IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 21, 2019, accepted July 1, 2019, date of publication July 10, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927736

Evidence Filter of Semantic Segmented Image From Around View Monitor in Automated Parking System

CHANSOO KIM[1], (Student Member, IEEE), SUNGJIN CHO[1], (Student Member, IEEE), CHULHOON JANG[1], (Member, IEEE), MYOUNGHO SUNWOO[1], (Member, IEEE), AND KICHUN JO[2], (Member, IEEE)

[1]Department of Automotive Engineering, Hanyang University, Seoul 04763, South Korea
[2]Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, South Korea

Corresponding author: Kichun Jo (kichun.jo@gmail.com)

This work was supported in part by the BK21 Plus Program (22A20130000045) under the Ministry of Education, South Korea, in part by the Industrial Strategy Technology Development Program under Grant 10039673, Grant 10060068, and Grant 10079961, the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry, and Energy (MOTIE Korea), and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MEST) under Grant 2011-0017495.

ABSTRACT An Around View Monitor (AVM) is widely used as one of the perception sensors for automated parking systems. By applying semantic segmentation based on a deep learning approach, the AVM can detect two essential elements for automated parking systems: slot marking and obstacles. However, the perception based on the deep learning approach in the AVM has certain limitations such as occlusion of the ego-vehicle region, distortion of 3D objects, and environmental noise. We overcome the problems by proposing an evidence filter that improves the detection performance based on evidence theory and a Simultaneous Localization and Mapping (SLAM) algorithm. The proposed algorithm is composed of three parts: the semantic segmentation of the AVM image, confidence modeling based on evidence theory, and evidence SLAM. Semantic segmentation classifies the grids in the AVM image into three states: *slot marking*, *freespace*, and *obstacle*. The grids with these three states are modeled by a confidence model based on evidence theory. Finally, the states of the grids around the ego-vehicle are accumulated and estimated by the evidence SLAM. The proposed filter was evaluated by experiments in real parking-lot environments.

INDEX TERMS Filtering, around view monitor, evidence theory, semantic segmentation, SLAM.

I. INTRODUCTION

Parking lots where vehicles usually stay for long times are one of the most important places that must facilitate access after driving. However, many drivers suffer when using relatively unfamiliar driving controls in parking lots such as backward movement and driving in high curvatures. In addition, the high vehicle density boosted by the increase of vehicles and rapid urbanization has caused that 23% car accidents have occurred in the parking lots [1]. To improve driver comfort and safety, automated parking systems have been widely researched [2]–[8].

When performing automated parking, perceiving the slot markings and other obstacles near the ego-vehicle is essential. The perceived information has three key roles in automated parking systems: parking position selection, obstacle detection, and ego-vehicle pose estimation. The position where the ego-vehicle will be parked can be selected through slot markings and empty spaces that are unoccupied by other obstacles [9]. In addition, measured obstacles are used to avoid collisions in path planning [8]. Finally, measured slot markings can provide the inferences to estimate the relative pose between the selected parking position and the ego-vehicle for controlling the ego-vehicle [8].

The perception information for automated parking systems has highly intimate connections with the perception sensors installed in the ego-vehicle; the perception sensors in automated parking systems have two requirements: cost and measurement range. Since automated parking systems are applied in mass-produced products, the sensors for the

The associate editor coordinating the review of this manuscript and approving it for publication was Chenping Hou.

system must be cheap to install or must be existing sensors that are installed in conventional vehicles for other purposes. Furthermore, the sensors are required to detect close environments in all directions for automated parking systems in narrow regions.

A semantic segmentation based on a deep learning approach, which has been recently researched in an image processing area, provides the potential to detect the slot markings and obstacles solely based on AVM [8], [9]. The semantic segmentation based on the deep learning can classify an AVM image as *slot marking*, *freespace*, and *obstacle*. Despite the potential, it is difficult to use the semantic segmentation algorithm practically, since the algorithm has five inherent problems caused by the properties of the AVM and the deep learning approach, as shown in figure 1.

- **Occlusion**: Information within the ego-vehicle region cannot be detected from the AVM.
- **Misclassification**: The deep learning can provide incorrect information due to environmental noise.
- **Geometric distortion**: A geometric relationship between the cameras and the ground can be distorted by the roll and pitch motion of the ego-vehicle.
- **Calibration error**: Incorrectly calibrating the AVM can cause errors in the AVM image.
- **Stretched obstacle**: Since the AVM makes 3D obstacles to be stretched, the information behind obstacles cannot be inferred.

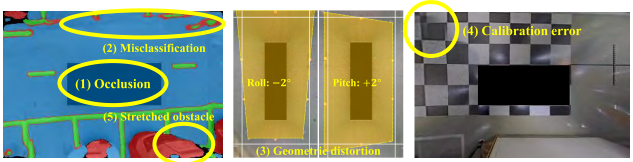


FIGURE 1. Limitations of the semantic segmentation in the AVM image: (1) occlusion, (2) misclassification, (3) geometric distortion, (4) calibration error, and (5) stretched obstacle.

In order to improve the semantic segmentation performance by overcoming five problems, we propose an evidence filter. Along with the moving of the ego-vehicle, the filter would accumulate sequential measurements from the semantic segmentation into a grid map around the vehicle. Since each grid in the grid map has information accumulated by previous and present measurements, the filter can determine the grid's precise state.

In order to apply the filter, three problems are considered. First, the reliability of the measurements is evaluated in advance to give more credit to reliable measurements in the accumulation process. The reliability is determined by five confidence models designed to reflect the inaccuracy of the five inherent problems. Second, a method to accumulate the measurements must be considered in each grid. Although probabilistic theory is generally used to accumulate measurements, this theory can only treat binary class problems which determine existence of objects [10] or slot markings [11]. The proposed filter solves the problem by applying an evidence theory to accumulate multiple class measurements such as *slot marking*, *freespace*, and *obstacle*. Finally, it is essential that the filter estimates the moving of the vehicle precisely because the filter accumulates measurements into the grid map based on the vehicle pose. To estimate the moving, we apply a Simultaneous Localization and Mapping (SLAM) algorithm based on motion information and matching information between measured slot markings.

This paper's goal is to improve the semantic segmentation performance based on the AVM by overcoming the five inherent problems. To achieve the goal, the paper has three main contributions.

- We propose the evidence filter based on Evidence theory and SLAM to improve the semantic segmentation performance.
- The evidence filter applies the confidence models designed to overcome the five inherent problems caused by the properties of the AVM and deep learning.

To describe the evidence filter, this paper is organized as follows. Section II explains works related to the paper. Section III describes the system architecture of the proposed algorithm. Sections IV, V, and VI explain semantic segmentation based on the deep learning approach, confidence modeling to accumulate the semantic information, and the evidence SLAM based on the evidence theory and the SLAM algorithm. In Section VII, experiments evaluate the proposed algorithm in real environments. Finally, the paper is concluded with Section VIII.

