation Program (10079730, Development and Evaluation of Automated Driving Systems for Motorway and City Road and driving environment) funded by the Ministry of Trade, Industry and Energy (MOTIE, South Korea), in part by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2016R1E1A1A01943543, and in part by the Ministry of Land, Infrastructure, and Transport (MOLIT, KOREA) [Project ID: 18TLRP-B146733-01, Project Name: Connected and Automated Public Transport Innovation (National Research and Development Project)]. The Associate Editor for this article was B. Fidan. *(Corresponding author: Kyongsu Yi.)*

The authors are with the Department of Mechanical Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: hj.lee091011@gmail.com; wjdtlr1915@snu.ac.kr; winqq1234@snu.ac.kr; kyi@snu.ac.kr).

Digital Object Identifier 10.1109/TITS.2020.2981938

models were utilized to estimate the geometry and dynamic state of a moving vehicle [18]. The virtual ray-based measurement model and the Rao-Blackwellized particle filter (RBPF) facilitate multi-vehicle tracking without clustering or data association, and lead to simultaneous estimation of vehicle shape. However, only vehicles within 50m were detected, and cyclists with a different shape of rectangle were not detected. If the vehicular side was not measured, errors in the yaw angle estimation occurred. They assumed the prior knowledge of the road information. In [19], the pose estimation based on coherent point drift using likelihood-field-based model was proposed and its recall and $F_1$ score reached 0.86 and 0.35 from KITTI benchmark dataset, respectively. However, buses or cyclists could not be detected because they assumed the fixed size and shape of the vehicle.

The second category establish the correspondence between points in successive scans. In [20], the relationship between the neighboring points of each pixel was expressed using the feature vector via local descriptor and the correspondence between the pixels of the previous step and the current step was established through the closest neighbor. Kaestner *et al.* [21] proposed the generative object detection algorithm via direct point state estimation but it run at about 0.5Hz. Detection based on the correspondence between each point in successive scans is a powerful method theoretically. However, in reality, it is computational demanding to directly establish the correspondence for more than 10,000 points generated in each scan as noted [21]. In order to overcome this problem, there have been attempts to efficiently detect moving objects by estimating stationary obstacles. Ferri *et al.* [22] detected moving obstacles through ray-casting on spherical voxelization and Zhong *et al.* [23] distinguished static obstacles through the temporal and spatial correlation. These studies only tested in low-speed robot (below 20 kph). Wang *et al.* [24] used global static background map to distinguish point clouds belong static obstacles. Vaquero *et al.* [25] accumulated data for one second (10 frames) and tracked all clusters using Multiple Hypothesis Tracking (MHT) to determine static objects. When data accumulates for a long time, it is difficult to apply it in a high-speed, because it takes a long time to determine the static objects. It is also inefficient in urban environments because it clusters all point clouds without using information previously determined to be static. For this reason, the algorithm was only tested in the low-speed truck.

This study proposes a novel approach to overcome the drawbacks of *track before detect* as mentioned before. The real-time application is available via efficient representation of static obstacles by Static Obstacle Map (SOM) and tracking moving objects using a Geometric-Model-Free Approach (GMFA). In the proposed approach, the static obstacles are represented by SOM in order to utilize *track before detect* in real-time. As a result, the number of points to establish the correspondence in the consecutive scan is drastically decreased and the real-time features are satisfied. The proposed approach tracks and detects moving objects ($>$13.5 kph) regardless of shapes and partial occlusion using GMFA. SOM is estimated via simple Bayes filter using ego-motion but in conjunction with GMFA, it can be used to accurately represent static obstacles. The three main contributions of this study are as follows:

1) The feedback structure of SOM and GMFA leads to efficient representation of the surrounding static objects and rapid correlation establishment between consecutive scans. The approach also facilitated the utilization of all the measured points in real-time without selecting the region of interest (ROI).

2) The proposed approach is robust with respect to object shape, sparse points due to long-distance ($>$40m), and partial occlusion resulting in an increase in $F_1$ score.

3) The estimated speed and yaw angle depend on the movement between the corresponding clusters with successive scans. In all scenarios including lane change, the estimation accuracy of speed and yaw angle was improved.

The proposed approach was verified using vehicle tests carried out in a MATLAB-Labview environment with a PC equipped with an i7-4790 4.00GHz CPU. The results of detection and the real-time characteristics of the algorithm were verified using a labeled dataset obtained via vehicle tests on urban roadways in Seoul, Korea. The estimation accuracy was analyzed under lane changes and lane keeping scenarios through RT range. The accuracy of detection and estimation was compared with the MBT results for the driving data.

The remainder of this study is organized as follows. In section II, the mathematical formulation and the algorithm of moving object detection via interaction between SOM and GMFA is discussed. In section III, SOM prediction and update are described. In section IV, the prediction, measurement update and track management of GMFA are detailed. In section V, the proposed approach is analyzed using driving datasets derived from the actual urban environment. Finally, section VI presents concluding remarks and future perspective.

## II. Interaction Between Static Obstacle Map and Geometric Model Free Approach

We present a mathematical formulation to the problem and outline an algorithm of our approach to elucidate and estimate real-time application. The following variables are used to describe the problem.

| | |
|---|---|
| $[k]$ | : k-th time steps |
| $\hat{x}, \bar{x}$ | : Estimation and Prediction of the variable $x$ |
| $O$ | : True states of the moving objects |
| $P_{host}$ | : Covariance of ego vehicle |
| $P_n$ | : Covariance of n-th track |
| $x_{host}$ | : Dynamic state of ego vehicle |
| $x_{static}$ | : Motion state of SOM (static $=1$, unknown $=0$) |
| $x_n$ | : States of n-th track |
| $\mathbf{Y}$ | : Point cloud from LiDAR |
| $\mathbf{Y}_m$ | : Measurements of moving objects |
| $\mathbf{Y}_s$ | : Measurements of static objects |
| $\mathbf{Y}_{moving}$ | : Predicted points to move by SOM |
| $z_{static}$ | : Measurements for SOM |
| $\mathbf{Z}$ | : Set of the clusters from current scan |
| $\mathbf{Z}_n$ | : Validated clusters for measurement of n-th track |
| $\hat{\mathbf{Z}}_n, \bar{\mathbf{Z}}_n$ | : Updated and predicted cumulative cluster of n-th track |

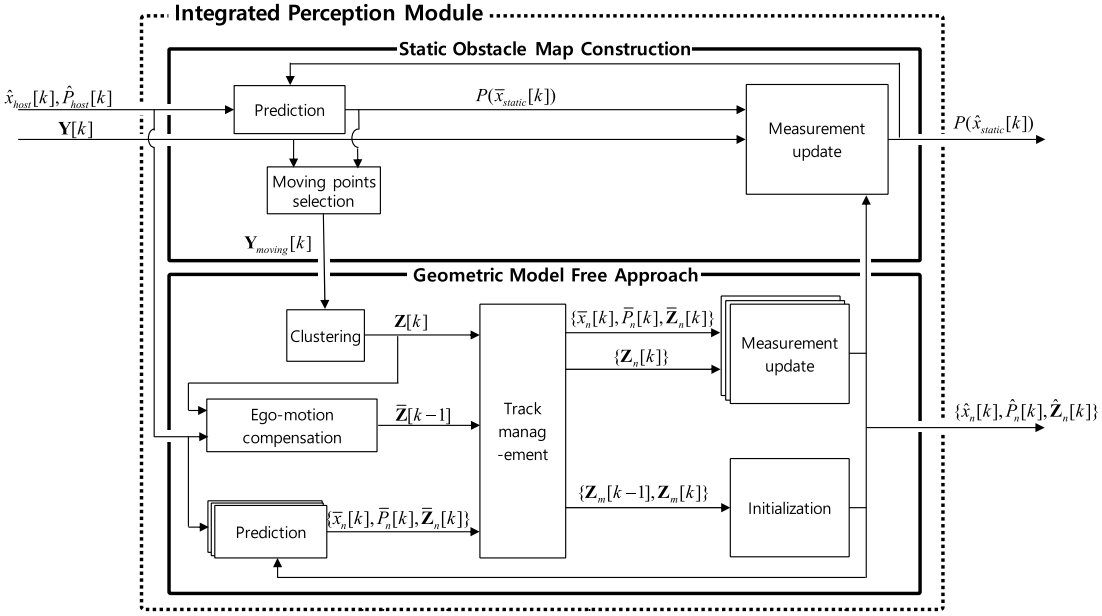



Fig. 1. Integrated perception module via interaction between Static Obstacle Map and Geometric Model Free Approach.

The goal of our approach is to compute the posterior probability $P(x_{static}[k], O[k]|x_{host}[k, \dots, 0], \mathbf{Y}[k, \dots 0])$. Since SOM represents the local static environment, the dynamic states of ego vehicle are considered as given and the global position of ego vehicle is not estimated. Three assumptions have been used for real-time computation of the posterior as follows:

1) The measurements can be decomposed into static objects and moving objects ($\mathbf{Y} = \mathbf{Y}_m \cup \mathbf{Y}_s$)

2) The proposed measurements satisfy the following equations.

$$\begin{aligned} P\{(\mathbf{Y}_m[k]|x_{static}[k])|O[k]\} &= P(\mathbf{Y}_{moving}[k]|O[k]) \\ P\{(\mathbf{Y}_s[k]|O[k])|x_{static}[k]\} &= P(z_{static}[k]|x_{static}[k]) \end{aligned} \quad (1)$$

3) The dynamics of static and moving objects are independent.

The assumptions 1) and 3) were proposed by Wang *et al.* [26], and 2) is proposed in this study. Since $\mathbf{Y}_{moving}$ is the points included in the low-probability grid of the predicted SOM and $z_{static}$ contains the moving object as a negative measurement, 2) is feasible. Using Bayes' rules and assumptions, the posterior can be expressed by the general recursive Bayesian formula (2), which shows the interaction between GMFA and SOM and their simultaneous feedforward mechanism.

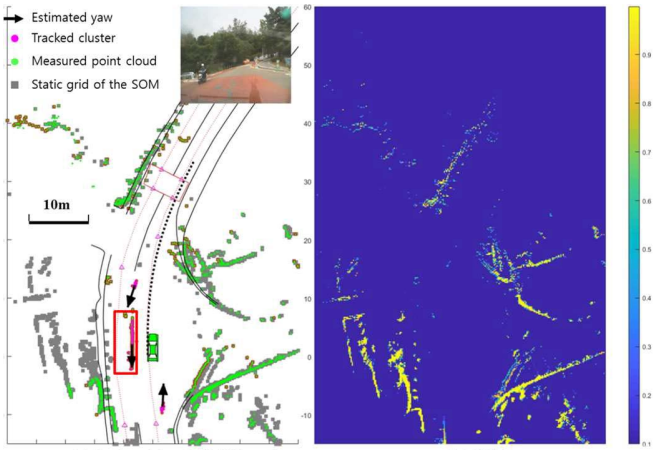
Fig. 1 shows the structure of the integrated perception module consisting of SOM construction and GMFA to detect and track moving objects without a direct correlation between each point of the consecutive point clouds. The integrated perception module generates SOM ($P(\hat{x}_{static}[k])$), $\{\hat{x}_n[k], \hat{\mathbf{Z}}_n[k]\}$ from $\mathbf{Y}[k]$, $\hat{x}_{host}[k]$. SOM refers to the probability of static objects in each grid space around the vehicle. It is 1 when the grid is static, and 0 when it is moving, free, or unknown.

$$\begin{aligned} &P(x_{static}[k], O[k]|x_{host}[k, \dots, 0], \mathbf{Y}[k, \dots 0]) \\ &\quad \propto P(\mathbf{Y}[k]|x_{static}[k], O[k]) \\ &\qquad \cdot P(x_{static}[k], O[k]|x_{host}[k, \dots, 0], \mathbf{Y}[k-1, \dots, 0]) \\ &\quad = P\{(\mathbf{Y}_m[k]|x_{static}[k])|O[k]\} \\ &\qquad \cdot P\{(\mathbf{Y}_s[k]|O[k])|x_{static}[k]\} \\ &\qquad \cdot P(O[k]|x_{host}[k, \dots, 0], \mathbf{Y}_m[k-1, \dots, 0]) \\ &\qquad \cdot P(x_{static}[k]|x_{host}[k, \dots, 0], \mathbf{Y}_s[k-1, \dots, 0]) \\ &\quad = \underbrace{P(\mathbf{Y}_{moving}[k]|O[k])}_{\text{GMFA Update}} \\ &\qquad \cdot \underbrace{P(O[k]|x_{host}[k, \dots, 0], \mathbf{Y}_{moving}[k-1, \dots, 0])}_{\text{GMFA Prediction}} \\ &\qquad \cdot \underbrace{P(z_{static}[k]|x_{static}[k])}_{\text{SOM Update}} \\ &\qquad \cdot \underbrace{P(x_{static}[k]|x_{host}[k, \dots, 0], z_{static}[k-1, \dots, 0])}_{\text{SOM Prediction}} \end{aligned} \quad (2)$$

The first step in our approach is the prediction of SOM using $P(\hat{x}_{static}[k-1])$, $\hat{x}_{host}[k]$, and $\{\hat{x}_n[k-1], \hat{\mathbf{Z}}_n[k-1]\}$. Going forward, it will be covered comprehensively in Section III. Using $P(\bar{x}_{static}[k])$, $\mathbf{Y}_{moving}[k]$ is collected among the current LiDAR point cloud and transferred to the GMFA.