II. RELATED WORKS

In the automated parking system, the targets of perception are categorized into two types: parking space detection and obstacle detection. Various range-finding sensors are widely used to perform these two functions: Ultra-sonic sensors [12]–[15], Short-range radars [16]–[18], single-layer LIDARs [19]–[21], structure from motion [22]–[24], and binocular stereo [25]–[27]. Since the introduction of automated parking systems to the public as a product, ultrasonic sensors installed in both sides of the bumper of the ego-vehicle have primarily been used due to their low cost. However, this approach has three problems: the compulsion of driving, positioning errors due to inaccurate motion sensors, and failures in no obstacle environments. Measuring empty spaces for parking space selection requires that the ego-vehicle is driven in front of empty spaces because the sensors have narrow measurements. Furthermore, since the measured empty spaces are registered based on the inaccurate vehicle motion sensors, the positioning errors can occur based on drift errors. Finally, ultra-sonic sensor-based perceptions cannot cope with the situation in no-obstacle environments. Although approaches based on short-range radar [16]–[18] and single-layer LIDAR [19]–[21] excluded the compulsion of driving problems, it is difficult for such sensors to be used in mass-produced products due to their high price. Since the range-finding sensors based on imaging sensors such as structure from motion [22]–[24] and the binocular stereo [25]–[27] are relatively cheap, imaging sensors are researched. However, feature point matching, which determines the performance of ranging sensors, is greatly affected by environmental noise such as day and night, underground, and rain. In addition, the generally used hand-craft features do not have any semantic information for geometric points such as obstacle and slot markings.

To cope with the environments without any obstacles, the parking slot marking detection has been researched. The slot marking detection can be classified into two categories: hand craft feature-based and artificial intelligence-based. The methods to extract the slot marking features from the images have been widely researched such as local threshold [28], PHD filter [29], steerable filter [30], canny edge method [31], randomized Hough transform [32], and probabilistic Hough transform [33]. However, the feature extraction methods suffered from false-positive features affected by the 3D objects and environmental noises. In order to remove the false-positive features, Lee et al. [11] proposes an occupancy filter based on probabilistic theory. Although the method can remove the false-positive features, the method can detect only slot markings other than the 3D obstacles. In addition, since the method uses only motion information to estimate the trajectory of the vehicle, the images are accumulated inaccurately. In addition to the hand-craft features, the slot marking detection based on the artificial intelligence started from Xu et al. [34]. They recognized colored slot markings through the neural network based color segmentation [34]. Recently, many approaches based on deep learning have been researched since the success of the AlexNet [35]. Valipour et al. determined the occupancies of the parking slots based on deep learning using the surveillance camera [36]. However, the system is not suitable for an automated parking system because the system requires infrastructures such as the surveillance cameras and the communication system with the vehicle. For the automated parking system, Zhang et al. proposed the deep convolutional neural network based parking slot detection algorithm using the AVM system [37], [38]. However, since the method did not consider the other vehicles, the parking system cannot be performed in the environment only with the other vehicles.

To detect both other obstacles and parking slot markings, Li et al. used a mapping vehicle with a 3D LIDAR sensor [39]. However, since the 3D LIDAR sensor is a high-cost sensor, it is not suitable for mass production of the intelligent vehicles. Different from this approach, Suhr et al. proposed the sensor fusion algorithm with the AVM system and ultrasonic sensors [40]. Since the approach detects the slot markings based on the hand-craft features, it can suffer from false-positive features. Also, it is difficult to fit the coincidence of calibration parameters of the multiple sensors.

III. SYSTEM ARCHITECTURE

As shown in figure 2, the evidence filter consists of three parts to measure the slot markings and obstacles precisely: a) semantic segmentation, b) confidence modeling, c) evidence SLAM. a) The semantic segmentation based on the deep learning approach [9] splits the AVM image measured from the intelligent vehicle into three semantic probabilistic channels: *slot marking* $p(S)$, *freespace* $p(F)$, and *obstacle* $p(O)$. Each pixel in the semantic segmented AVM image has three probabilities whose summation is 1. In the process, the ego-vehicle region is excluded, because the region cannot be measured from the AVM image.

b) The confidence modeling based on the evidence theory is performed in order to reflect the reliability of the semantic segmented AVM image. The modeling converts the semantic segmented AVM image into the local Evidence Grid Map (EGM), which consists of two-dimensional discrete cells around the ego-vehicle. To overcome the five problems as shown in figure 1, the local EGM considers the five confidence models with the properties of the AVM image and the deep learning approach: (1) ego-vehicle model, (2) misclassification model, (3) geometric distortion model, (4) calibration error model, and (5) stretched obstacle model. First, the ego-vehicle model classifies the region in the ego-vehicle as unknown states. Second, the misclassification model applies the classification accuracy of the deep learning approach. Third, the geometric distortion model reflects the error of the image deformed from inaccurate geometric relationship between the cameras and the ground. Fourth, the calibration error model reflects the rectified errors of the AVM cameras. Finally, the stretch of the 3D obstacle from the AVM image is considered in the confidence model. As shown in figure 2-(b), the pixel in the local EGM modeled by the five confidence models includes four channels such as the red, green, black, and blue colors. The four channels represent *slot marking*, *freespace*, *obstacle*, and *unknown*, respectively. The local EGM is used as the measurement of the evidence SLAM based filtering algorithm.

c) The evidence SLAM-based filtering algorithm is applied to estimate the precise states around the ego-vehicle by accumulating the local EGM to a global EGM, which is represented in the global coordinate. Since the relative positioning in the local coordinate is important to accumulate the local information, the evidence SLAM uses the GraphSLAM algorithm [41]. The GraphSLAM algorithm based on the motion and matching information with the slot markings is effective to estimate the relative positioning, because the slot markings on the ground do not have the deformation property differently with the obstacles. Based on the relative positioning, the evidence SLAM merges the local EGM to the global EGM. The states in the global EGM can be more precisely estimated by accumulating the past-and-present measurements than by the present measurement only. Finally, the semantic states around the ego-vehicle are extracted from the global EGM and determined to the deterministic filtered states.

IV. SEMANTIC SEGMENTATION

Many researchers have worldwide researched line extraction algorithms to acquire the slot markings for advanced driver

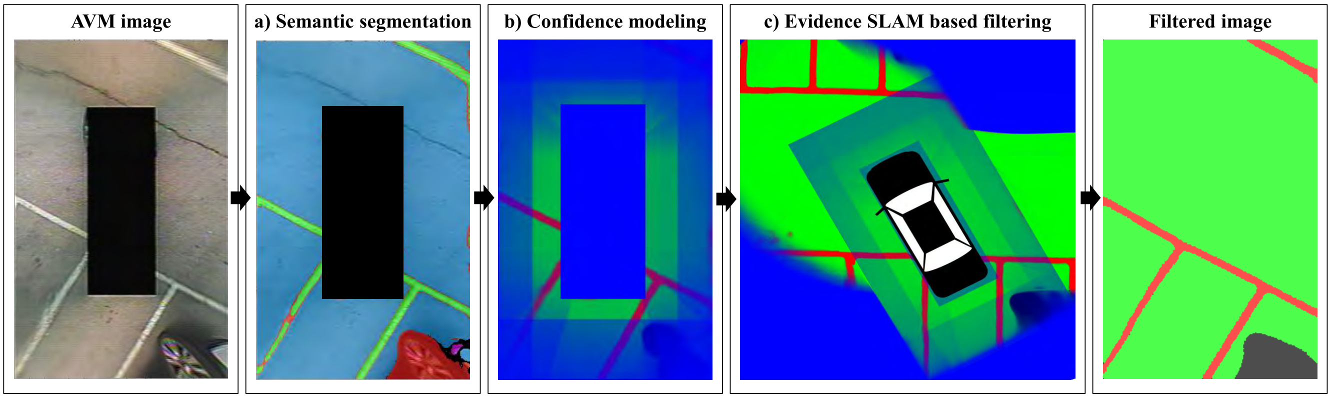


FIGURE 2. System architecture of the evidence filter.

assistance systems. However, the approaches have some limitations to an automated parking system, because other environmental information such as *freespace* and *obstacle* is not recognized. The deep learning approaches widely researched in the computer vision area allow detection of all information from the image. In order to detect the *slot marking*, *freespace*, and *obstacle*, a fully convolutional neural network(FCN)-based semantic segmentation algorithm is applied [42]. It has been researched in several papers that it is effective for the deep learning approach to classify the AVM image into semantic information in the automated parking system [8], [9].