The term $\mathbf{Z}[k]$ is generated from $\mathbf{Y}_{moving}[k]$ via Euclidean clustering compared to the tracks, $\{\bar{x}_n[k], \bar{\mathbf{Z}}_n[k]\}$, and the clusters are assigned to the existing tracks for measurement or creation of new track. $\hat{x}_n[k]$ is estimated using iterative closest point (ICP) and extended Kalman filter (EKF), which will be discussed further in Section IV.

The motion state of the points is classified into Static and Moving types according to the estimated speed. The other points are considered as Unclassified, e.g. the points classified static via predicted SOM, $\mathbf{Y}^c_{moving}[k]$, and not included in $\hat{\mathbf{Z}}_n[k]$




(a) Static grid and GMFT result (b) SOM

Fig. 2. Histogram representation of SOM. (a) Point cloud, high static probability grids, and tracking result. (b) The static probability of each grid.

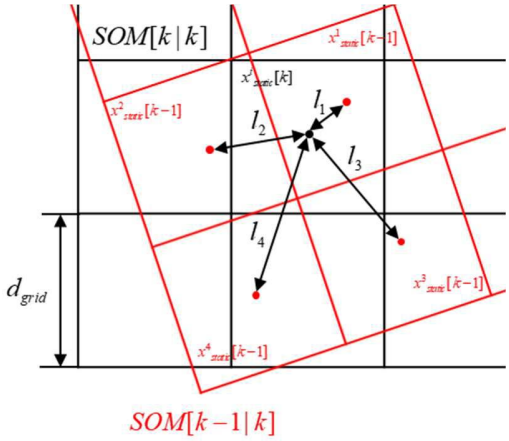



Fig. 3. Relationship between the SOM of previous and current step.

among $\mathbf{Y}_{moving}[k]$. The static probability of each grid of SOM, $P(x^j_{static}[k] = 1)$, is updated using a Bayes filter with the motion state of the points included in each grid, which will be discussed further in Section III.

It is theoretically correct to establish the correlation between consecutive points in the successive scan using Signature of Histograms of OrienTations (SHOT) descriptor [7] or Conditional Random Fields [27] and clustering point cloud. However, it is impossible to specify the number of iterations required for convergence, and the high computational complexity is required for practical application. In this context, the proposed approach facilitates the estimation of the motion state of each point and clustering of the points from similar moving objects, without establishing the correspondence between each point in successive scans.

## III. Static Obstacle Map Construction

SOM proposed in this study is a grid representation of the local coordinate system in the ego vehicle involving $P(x^j_{static}[k] = 1)$, which is different from the SLAM-based occupancy grid map proposed in the previous study [28]. The occupancy grid map classifies each grid into Static, Free, and Occluded states, after discretizing the space in the global coordinate. Our goal is to map the unknown space and estimate the state of the ego vehicle. However, SOM represents the environment around the ego vehicle locally using a Static and Unknown (including Moving, Occluded, and Free) grid. SOM is proposed to distinguish the static and the moving LiDAR points, and to efficiently represent static obstacles with various shapes including accident vehicles, scaffolds, and cones.

SOM is a grid map with the static probability of each grid bound between 0.05 and 0.95. Likewise, Fig. 2-(b) represents the SOM as a two-dimensional histogram, and the point cloud and tracking results are shown in Fig. 2-(a). In these terms, the errors from moving objects are corrected via interaction with GMFA. If the points belong to a moving object next to the ego vehicle as shown in the red box in Fig. 2-(a), the grids containing the points are considered static without interaction with GMFA. However, the points are considered as moving via interaction with GMFA, which decreases the static probability of the moving object region, as shown in Fig. 2-(b). With this in mind, SOM is intended to represent a static environment efficiently and to improve the detection and tracking performance of moving objects. Therefore, SOM is configured locally and predicted using only Inertial Navigation System (INS). Accordingly, it is noted that its accuracy is not the primary focus of our study.

During the initialization phase, SOM is initialized to 0.05. The static estimation of points through GMFA involves at least 2 steps, and the moving estimation is based on at least 7 steps, so SOM is initially updated in the same manner as scan differencing.

### A. Prediction of Static Obstacle Map

Since SOM represents the local environment, the ego motion in consecutive steps must be corrected depending on the state of the ego vehicle using the method shown in Fig. 3. The red grid represents the grid in the previous SOM corrected via ego motion, and the black grid refers to the grid in the current SOM. When calculating the probability of the j-th grid in the current step, the four grids of the previous SOM can be determined according to the distance from the midpoint of the current j-th grid, $l_i$ $[i = 1, 2, 3, 4]$, less than $\sqrt{2}\, d_{grid}$. Therefore, the predicted probability of the j-th grid is calculated by the weighted average of the probability of the four grids whose distance is less than $\sqrt{2}\, d_{grid}$ as shown in (3). If $l_i = 0$, the j-th grid is physically equivalent to the previous grid, such that the probability of the j-th grid reflected the probability of the previous grid.

It is assumed that each grid is independent without transition between states when SOM is predicted. The lack of transition probability between states is based on the absence of valid state transition matrix because the position physically represented by each grid changes with the motion of the ego vehicle. In addition, since the probability of each grid is bounded, it can be estimated accurately via measurement update despite the

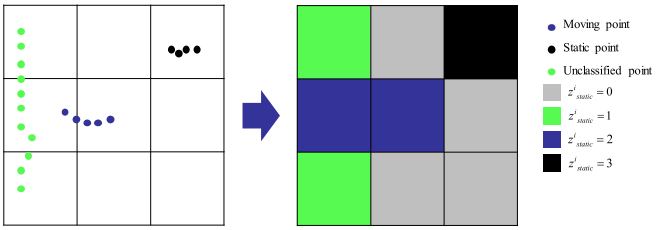



Fig. 4. Measurements of the motion state in each grid according to the motion state of the points contained in the grid.

lack of state transition.

$$L=\sum_{i=1}^{4} l_i^{-1}$$

$$P(\bar{x}^j_{static}[k])=\begin{cases}\sum_i \frac{l_i^{-1}}{L} P(\hat{x}^i_{static}[k-1]) & (l_i\neq 0)\\ P(\hat{x}^i_{static}[k-1]) & (l_i=0)\end{cases} \quad (3)$$