A semantic segmentation model consists of two parts: the encoder and decoder. The encoder plays a role in high-level feature extraction from the AVM image. Using the high-level features, the decoder estimates pixel-wise semantic information by using an up-sampling method. The architecture of the semantic segmentation is based on the paper proposed by Long et al. [42]. The proposed structure has three benefits: usage of predefined parameters, end-to-end learning, and scalability. In order to extract high-level features, an effectiveness-proven model (VGG 16) is used as predefined parameters for the encoder. Finally, the structure has no effect from the image size. As a result of the learning algorithm, the semantic inference from the AVM image gives the pixel-wise semantic information, which consists of three states: *slot marking*, *freespace*, and *obstacle*. Each pixel except the ego-vehicle region has three probabilities for the three states, the summation of which must be one. The ego-vehicle region is not treated in the semantic segmentation process.

To train the learning model for semantic segmentation, 6, 763 images with manual annotations are constructed manually [43]. The dataset has many images under various environments, such as rainy day, sunny day, and indoor environments. In order to improve the performance of the deep learning approach, an augmentation of the data set is required. For the augmentation, random cropping, flipping, and rotating are applied. The result of the semantic segmentation is shown in figure 3.

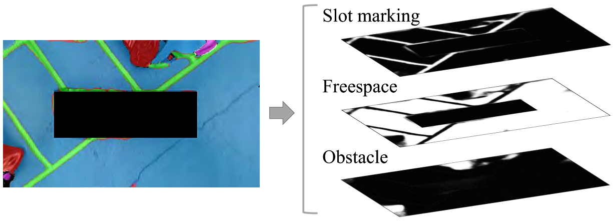


FIGURE 3. Result of semantic segmentation from the AVM image.

V. CONFIDENCE MODELING

A. LOCAL EGM BASED ON EVIDENCE THEORY

As shown in figure 2, the semantic segmented AVM image is accumulated into the global semantic space for acquiring the semantically estimated information around the vehicle. Unfortunately, the semantic segmented AVM image cannot reflect the real semantic information around the ego-vehicle precisely due to the several properties of the AVM image and the deep learning approach as shown in figure 1. First, the region in the ego-vehicle cannot be measured in the AVM image. Next, inaccurate geometric distortion between the cameras and the ground can distort the AVM image, because it is assumed that the AVM image is rectified in the parallel condition between the ego-vehicle and the ground. In addition, the inaccurate calibrations of the camera sensors can distort the AVM image. Furthermore, 3D objects with heights are stretched in the AVM image. Finally, the semantic segmentation based on the deep learning can misclassify the semantic information from the AVM image. These problems cause the inaccuracy in each pixel of the semantic AVM image. Since the inaccuracy affects the performance of accumulation into the global semantic space, the inaccuracy must be considered in the accumulation process. In order to reflect the inaccuracy of the information, the semantic AVM image is converted to the local EGM based on the evidence theory [44].

The local EGM models environments around the ego-vehicle into two-dimensional discrete cells. Each cell of the local EGM is inherently affiliated to one of three states: *slot marking* (S), *freespace* (F), and *obstacle* (O). The three states are used as a frame of discernment $\Omega = \{S, F, O\}$. The states of the local EGM can be extended to the power set $2^{\Omega} = \{\emptyset, S, F, O, S{\cup}F, S{\cup}O, F{\cup}O, S{\cup}F{\cup}O\}$, which is the set of all subsets of the $\Omega = \{S, F, O\}$. To quantify the evidence of each element of the power set, a mass function, m, is applied. The $m(\emptyset)$ denotes the *conflict* mass function, which represents a situation where the different measurements are in conflict. The $m(S)$, $m(F)$, and $m(O)$ demote the mass functions that the cell is *slot marking*, *freespace*, and *obstacle*, respectively. The $m(S \cup F)$ denotes the mass function that the cell is either *slot marking* or *freespace*. Similarly, the $m(S \cup O)$ and $m(F \cup O)$ denote the mass functions that the cell is either *slot marking* or *obstacle*, and either *freespace* or *obstacle*, respectively. Finally, the $m(S \cup F \cup O\}$ means the evidence of the *unknown* that there is no inference for the cell, which is denoted as $m(\Omega)$. The summation of all mass functions in the power set must be one based on the evidence theory.

For conversion of the local EGM from the semantic segmented AVM image, the eight mass functions $m_{L(i,j)}$ located at index $L(i,j)$ in the local EGM are determined by probabilities $p_{L(i,j)}$ of the semantic segmented AVM image. As shown in equation (1), the probabilities of *slot marking* $p(S)$, *freespace* $p(F)$, and *obstacle* $p(O)$ in the pixel of the semantic segmented AVM image are propagated to the mass functions of *slot marking* $m(R)$, *freespace* $m(F)$, and *obstacle* $m(O)$, respectively. Since some mass functions, $m_{L(i,j)}(\emptyset)$, $m_{L(i,j)}(S{\cup}F)$, $m_{L(i,j)}(S{\cup}O)$, $m_{L(i,j)}(F{\cup}O)$, are zeros, only four mass functions, $m_{L(i,j)}(S)$, $m_{L(i,j)}(F)$, $m_{L(i,j)}(O)$, $m_{L(i,j)}(\Omega)$, are explicitly managed. In order to consider the inaccuracy of the semantic AVM image, the confidence $\lambda_{L(i,j)}$ is applied at each cell. While low confidence entails a high *unknown* state, high confidence entails a low *unknown* state. To overcome five problems as shown in the figure 1, the confidence of each pixel is determined by the five models explained in the section V-B. As a result, the semantic segmentation of deep learning in the figure 2-(a) is converted to the local EGM in the figure 2-(b). The red, green, black, and blue colors in the EGM mean *slot marking*, *freespace*, *obstacle*, and *unknown*, respectively.

$$\begin{aligned} m_{L(i,j)}(\emptyset) &= 0 \\ m_{L(i,j)}(S) &= \lambda_{L(i,j)} \cdot p_{L(i,j)}(S) \\ m_{L(i,j)}(F) &= \lambda_{L(i,j)} \cdot p_{L(i,j)}(F) \\ m_{L(i,j)}(O) &= \lambda_{L(i,j)} \cdot p_{L(i,j)}(O) \\ m_{L(i,j)}(S \cup F) &= m_{L(i,j)}(S \cup O) = m_{L(i,j)}(F \cup O) = 0 \\ m_{L(i,j)}(\Omega) &= 1 - \lambda_{L(i,j)} \end{aligned} \tag{1}$$

B. CONFIDENCE MODELING FOR LOCAL EGM

In order to reflect inaccuracy of the measurements, the confidence $\lambda_{L(i,j)}$ is applied in equation (1). The confidence $\lambda_{L(i,j)}$ consists of five confidence models: ego-vehicle model λ_e, classification rate model λ_r, unexpected motion model λ_m, calibration error model λ_c, and obstacle model λ_o.