### *B. Measurement Update of Static Obstacle Map*

The measurement update refers to updating the probability of a stationary object in each grid of the predicted SOM via the motion state of each point classified by GMFA. According to the motion state of the points included in the j-th grid, the measurement of j-th grid ($z^j_{static}$) is determined as shown in Fig. 4. One of the following four values can be used to measurement of the j-th grid: Free, 0; Unclassified, 1; Moving, 2; and Static, 3. The measured value is 2 for moving points, 3 for static points without moving points, 1 for only unclassified points, and 0 if there is any point. As shown in Fig. 4, the blue and black dots indicate moving and static motion states, respectively, through GMFA. The green dot indicates that the unclassified state of motion. When the points on each grid are similar to the left, the motion measurement of each grid is determined as shown on the right side. The gray, green, blue, and black colors on the grid represent free, unclassified, moving, and static motion measurements, respectively. The measurement model is valid because the grid is small enough that the motion of each grid can be expressed as a single state.

After each grid is measured as described above, the motion state of each grid can be estimated using the (4) with 1st order Markov assumption via the predicted SOM and the likelihood of measurement. The likelihood is predetermined as TABLE I. It is tuned through the actual data stream.

$$\begin{aligned}&P(\hat{x}^j_{static}[k]=1)\\ &=P(x^j_{static}=1|z^j_{static}[k],\ldots,z^j_{static}[0])\\ &=\frac{P(z^j_{static}[k]|x^j_{static}=1)P(\bar{x}^j_{static}[k]=1)}{\sum_{i=0,1}P(z^j_{static}[k]|x^j_{static}=i)P(\bar{x}^j_{static}[k]=i)}\end{aligned} \quad (4)$$

## IV. Geometric Model-Free Approach for Tracking of Moving Objects

The Geometric Model-Free Approach (GMFA) uses $\mathbf{Y}_{moving}[k]$ to track the moving objects and estimate their states. Thus, it is possible to construct the correspondence between non-static points in the consecutive scan and to update the SOM based on the motion state of each point. In our approach, compared with previous studies, each point depended on clustering using Euclidean distance. Since the correspondence between points is determined by the distance between the mean points of the cluster and the similarity of shape, the correspondence between points in consecutive scans can be established even with a small calculation. After establishing the correspondence, matching for each cluster is performed using ICP, and the states of the moving objects are estimated via EKF based on the moving distance and direction of the cluster mean.

TABLE I

The Likelihood of Static Obstacle Map

| $x^j_{static}$ \ $z^j_{static}$ | Free (0) | Unclassified (1) | Moving (2) | Static (3) |
|---|---|---|---|---|
| Unknown (= 0) | 0.30 | 0.14 | 0.33 | 0.23 |
| Static (= 1) | 0.15 | 0.47 | 0.01 | 0.37 |

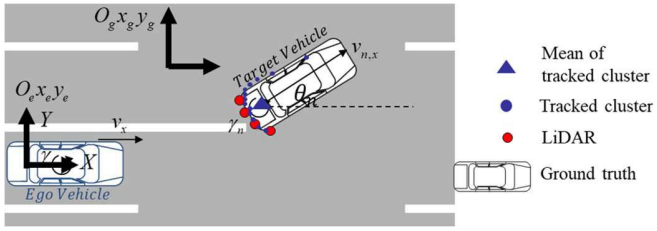



Fig. 5. States of moving object for Geometric Model Free Approach.

GMFA uses two coordinates: $O_g x_g y_g$, which is a fixed global coordinate system, and $O_e x_e y_e$, which is a local moving coordinate system that moves with the rear axle of the ego vehicle (Fig. 5). There are seven states, $x_n = \left[\, p_{n,x} \;\; p_{n,y} \;\; \theta_n \;\; v_{n,x} \;\; \gamma_n \;\; a_{n,x} \;\; \dot{\gamma}_n \,\right]^T$, and $\{\bar{\mathbf{Z}}_n, \hat{\mathbf{Z}}_n\}$, which express the $n$-th track, and $p_{n,x}, p_{n,y}$ represent the mean position of the cluster with respect to $O_e x_e y_e$. After completing the measurement update at every step, it is replaced with the new mean point when the cluster point configuration changes $\theta_n$ denotes the yaw angle of the moving object with respect to $O_e x_e y_e$. $v_{n,x}$ indicates the speed in the direction with respect to $O_g x_g y_g$. $\gamma_n$, $a_{n,x}$, and $\dot{\gamma}_n$ indicate the yaw rate, the acceleration, and the angular acceleration with respect to $O_g x_g y_g$, respectively. $v_x, \gamma$ represent speed and yaw rate of ego vehicle at $O_g x_g y_g$ respectively. The cumulative cluster $(\bar{\mathbf{Z}}_n, \hat{\mathbf{Z}}_n)$ is in a queue format and points accumulated more than four steps have been removed.

### *A. Prediction of Geometric Model-Free Approach*

Each track is predicted via process update of the discrete-time EKF using the model (5). Discretization has been accomplished as a second order [29]. All points in $\bar{\mathbf{Z}}_n, \hat{\mathbf{Z}}_n$ carry the same dynamic states. In this case, the shape of the cluster might be changed theoretically, but the impact is limited because LiDAR measures every 80 msec.

It is necessary to convert the previous clusters to current step $O_e x_e y_e$ based on the static assumption to initialize the tracks and estimate the velocity of moving objects. This process

referred to as ego-motion compensation is illustrated in Fig. 1, and the clusters in the previous step, $\mathbf{Z}[k-1]$, are converted to the current step, $\bar{\mathbf{Z}}[k-1]$, using dead reckoning via speed and yaw rate of ego vehicle under the static assumption.

$$\begin{aligned}
\dot{x}_n &= \mathbf{a}\,(x_n, u) + \mathbf{q} \\
&= \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{a}_7 \end{bmatrix}^T + \mathbf{q} \\
u &= [v_x, \gamma\,] \\
\mathbf{a}_1 &= v_{n,x}\cos\theta_n - v_x + p_{n,y}\cdot\gamma \\
\mathbf{a}_2 &= v_{n,x}\sin\theta_n - p_{n,x}\cdot\gamma \\
\mathbf{a}_3 &= \gamma_n - \gamma \;\; \mathbf{a}_4 = a_{n,x} \\
\mathbf{a}_5 &= \dot{\gamma}_n \mathbf{a}_6 = -k_a \mathbf{a}_7 = -k_{\dot{\gamma}} \\
\mathbf{q} &\sim (\mathbf{0}, \mathbf{Q})
\end{aligned} \tag{5}$$

### B. Track Management

Track management refers to the cluster assignment in the current step to the predicted tracks, initializing the tracks using clusters not assigned to the predicted tracks, and discontinuation of the tracks that have yet to be updated for a certain period. The assignment of clusters to the predicted track is performed via Global Nearest Neighbor (GNN). For a comprehensive analysis of track management, we disclose the configuration of $\bar{\mathbf{Z}}[k-1]$, $\mathbf{Z}[k]$, and $\{\bar{\mathbf{Z}}_n[k]\}$. $\bar{\mathbf{Z}}[k-1]$ consists of p clusters, $\{\bar{\mathbf{Y}}_1, \bar{\mathbf{Y}}_2, \cdots, \bar{\mathbf{Y}}_i, \cdots, \bar{\mathbf{Y}}_p\}$, and each $\bar{\mathbf{Y}}_i$ comprises $n_i$ 2D points. $\mathbf{Z}[k]$ and $\{\bar{\mathbf{Z}}_n[k]\}$ also comprise $q$ and $N$ clusters, $\{\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_j, \cdots, \mathbf{Y}_q\}$ and $\{\bar{\mathbf{Z}}_1, \cdots, \bar{\mathbf{Z}}_n, \cdots, \bar{\mathbf{Z}}_N\}$, respectively. The feature vector, $f$, for each cluster, $A$, for GNN is defined previously (6). The feature vector is a 4D vector consisting of a mean point and eigenvalues of covariance matrix of the clusters. The eigenvalues provide shape information independent of rotation. In a 4D feature space, a weighted 2-norm is defined as a distance, and when the distance between $\bar{\mathbf{Z}}_n$ and $\mathbf{Y}_j$ is less than a predefined threshold, $\mathbf{Y}_j$ represents a measure of $n$-th track, $\mathbf{Z}_n$.

$$\begin{aligned}
f &\triangleq [x, y, \lambda_{MAX}, \lambda_{\min}]^T \\
[x, y] &= mean(A) \\
[\lambda_{MAX}, \lambda_{\min}] &= eig(cov(A)) \quad when\ \lambda_{MAX} \geq \lambda_{\min}
\end{aligned} \tag{6}$$

Track initialization and discontinuation are conducted when the assignment of measurements to the predicted tracks is complete. If the track is not updated for more than 30% of the lifetime, or for three steps continuously, the track is discontinued. Track initialization refers to creation of a new track using clusters $(\bar{\mathbf{Y}}_i, \mathbf{Y}_j)$ that are not assigned to a track. If the distance between $\bar{\mathbf{Y}}_i$ and $\mathbf{Y}_j$ is smaller than the predefined threshold, a correspondence is established to generate the new track. $\mathbf{Y}_j$ and $\bar{\mathbf{Y}}_i$ become $\mathbf{Z}_m[k]$ and $\mathbf{Z}_m[k-1]$, respectively, as shown in Fig. 1. The position, yaw, and speed are initialized via ICP matching, and the others initialized to zero.

### C. Measurement Update of Geometric Model-Free Approach

The EKF structure is used to predict the measurement update. Since the process model has been described in Section IV.A, this section will discuss the measurement and the actual calculation from the assigned cluster for $n$-th track, $\mathbf{Z}_n$. In the proposed approach, the three measurements obtained through $\mathbf{Z}_n$ include the position and the yaw angle of the moving objects. When $\mathbf{z}_n$ is a measurement of the EKF, $\mathbf{z}_n$ is expressed in $[h_{n,1}, h_{n,2}, h_{n,3}]^T$ as a 3D vector. The three elements of $\mathbf{z}_n$ represent the mean point and yaw angle of the moving object at $O_e x_e y_e$, respectively. The position of $n$-th track is considered as the mean of the matched $\bar{\mathbf{Z}}_n$ after matching $\bar{\mathbf{Z}}_n$ to $\mathbf{Z}_n$ by ICP. The moving direction of the object refers to the direction of the displacement vector from the mean of $\hat{\mathbf{Z}}_n[k-1]$ to the mean of matched $\bar{\mathbf{Z}}_n$. These measurements are shown in Fig. 6, and the measurement model based on these measurements is linear as shown in (7),

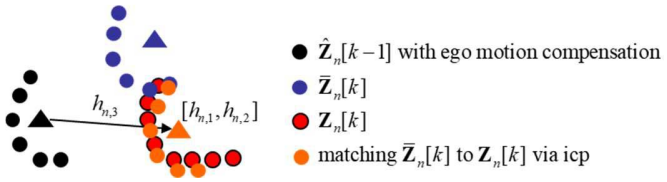



Fig. 6. The measurement of $n$-th track from corresponded cluster. The triangle denotes mean point of each cluster.

$$\begin{aligned}
\mathbf{z}_n[k] &= \mathbf{H}_n x_n[k] + \mathbf{v}_n[k] \\
\mathbf{v}_n[k] &\sim N\,(0,\ \mathbf{V}_n[k]) \\
\mathbf{H}_n &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{7}$$

assuming that the measured values display white Gaussian noise with a covariance matrix of $\mathbf{V}_n$. As shown in Fig. 6, there is no ghost motion due to geometry changes, since the movement of the mean point in the identical cluster is used as a measure.

## V. Vehicle Tests

The proposed approach has been verified by comparing with geometric model-based tracking (MBT). MBT extracts the possible shape candidates of targets from the current point cloud, and tracks the shape candidates using the multiple hypothesis tracking (MHT) framework proposed in [30]. After clustering in the current point cloud, we extract the shape candidates using the bounding box and the virtual ray. The tracks updated continuously for more than three steps are identified as moving objects and we only treat the points on the road for MBT using a pre-configured environment map to prevent mixing with static obstacles. Various studies reported object detection using point cloud, however, based on real-time characteristics, the proposed approach is compared with the MBT.