1) EGO-VEHICLE MODEL

The ego-vehicle model considers the no inference region from the semantic segmented AVM image. The confidence λ_e located in the ego-vehicle region R_{ego} is set to 0. In contrast with the ego-vehicle region, the confidence λ_e is set to 1 in the other region.

$$\lambda_{L(i,j),e} = \begin{cases} 0, & \text{if } L(i,j) \in R_{ego} \\ 1, & \text{if } L(i,j) \notin R_{ego} \end{cases} \tag{2}$$

2) MISCLASSIFICATION MODEL

In the semantic segmentation algorithm, the class of each pixel is not sometimes estimated precisely. In order to reflect the false classification, the classification confidence model reflects the classification rate of the semantic segmentation algorithm. The classification rate is the constant value determined by the performance of the pre-trained learning model of the semantic segmentation. As shown in equation (3), the confidence λ_m is the ratio of the correctly estimated cases to all cases N.

$$\lambda_m = \frac{\sum^{N} f(x)}{N}, \quad f(x) = \begin{cases} 0, & \text{if } x \neq \hat{x} \\ 1, & \text{if } x = \hat{x} \end{cases} \tag{3}$$

where the x means a true class in a cell and the $\hat{x}$ means the estimated class in the cell. The confidence λ_m is acquired in Section VII. The confidence λ_m is the same in all regions.

3) GEOMETRIC DISTORTION MODEL

The inaccurate geometric distortion by the pitch and roll of the vehicle in real-driving conditions can cause deformation of the AVM image, which degrades the performance of the accumulation. To relieve the effect of the unexpected vehicle motion, the confidence λ_g of each pixel in the AVM image can be applied. As represented in figure 4-(a) and (b), the Gaussian distribution at each pixel is derived by applying the real vehicle motion data to the geometry relation between the ground information and the AVM image [45]. In other words, the mean μ_g and the covariance Σ_g at every pixel in the image are acquired based on the real-driving vehicle motion data. Accordingly, each Gaussian distribution with the mean μ_g and the covariance Σ_g represents the model of the true position in the real driving. The confidence λ_g, which means the probability that the true position is located in the

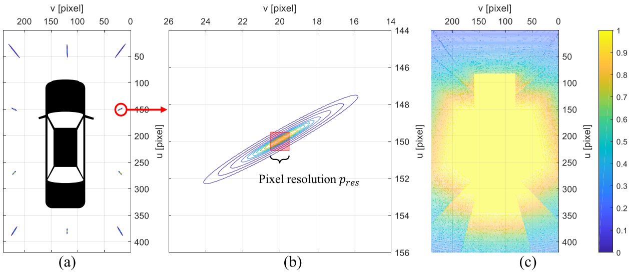


FIGURE 4. **Confidence model by geometric distortion. (a) 10 samples for displaying the Gaussian distributions. (b) Integration of the Gaussian distribution within a pixel of the mean. (c) Confidence map considering geometric distortion model.**

pixel of the mean, is acquired by the integral of the distribution defined by the pixel resolution p_{res}, as represented in equation (4).

$$\lambda_{\mu_g,g} = \frac{1}{|det(\Sigma_g)|} \cdot \iint_{\mu_g \pm p_{res}/2} exp(-\frac{1}{2}(X-\mu_g)^T \Sigma_g^{-1}(X-\mu_g))dydx, \quad (4)$$

where $X(x, y)$ means true position including x to the longitudinal direction and y to the lateral direction. Since the λ_g is derived by the integral of the Gaussian distribution, the λ_g is represented from 0 to 1. Finally, a confidence map considering the non-horizontal model in the AVM image is derived as shown in figure 4-(c).

4) CALIBRATION ERROR MODEL

The error of the calibration from the cameras to the AVM image can cause the deformation of the AVM image. In order to relieve the effect of the calibration error, a confidence model based on the calibration error is proposed. In the AVM image, the position $\hat{X}$ computed by the image geometry is known based on the multiplication between the location of the cell and the pixel resolution. On the other hand, to calculate the image position X measured in the AVM image, the AVM image is acquired on the chessboard with 1 m grids as shown in figure 5-(a). Since the image positions at the vertices of the grids are measured from the AVM image, other image positions are acquired through the interpolation. The position $\hat{X}$ computed by the image geometry and the image position X are represented as a red point and a blue point in figure 5-(b). The difference between two points is modeled as the Gaussian distribution for calibration error model. In every pixel in the AVM image, every Gaussian distribution with the mean X and covariance Σ_c is derived as shown in figure 5-(b), assuming that the covariance Σ_c is predefined. As shown in equation (5), the calibration confidence is derived by computing the value about the position of the computed position $\hat{X}$ in the Gaussian distribution of the image position X. Accordingly, the high calibration error causes the low confidence. As a result, the confidence map from the calibration error model is represented in figure 5-(c).

$$\lambda_c = exp(-\frac{1}{2}(X-\hat{X})^T \Sigma_c^{-1}(X-\hat{X})) \quad (5)$$

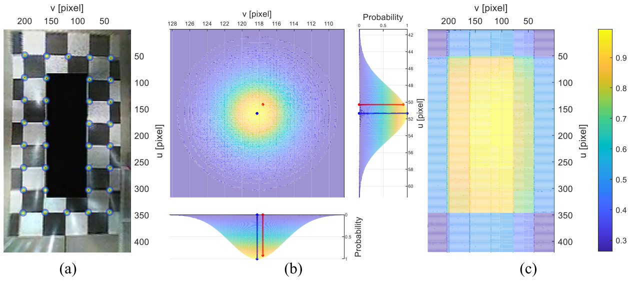


FIGURE 5. Confidence model by calibration error. (a) AVM image on the chessboard. (b) Gaussian distribution of pixel error at the corner points of the chessboard. (c) Confidence map considering calibration error model.

5) STRETCHED OBSTACLE MODEL

The property that the AVM models the ground image from multiple cameras causes the stretch problem of the 3D obstacles with height information such as other vehicles, curbs, and poles. Each pixel in the AVM image stretched from the height-ed obstacles has high obstacle probability $p_{L(i,j)}(O)$ from the semantic segmentation algorithm. However, the state of the region behind the obstacles cannot be known exactly. In order to solve this problem, the confidence λ_s is modeled. In figure 6-(a), the pixel information can be measured along to the ray started at the camera. When the ray crosses the obstacle region, the confidence λ_s is considered. As it is assumed that the object has the rectangle shape, there are many candidates to represent the region from the x_{start} point to the x_{end} point as the obstacle states, as shown in figure 6-(b). Since the probability of the width of the rectangle is equal as shown in figure 6-(c), the existence probability of each rectangle is represented as $\frac{1}{N}$. Therefore, the obstacle probability of each state is derived by the integral of the existence probability as represented in figure 6-(d), which is computed by equation (6). Finally, the confidence map considering the stretched obstacle model is represented in figure 6-(e).

$$\lambda_s = \int_x^{x_{end}} \frac{1}{N} dx, \quad x_{start} \leq x \leq x_{end} \quad (6)$$

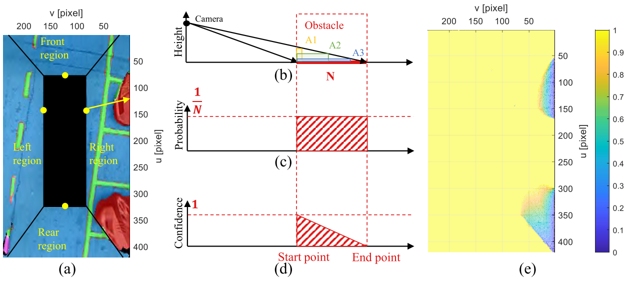


FIGURE 6. Confidence model by stretched obstacles. (a) Semantic AVM image with stretched obstacles. (b,c,d) Confidence modeling for stretched obstacles. (c) Confidence map considering stretched obstacles.