To obtain the detection results, the driving data from Nambu-Beltway and Seoul National University (SNU) campus using the Ioniq described in Section V.A were labeled with moving objects and analyzed via Precision, Recall, and $F_1$ scores. Data of $80\text{m} > x > -15\text{m}$, $|y| < 25\text{m}$ were used for labeling using a front camera. The estimation accuracy of GMFA was verified via comparison with MBT based on

TABLE II
THE MOVING OBJECT DETECTION RESULT OF GMFA AND MBT

| Method | Dataset | Moving Objects | Detected Objects | Correctly Detected | Precision | Recall | $F_1$score |
|---|---|---|---|---|---|---|---|
| GMFA | Nambu-Beltway | 3915 | 3828 | 3508 | 0.916 | 0.896 | 0.906 |
| | SNU Campus | 540 | 568 | 486 | 0.856 | 0.900 | 0.877 |
| | Overall | 4455 | 4396 | 3994 | **0.909** | **0.897** | **0.902** |
| MBT | Nambu-Beltway | 3915 | 3359 | 2690 | 0.801 | 0.687 | 0.740 |
| | SNU Campus | 540 | 287 | 228 | 0.794 | 0.422 | 0.551 |
| | Overall | 4455 | 3646 | 2918 | **0.800** | **0.655** | **0.720** |

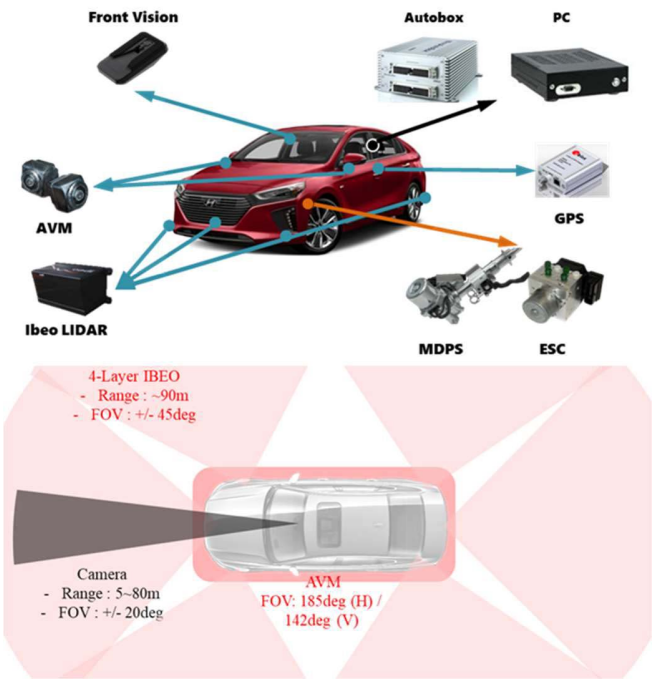
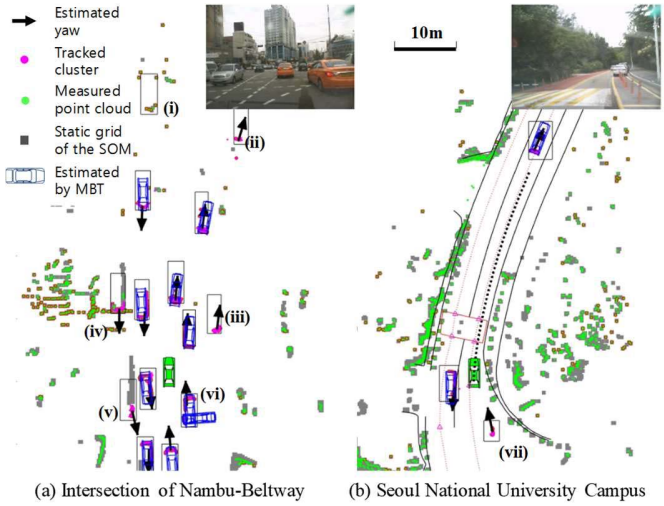



Fig. 7. The vehicle platform to verify the proposed approach. The upper part shows the equipment installed in Ioniq and the lower part shows the FOV of the mounted sensors in bird eye view.




(a) Intersection of Nambu-Beltway (b) Seoul National University Campus

Fig. 8. Comparison of the moving object detection result. (a) depicts a scene of Nambu-Beltway driving data and (b) depicts a scene of SNU campus driving data.

the standard deviation of estimation error. To analyze the estimation errors, the lane keeping (LK) and lane change (LC) driving data were obtained at various relative positions using the RT-range.

### *A. Vehicle Platform*

The vehicle platform used in this study is illustrated in Fig. 7. Ioniq was developed for autonomous driving under urban environment. The wide range of the vehicle was measured using six 4-layer LiDAR, front vision sensor, and Around View Monitoring (AVM) systems. As shown in Fig. 7, the bird's-eye view shows an upper part displaying the equipment installed in Ioniq and the lower portion shows the Field of View (FOV) in the mounted sensors. Ioniq perceives the surroundings in the MATLAB-Labview environment of PC, and operates an algorithm in real time using only i7-4790 4.00GHz CPU.

### *B. Moving Object Detection*

In this section, data obtained at Nambu-Beltway and SNU campus are used to determine the results. Each data frame was labeled with moving objects using a front camera, and 4455 moving objects were labeled including cars, trucks, buses, and motorcycles. In this study, the grid size of SOM was 0.2 m and was labeled as a moving object with a speed of 13.5 kph or more, because an object was considered moving when it moved more than 1.5 grid at a time interval of a consecutive scan of 0.08s on average. The test road included a variety of urban environments such as intersections, pocket roads, speed bumps, and crosswalks.

The detection results are presented in TABLE II. The $F_1$ score of the proposed algorithm is approximately 25% higher than that of MBT because both Precision and Recall are increased, as shown in TABLE II. In this respect Precision is improved by 0.109, and Recall is improved by 0.242. An increase in Precision indicates the reduced number of false alarms, and a significant improvement in Recall suggests a decrease in the frequency of false-negative outcomes.

Fig. 8 depicts a frame for the evaluation of actual data to intuitively explain the difference in the results of detection. In Fig. 8, the green vehicle represents the ego vehicle, whereas the magenta clusters and the black arrows indicate the results of GMFA. The blue vehicles denote the results of MBT, and the black squares represent the label. Here Fig. 8-(a) represents a frame of the Nambu-Beltway dataset and Fig. 8-(b) is a frame of the SNU campus dataset.

In the case of (i) in Fig. 8-(a), the moving object within 3 frames from the first measured value cannot be detected via

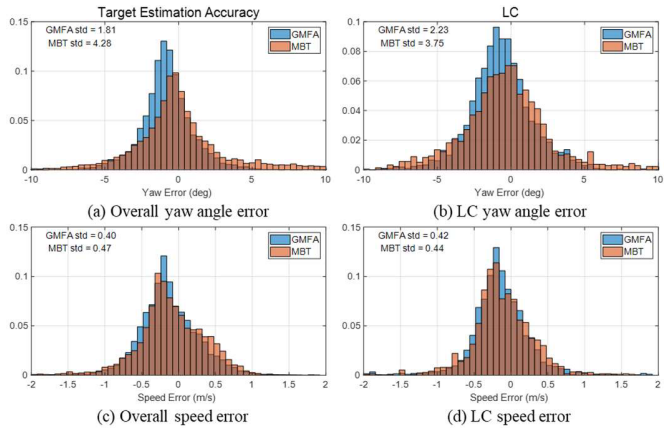



Fig. 9. Estimation error distributions in yaw angle and speed of GMFA and MBT at LC and LK scenarios.