6) INTEGRATED CONFIDENCE MAP BASED ON FIVE MODELS

A confidence map represented by the confidences of grids in the local EGM is derived by equation (7). The five confidences λ_e, λ_m, λ_g, λ_c, and λ_s are multiplied. The result of the product has the value between 0 and 1 because five confidences do not exceed the 1. Therefore, the semantic segmented AVM image is converted to the local EGM based on the confidence map shown in figure 7.

$$\lambda = \lambda_e \cdot \lambda_m \cdot \lambda_n \cdot \lambda_c \cdot \lambda_d \quad (7)$$

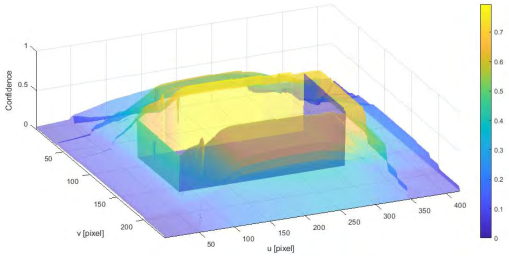
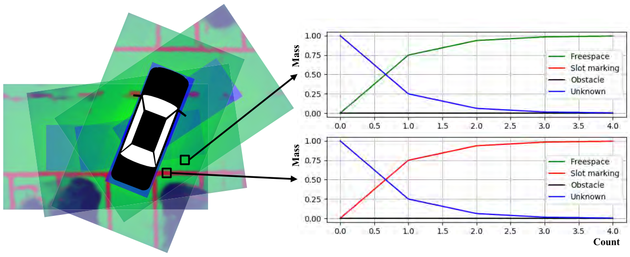


FIGURE 7. Integrated confidence map.

VI. EVIDENCE SLAM BASED FILTERING ALGORITHM

The local EGM modeled by the confidence map in figure 7 is used in the evidence SLAM for filtering of semantic information. In order to estimate the states around the ego-vehicle, the local EGM is incrementally accumulated to the global EGM as shown in figure 8-(a). In other words, the multiple states of each cell in the local EGM can be accumulated into the multiple states of each cell in the global EGM. The global EGM, represented by the cells of Cartesian coordinate, has an origin which is predefined at the starting point of the algorithm. The cells of the global EGM far from the ego-vehicle are thrown out to process the algorithm in real-time by reducing the processing time. Accordingly, the relative positioning between the present pose and the previous poses is an important factor for successful updating the global EGM. After updating the global EGM, the cells of the global EGM around the ego-vehicle are extracted. Finally, the estimated states around the ego-vehicle can be acquired. In this process, there are three problems to be solved: how to merge the four states, how to know the precise relative positioning of the ego-vehicle, and how to extract the estimated information. The solutions are explained in following three sections.



(a) (b)

FIGURE 8. Merging of global EGM based on the local EGM. (a) Merging of several local EGMs. (b) Mass of each pixel in the global EGM.

A. MERGING SECTION

First of all, the mass functions of all cells in the global EGM $m_{G(i,j)}$ are initialized by equation (8), which means no prior information to merge the local EGM as shown in figure 8-(a).

$$\begin{aligned} m_{G(i,j)}(\emptyset) &= 0 \\ m_{G(i,j)}(S) &= m_{G(i,j)}(F) = m_{G(i,j)}(O) = 0 \\ m_{G(i,j)}(S \cup F) &= m_{G(i,j)}(S \cup O) = m_{G(i,j)}(F \cup O) = 0 \\ m_{G(i,j)}(\Omega) &= 1 \end{aligned} \tag{8}$$

When the relative pose from the global coordinate is estimated precisely, the cells of the local EGM $m_{L(i,j),t}$ at time t can update the corresponding cells of the global EGM $m_{G(i,j),t-1}$ at time $t-1$ to the $m_{G(i,j),t}$. In order to update the multiple mass functions from the local EGM $m_{L(i,j),t}$ to the global EGM $m_{L(i,j),t-1}$, the Demster-Shafer merging rule based on the evidence theory is applied (9). As a result, the mass functions of *slot marking*, *freespace*, *obstacle*, and *unknown* can be updated in each pixel as shown in figure 8-(b).

$$\begin{aligned} m_{G(i,j),t} &= m_{G(i,j),t-1} \oplus m_{L(i,j),t} \\ \forall A \subseteq \Omega, m_{1\cap 2}(A) &= \sum_{B \cap C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(B) \\ m_{1\oplus 2}(A) &= \frac{m_{1\cap 2}(A)}{1 - m_{1\cap 2}(\phi)}, \quad \forall A \subseteq \Omega,\ A \neq \phi \\ m_{1\oplus 2}(\phi) &= 0 \end{aligned} \tag{9}$$

Similar with the local EGM, the global EGM has explicitly only four mass functions, $m_{G(i,j)}(S)$, $m_{G(i,j)}(F)$, $m_{G(i,j)}(O)$, and $m_{G(i,j)}(\Omega)$, because the merging rule (9) does not generate other mass functions such as $m_{G(i,j)}(\emptyset)$, $m_{G(i,j)}(S \cup F)$, $m_{G(i,j)}(S \cup O)$, and $m_{G(i,j)}(F \cup O)$.

B. POSITIONING SECTION

The precise positioning is very important to accumulate the local EGM to global EGM precisely. For the positioning, the GraphSLAM algorithm [41] is used. The algorithm consists of four parts: node, edge, solver, and node management. The node represents the target states to be estimated, measurements, and global EGM as the circle in figure 9. The edge, represented as the arrow connecting nodes in figure 9, means the relative constraint between nodes. The solver estimates the target states by optimizing the relationships based on the nodes and the edges. Finally, the node management manages addition and delete of the nodes to perform the GraphSLAM algorithm in real-time.

1) NODE

The node consists of four parts: vehicle pose x_t, motion information u_t, local EGM $m_{L,t}$, and global EGM m_G. The pose of the ego-vehicle is represented as two-dimensional Special Euclidean groups ($SE(2)$). The $SE(2)$ takes the relative distance and relative heading angle from the fixed origin of the global EGM. The motion information u_t consists of the velocity and the yaw rate at time t. The local EGM and the global EGM are each represented as two-dimensional grid spaces.

The optimization for the GraphSLAM algorithm requires that the estimation target state, the vehicle pose x_t, is initialized. The initialized pose which is approximated near the real pose can improve the performance of the optimization. Accordingly, the vehicle pose is initialized by the vehicle motion information and the vehicle motion model [46].

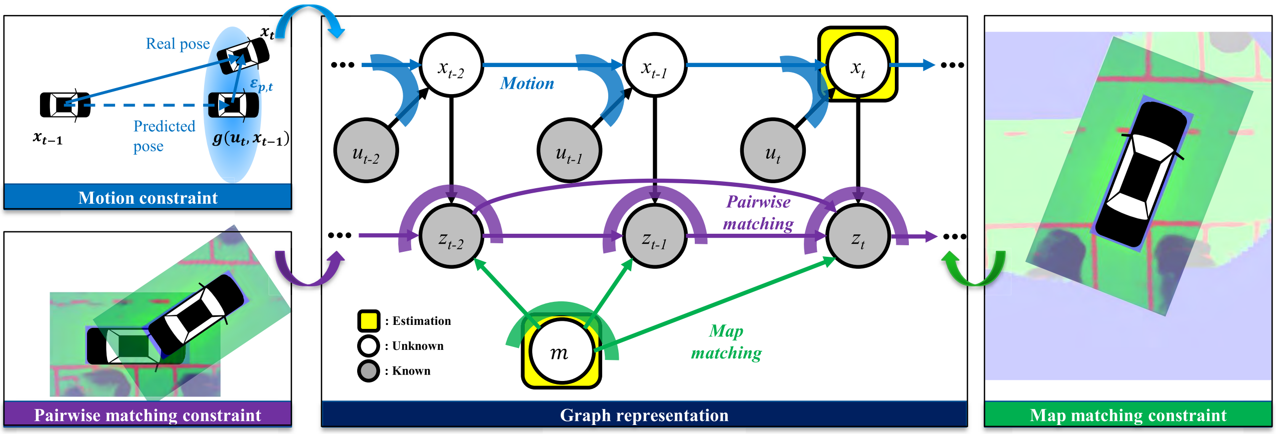


FIGURE 9. Graph representation for evidence SLAM.

2) EDGE

The edge refers to the constraint between connected nodes. The edges consists of three parts: motion constraint, pairwise matching constraint, and map matching constraint.

The motion constraint is derived by the transition model $p(x_t|x_{t-1}, u_t)$. The next pose x_t at time t is predicted from the previous pose x_{t-1} at time $t-1$ based on the motion information u_t at time t.

$$x_t = g(x_{t-1}, u_t) + \epsilon_{p,t} \tag{10}$$

$$F_{p,t} = \epsilon_{p,t}^T P_{p,t}^{-1} \epsilon_{p,t} \tag{11}$$

The term $g(x_{t-1}, u_t)$ in equation (10) predicts the next vehicle pose based on the motion model. However, the motion-based prediction has some error $\epsilon_{p,t}$ from the real pose x_t. Assuming that the error, $\epsilon_{p,t}$, is modeled as a Gaussian noise model with zero mean and covariance $P_{p,t}$, the motion constraint is represented as negative likelihood $F_{p,t}$ as shown in equation (11). Accordingly, the vehicle pose can be estimated precisely by minimizing the motion constraint $F_{p,t}$.

The pairwise-matching constraint connects two local EGMs $m_{L,t-k}$ and $m_{L,t}$. The matching between these EGMs can cause the effect similar with the motion constraint.

$$x_t = h_{pm}(x_{t-k}, m_{L,t-k}, m_{L,t}) + \epsilon_{m,t-k,t} \tag{12}$$

$$F_{pm,t-k,t} = \epsilon_{pm,t-k,t}^T P_{pm,t-k,t}^{-1} \epsilon_{pm,t-k,t} \tag{13}$$

The next vehicle pose x_t can be computed by the matching model $h_{pm}(x_{t-k}, m_{L,t-k}, m_{L,t})$ based on the two local EGMs and other pose, x_{t-k}, as shown in equation (12). For the matching process, the cells with dominant *slot marking* are extracted as points, because the *slot marking* has the properties of invariability. The extracted slot marking points from two local EGMs are matched based on the normal distributions transform (NDT) matching algorithm [47]. At that time, in order to reflect the inaccuracy of the *slot marking* in the local EGM, the modified NDT which can consider the weights of the points is applied based on the confidence of the local EGM. This algorithm can calculate the transformation matrix between two selected poses x_{t-k} and x_t. Though the process for precise matching, there may be some error $\epsilon_{pm,t-k,t}$ by inaccurate calibration and matching performance. Assuming that the error $\epsilon_{pm,t-k,t}$ is modeled as a Gaussian noise model with zero mean and covariance $P_{pm,t-k,t}$, the pairwise matching constraint can be represented as negative likelihood $F_{pm,t-k,t}$, as shown in equation (13).

A map-matching model matching between the accumulated global EGM $m_{G,t}$, and the local EGM $m_{L,t}$ generates the map-matching constraint.

$$x_t = h_{mm}(m_{G,t}, m_{L,t}) + \epsilon_{mm,t} \tag{14}$$

$$F_{mm,t} = \epsilon_{mm,t}^T P_{mm,t}^{-1} \epsilon_{mm,t} \tag{15}$$

The vehicle pose x_t can be derived by the map-matching model (14). Similarly with the pairwise-matching process, the map-matching also uses the weighted NDT algorithm based on the slot-marking points. The weights of the slot marking points can be extracted by the confidences in the local EGM and the global EGM. The consideration of the map-matching model makes the global EGM consistent. Like the pairwise-matching model, there may be an error $\epsilon_{mm,t}$ in map-matching estimation by inaccurate calibration and matching performance. Assuming that the error $\epsilon_{mm,t}$ is modeled as a Gaussian noise mode with zero mean and covariance $P_{mm,t}$, the negative log likelihood $F_{mm,t}$ can be derived in equation (15).

3) SOLVER

An optimization of the GraphSLAM can be performed by the minimization of a cost function derived from the summation of the negative log likelihoods in three equations ((11), (13), and (15)) as shown in equation (16).

$$J = \sum_t \epsilon_{p,t}^T P_{p,t}^{-1} \epsilon_{p,t} + \sum_{t,k} \epsilon_{pm,t-k,t}^T P_{pm,t-k,t}^{-1} \epsilon_{pm,t-k,t} + \sum_t \epsilon_{mm,t}^T P_{mm,t}^{-1} \epsilon_{mm,t} \tag{16}$$

In the cost function (16), the local EGM m_L is derived by the semantic segmented AVM image. In addition, the global EGM m_G is derived by the accumulation of the local EGM m_L. The unknown variables x_t are estimated by minimizing the cost function (16).

4) NODE MANAGEMENT

Whenever time is passing, to add the node into the graph causes too many loop closure edges and nodes in the graph. The big graph makes it difficult for the GraphSLAM algorithm to be processed in real-time. For real-time processing of the filtering algorithm, the node size in the graph is limited along to the time window scheme as shown in figure 10. The principle of the time window scheme follows the first-input and first-output (FIFO) rule. When a new node is generated and entered into the time window buffer, the latest node is omitted from the buffer. The edges connected from the latest node are also removed. Along with the limited node size, the GraphSLAM based filtering algorithm can be processed in real-time.

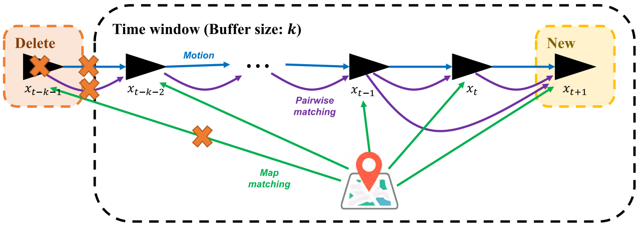


FIGURE 10. Time windows scheme for real-time processing.