GMFA due to the nature of track before detection. MBT also failed to detect the object (i) because the object was tracked for more than 3 steps as the $F_1$ score was reduced when the shape was immediately established. The components (ii), (iii), (iv), and (v) were detected in the case of GMFA without shape assumption. However, MBT failed to detect the objects because of the difference between the assumed vehicle shape and the actual cluster shape caused by occlusion. In the case of (vi), GMFA accurately perceived a single object. However, in the case of MBT, false positives occurred because all the possible object shapes were generated using point cloud at one instant and tracked using the shapes.

In Fig. 8-(b), both the front and left vehicles were adequately matched with the geometrical assumption of MBT, such that both GMFA and MBT detected accurately. However, in the case of (vii), which represent a motorcycle, MBT undetected because of different shapes compared with the actual shape of the object. However, it was confirmed that GMFA detected accurately without shape assumption. Due to the various forms of traffic and occlusion under an urban environment, moving objects are represented by different shapes that cannot be expressed using a shape model, resulting in higher detection using GMFA without shape assumption compared with MBT.

### *C. Accuracy of Moving Object State Estimation*

In this section, the estimation accuracy of the yaw angle and the speed of moving object were validated by driving data with RT range. The proposed approach was reliable for position analysis as it tracked the point clusters of moving objects directly. The reference data were acquired using RT-range and autonomous vehicles under 10 driving scenarios. Notably, the LK scenarios involved target vehicle driving along the lane at the same speed as the hunter vehicle in the front, front side, rear, and rear sides based on the hunter vehicle speeds of 40 kph and 80 kph. LC scenarios were also determined in which the target changes lanes in the forward and backward directions of the ego vehicle at a driving speed of 40 kph.

Fig. 9 shows the error distribution of yaw angle and speed estimated by GMFA and MBT. The blue and red represent the error distributions of the proposed approach and

TABLE III
THE MOVING OBJECT STATES ESTIMATION RESULT OF GMFA AND MBT

| Method | Standard deviation of | Lane Keeping | Lane Changing | Total |
|---|---|---|---|---|
| GMFA | Yaw angle [deg] | 1.64 | 2.23 | 1.81 |
| | Speed [kph] | 0.40 | 0.42 | 0.40 |
| MBT | Yaw angle [deg] | 4.46 | 3.75 | 4.28 |
| | Speed [kph] | 0.48 | 0.44 | 0.47 |

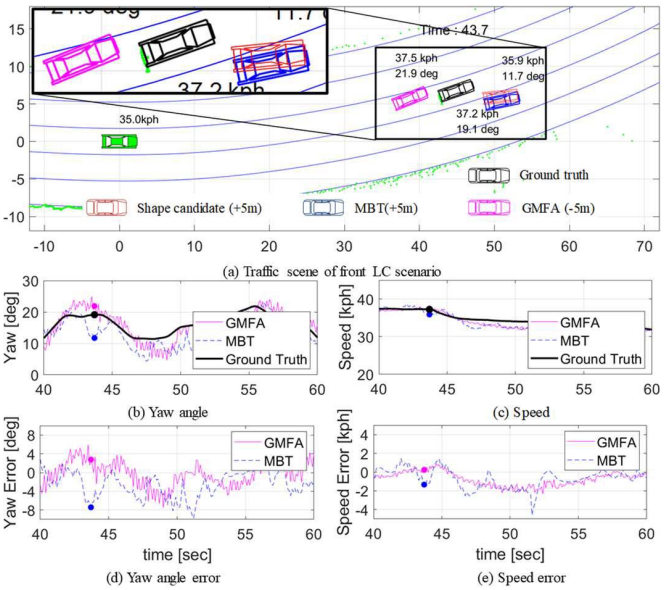



Fig. 10. Estimation error in yaw angle and speed of GMFA and MBT at front LC scenarios.

MBT, respectively. The data bias is attributed to calibration errors in the RT range and LiDAR, and asymmetry of the target position and the road shape during data acquisition. Therefore, the standard deviation indicating the distribution of errors is recommended for the validation of estimation accuracy, and the values of each approach are shown in TABLE III.

As shown in TABLE III, GMFA enhances the estimation accuracy compared with MBT under all scenarios. The estimation accuracy of the yaw angle was improved by 58% compared with MBT (Fig. 10). Fig. 10 displays the reference states (b) and the estimated states (c), the estimation error (d and e), and the traffic environment (a) during forward lane changes over time. Fig. 10-(a) represents the traffic scene, extracted shape, and estimated states at the instant represented by points in Fig. 10-(b), Fig. 10-(c), Fig. 10-(d), and Fig. 10-(e). For the visualization, the estimation results via GMFA and MBT were plotted with + 5 m and - 5 m offset, respectively, in Fig. 10-(a).

The differences in the estimated accuracy of yaw angle are shown in Fig. 10-(b) and Fig. 10-(d). The standard deviation of MBT was larger than that of the proposed approach because of the error in yaw angle greater than 6 degrees (44 s and 51 s in Fig. 10-(b) and Fig. 10-(d), respectively). The error has been attributed to the results of the shape extraction as indicated by the red vehicle in Fig. 10-(a). In this sense, the extraction of the shape using only point cloud in the current step limits

TABLE IV
OPERATION TIME OF THE PROPOSED APPROACH

| Time/ Frame | Data Parsing | SOM predict | Cluster -ing | GMFA | SOM update | Total |
|---|---|---|---|---|---|---|
| Mean [ms] | 6.3 | 7.3 | 5.6 | 30.6 | 4.3 | 54.1 |
| Max [ms] | 11.3 | 15.6 | 13.3 | 80.2 | 9.2 | 118.2 |

the accuracy of the yaw angle of the extracted shape. The extracted shape varied from the actual yaw angle by more than 10°, as shown in Fig. 10-(a). A significant deviation of shape extraction from the true value results in inaccurate estimation of MBT. However, GMFA estimates the yaw angle via the direction of cluster movement as compared with the consecutive cluster, which explains the higher accuracy as compared with MBT.

### *D. Operation Time*

All the experiments and simulation tests were conducted with an Intel Core i7-4790 4.00GHz CPU at MATLAB R2018a. As shown in TABLE IV, the operation time per frame for Nambu-Beltway and SNU campus datasets confirmed the results of detection. The first frame of each dataset was excluded from the operation time because of the need for additional time for initialization. Therefore, the operation time was analyzed for a total of 494 frames. The average and maximum values are shown. TABLE IV displays the operation time of the sub-functions in a sequential order. The scan frequency of Ibeo 2010 Lux used in this study was 12.5Hz, suggesting the feasibility of real-time application of the proposed approach for each scan without additional optimization, parallel computation, or ROI selection in the above environment.

## VI. CONCLUSION AND FUTURE WORK

In order to improve the efficiency of moving object detection and tracking in the urban environment, a novel method based on the interaction between SOM and GMFA has been developed. All points were efficiently handled in the urban environment by effectively expressing static objects including temporary stationary obstacles using SOM, which were performed without ROI selection. In addition, the SOM was constructed rapidly and accurately in the local coordinate system by eliminating the interference of the moving objects in conjunction with GMFA. The points of moving object were classified via SOM and tracked using GMFA to suppress the detection and tracking failure due to stationary objects, and worked to establish the correspondence between consecutive scans efficiently. In addition, since tracking was achieved without assuming the shape of the moving object, efficient detection and tracking of various moving objects encountered in the urban environment was successfully performed. The yaw angle and speed were estimated with a high degree of confidence via correlation of points in consecutive scans.

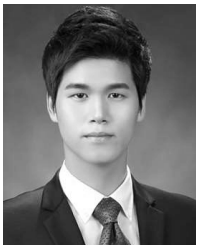
In fact, the $F_1$ score was improved by 25% and the standard deviation of the yaw angle error was reduced by 58%, as compared with the MBT using shape assumption. At the same time, since the average operation time per frame was shorter than the LiDAR scan period, the proposed approach complied with real-time requirements without additional optimization or parallel operation.

The proposed algorithm effectively detects and tracks the object with high speed ($>13.5$ kph) and a low relative speed. The accuracy for low speed pedestrians or high relative speed vehicles is low, due to confusion with static obstacle and large geometry changes. In the next step, we will focus on improving the detection and tracking performance of these objects while maintaining real-time characteristics.

## REFERENCES

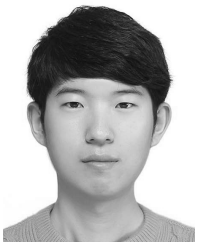
[1] *Co-operative Systems in Support of Networked Automated Driving by 2030*, AutoNet2030, Amsterdam, The Netherlands, 2014.
[2] M. C. Hutchison, J. A. Pautler, and M. A. Smith, "Traffic light signal system using radar-based target detection and tracking," Google Patents 7 821 422, Oct. 26, 2010.
[3] T. Giese, J. Klappstein, J. Dickmann, and C. Wohler, "Road course estimation using deep learning on radar data," in *Proc. 18th Int. Radar Symp. (IRS)*, Jun. 2017, pp. 1–7.
[4] A. Borcs, B. Nagy, and C. Benedek, "Instant object detection in lidar point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 992–996, Jul. 2017.
[5] V. Magnier, D. Gruyer, and J. Godelle, "Automotive LIDAR objects detection and classification algorithm using the belief theory," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 746–751.
[6] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 84–89.
[7] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understand.*, vol. 125, pp. 251–264, Aug. 2014.
[8] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 1044–1049.
[9] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.
[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[11] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4312–4319.
[12] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.
[13] N. Wojke and M. Haselich, "Moving vehicle detection and tracking in unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3082–3087.
[14] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018.
[15] Y. Ye, L. Fu, and B. Li, "Object detection and tracking using multi-layer laser for autonomous urban driving," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 259–264.
[16] 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.
[17] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
[18] 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, Apr. 2009.
[19] K. Liu, W. Wang, R. Tharmarasa, and J. Wang, "Dynamic vehicle detection with sparse point clouds based on PE-CPD," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1964–1977, May 2019.

[20] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 142–147.
[21] R. Kaestner, J. Maye, Y. Pilat, and R. Siegwart, "Generative object detection and tracking in 3D range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3075–3081.
[22] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Dynamic obstacles detection and 3D map updating," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 5694–5699.
[23] Q. Zhong, Y. Liu, X. Guo, and L. Ren, "Dynamic obstacle detection and tracking based on 3D lidar," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 22, no. 5, pp. 602–610, 2018.
[24] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 1039–1063, Jun. 2015.
[25] V. Vaquero, E. Repiso, and A. Sanfeliu, "Robust and real-time detection and tracking of moving objects with minimum 2D LiDAR information to advance autonomous cargo handling in ports," *Sensors*, vol. 19, no. 1, p. 107, 2019.
[26] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.
[27] J. van de Ven, F. Ramos, and G. D. Tipaldi, "An integrated probabilistic model for scan-matching, moving object detection and motion estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 887–894.
[28] L. Xu, C. Feng, V. R. Kamat, and C. C. Menassa, "An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments," *Autom. Construct.*, vol. 104, pp. 230–245, Aug. 2019.
[29] B. Kim, K. Yi, H.-J. Yoo, H.-J. Chong, and B. Ko, "An IMM/EKF approach for enhanced multitarget state estimation for application to integrated risk management system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 876–889, Mar. 2015.
[30] 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 2014, pp. 1836–1843.



**Hojoon Lee** received the B.S. degree in mechanical engineering from Seoul National University, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His research focuses on detection and tracking of moving objects for automated driving vehicle via sensor fusion of local sensors such as LiDAR, Radar, and vision.

**Jeongsik Yoon** received the B.S. degree in mechanical engineering from Seoul National University, South Korea, in 2017, where he is currently pursuing the M.S. degree in mechanical and aerospace engineering. His research interests are sensor fusion, estimation theory, mapping and localization, and overall perception issues such as multitarget tracking in robotics and autonomous vehicle area.



**Yonghwan Jeong** received the B.S. degree in mechanical engineering from Seoul National University, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His research interests are sensor fusion, risk assessment, automated vehicle motion planning in urban roads, automated vehicle control, and intersection assistance systems.

**Kyongsu Yi** (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, South Korea, in 1985 and 1987, respectively, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 1992. He is currently a Professor with the School of Mechanical and Aerospace Engineering, Seoul National University, South Korea. He currently serves as a member of the editorial boards of the Mechatronics and Chair of the KSME IT Fusion Technology Division. His research interests are control systems, driver assistant systems active safety systems, and automated driving of ground vehicles.