C. EXTRACTING SECTION

After the GraphSLAM algorithm, the merged global EGM consists of the cells with evidence masses for *slot marking*, *freespace*, *obstacle*, and *unknown*. This global EGM is cropped with AVM image size whose center is calculated from the optimized vehicle pose. In order to determine the class of each cell in the cropped EGM, an extraction process is applied. In this extraction process, if the summation of the three evidence masses for *slot marking*, *freespace*, and *obstacle* is higher than the confidence value from the confidence model of a corresponding cell, the class with the highest evidence mass is selected for the cell's class. Otherwise, if the summation is less than the confidence value, then the cell cannot be classified by one specific class. For this cell, the class of the cell is determined by *unknown*.

VII. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENTS

A test site for the experiments is a parking lot in Korea, as shown in figure 11. A test vehicle acquired five scenarios data for evaluation in the parking lot. The test vehicle was equipped with in-vehicle motion sensors (a wheel speed sensor and a yaw rate sensor), a commercial AVM system with four cameras, two LIDARs with 16 layers (Velodyne VLP-16), and a high-precision GNSS/INS (OXTS RT3002).

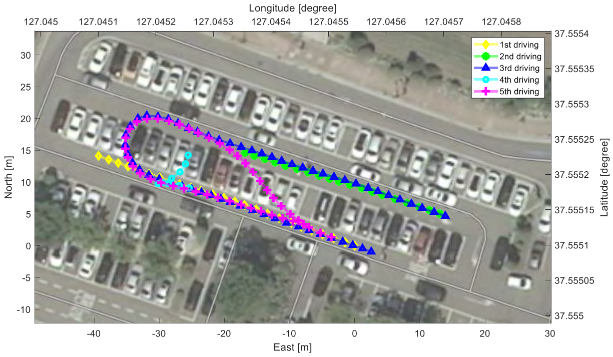


FIGURE 11. Test site for experiments: Parking lot in Korea.

The in-vehicle motion information is obtained from the Controller Area Network (CAN). The AVM generates the reconstructed image around the ego-vehicle by stitching four images measured from the four cameras. The size of the AVM image is 240 × 420 pixels whose resolution is 0.024 *m*. The LIDARs can measure the surrounding environment by acquiring reflected point clouds at 10 FPS in horizontal 360°. The high-precision GNSS/INS have 0.01 *m* position accuracy and 0.1° heading angle accuracy. As represented in figure 11, the test vehicle was driven at a similar speed in the parking lot for five test scenarios. The test scenarios include two straight-forward driving scenarios (1st and 2nd), one curved scenario (3rd), one parking scenario (4th), and one loop closing driving (5th).

The semantic information from the proposed algorithm is evaluated by comparing with the ground truth semantic information in the test site. In order to extract the ground truth semantic information, two LIDARs and the high-precision GNSS/INS are used. By accumulating the point cloud from the two LIDARs along to the trajectory of the vehicle measured by the high-precision GNSS/INS, the point cloud map is generated as shown in figure 12-(a). Since each point in the point cloud map has each intensity, *slot marking* and *freespace* are distinguished from the points located at the ground precisely. In addition, through the heights of the points, the obstacle information is extracted delicately. A professional refines the automatically-extracted semantic map manually to improve the labeling result. Therefore, the ground truth semantic map with the four channels, *slot marking*, *freespace*, *obstacle*, and *unknown*, is constructed as represented in figure 12-(b). A ground truth semantic image can be exactly acquired from the ground truth semantic map

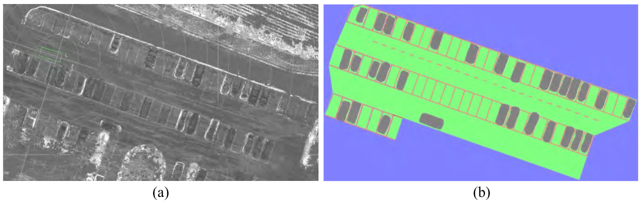


FIGURE 12. (a) Precise point cloud map with intensities. (b) Semantic information for the test site.

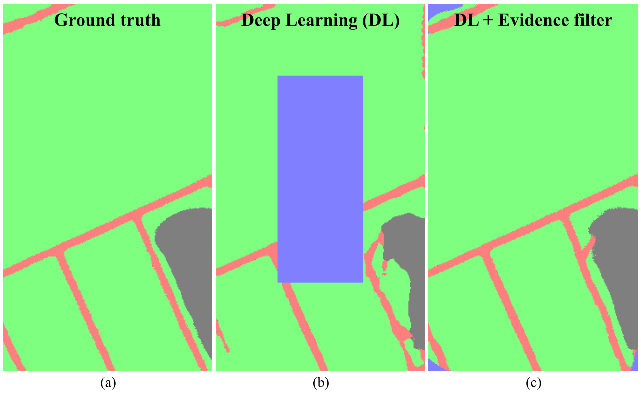


FIGURE 13. Sample data for constructing confusion matrix: (a) a ground truth semantic image, (b) a semantic image based on the deep learning, and (c) a filtered semantic image based on the proposed algorithm.

as shown in figure 13-(a), because the precise position of the vehicle is measured from the GNSS/INS. The extracted ground truth semantic image is used as evaluation data set.

B. COMPARISON WITH DEEP LEARNING

The three images in figure 13 show the semantic information from the ground truth, the deep learning, and the evidence filter, respectively. Since the images represent semantic information in the same region, the evaluation of two algorithms such as deep learning and the evidence filter can be performed through the pixel-wise comparison-based confusion matrix.

In each test scenario, sets of the three images including the ground truth image, the deep learning image, and the evidence filtered image are extracted at vehicle poses along the trajectory. The sets of images are sampled with a 0.5 *m* interval based on the distance. The results of the confusion matrices based on the sampled sets of images are shown in figure 14. The first column in figure 14 refers to the division for ground truth information in test driving scenarios. The second and third columns in figure 14 mean confusion matrices of the deep learning images and the semantic filtered images, respectively.

The scenarios, including straightforward driving, curved driving, parking, and loop closing, have their own specifications. In the straightforward driving scenarios, the ratio of *slot marking* and *obstacle* is less than the other driving scenarios', because the vehicles in other driving scenarios passed the *slot marking* as shown in figure 11. In the two straightforward driving scenarios, accuracy of *slot marking* in the evidence filter increase by 4.6% and 9.3% from 84.0% and 81.0% in the deep learning, respectively. Accuracy of *freespace* in the evidence filter dramatically increase to 96.7% and 95.9% from 70.4% and 70.6%. In addition, the *obstacle* accuracy increase slightly from 83.7% and 78.3% by 2.3% and 1.2%, respectively.

The parking scenario has more ratio of *obstacle* than others. In the parking scenario, accuracy of *slot marking*, *freespace*, and *obstacle* in the evidence filter increase drastically by 17.1%, 21.8%, and 14.4%, from 72.8%, 70.5%, and 64.7%, respectively. Although the curved and loop closing scenarios have more ratio of *slot marking* than others, the performances of the evidence filter in two scenarios are improved similarly with other scenarios.

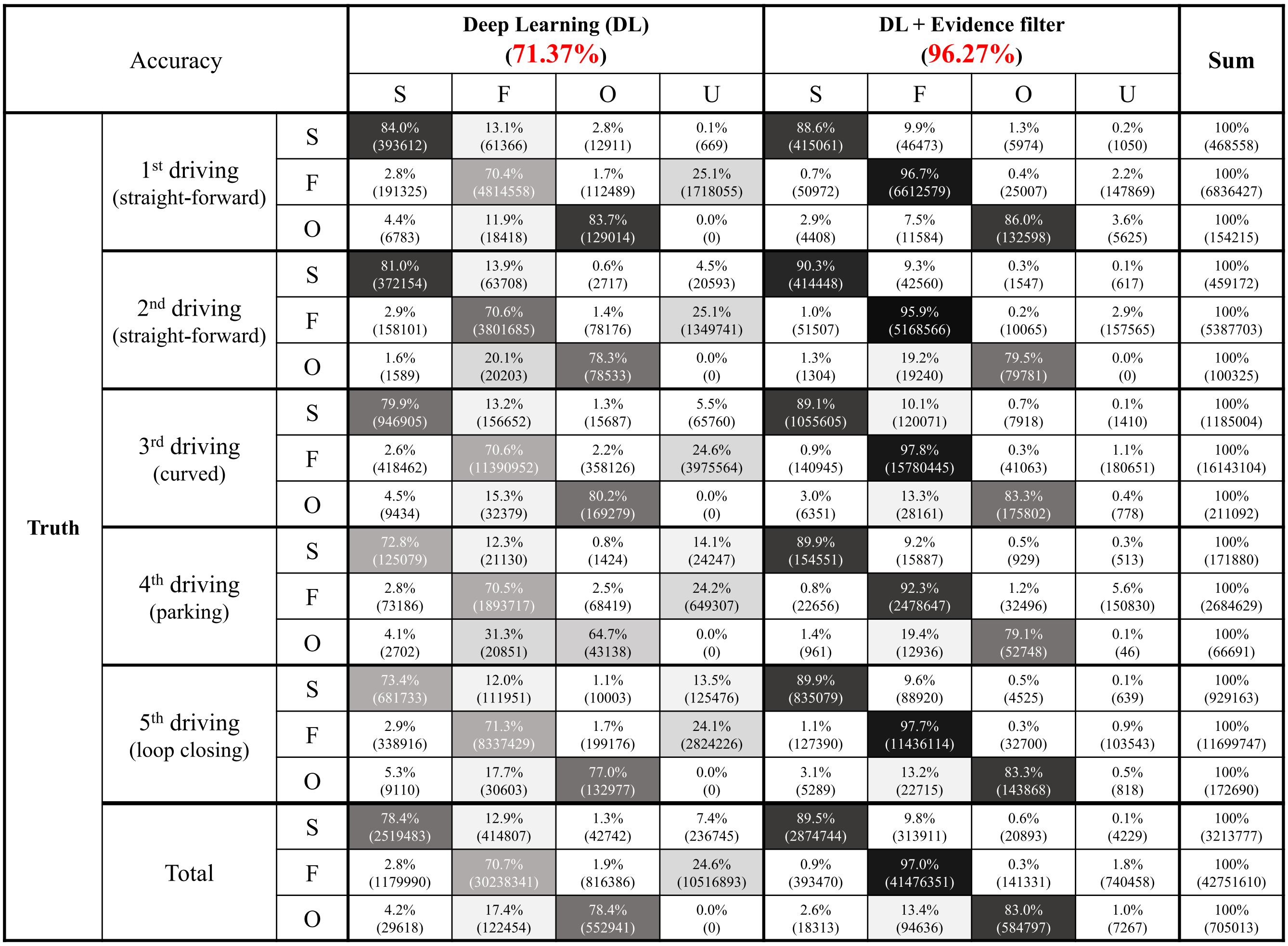
The total samples from the multiple driving scenarios are summed up into the final row. The classification performance of *slot marking*, *freespace*, and *obstacle* is boosted by 11.1%, 26.3%, and 4.6% from 78.4%, 70.7%, and 78.4%, respectively. To summarize the result of the proposed algorithm, the total accuracy is acquired. The evidence filter improves the performance of semantic segmentation from 71.37% to 96.27%.

The most dominant factor related to the improvement is the consideration of the ego-vehicle occlusion. The ego-vehicle regions, represented as unknown regions, can be estimated by the evidence filter. If the ego-vehicle regions (a blue rectangle in figure 13-(b)) is excluded in the evaluation process, accuracy of the deep learning approach is 92.74%. When the accuracy of the evidence filter except the ego-vehicle region is approximated to be 96.27%, consideration of other factors improves the performance of the evidence filter by 3.53%.

C. COMPARISON WITH PROBABILISTIC FILTER

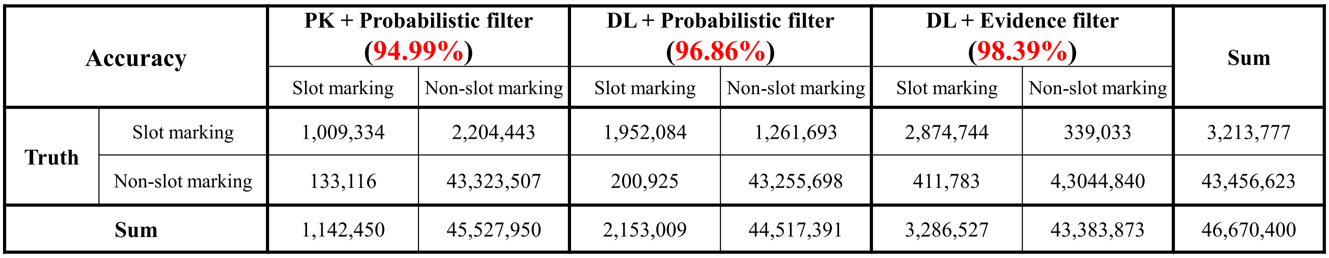
In order to confirm the improvement by the proposed algorithm, we compared the evidence filter with the previous method which is denoted as the probabilistic filter [11]. The probabilistic filter improves the performance of the slot marking detection by accumulating slot markings measured by a Pillbox Kernel [48]. For the comparison except for the effect of the feature detection method, the deep learning, as well as the pillbox kernel, is applied in the probabilistic filter. The three algorithms (the probabilistic filter with the pillbox kernel, the probabilistic filter with the deep learning, and the evidence filter with the deep learning) make the filtered images as shown in figure 16-(a), (b), and (c), respectively. Since the probabilistic filter can estimate only slot markings, confusion matrices with two states (slot marking and non-slot marking) are used to evaluate the performances. The three confusion matrices derived from three algorithms are represented in figure 15.

The accuracy of the probabilistic filter with the deep learning is 96.86% which is higher than the 94.99% of the probabilistic filter with the pillbox kernel. Since the pillbox kernel does not distinguish false positive slot markings from other obstacles within yellow circles (A) of figure 16, the false positive measurements can degrade the performance of the slot marking detection. The evidence filter causes the performance improved to 98.39% from 96.86% of the probabilistic filter due to two reasons: confidence map and GraphSLAM algorithm. The confidence map composed of five confidence models can relieve false positive estimation as represented in blue circles (B) of figure 16. In addition, the GraphSLAM algorithm in the evidence filter prevents misclassification by misalignment of slot markings as shown in red circles (C) of figure 16. On the other hand, the evidence filter does not only



Accuracy			Deep Learning (DL) (71.37%)				DL + Evidence filter (96.27%)				Sum
			S	F	O	U	S	F	O	U	
Truth	1st driving (straight-forward)	S	84.0% (393612)	13.1% (61366)	2.8% (12911)	0.1% (669)	88.6% (415061)	9.9% (46473)	1.3% (5974)	0.2% (1050)	100% (468558)
		F	2.8% (191325)	70.4% (4814558)	1.7% (112489)	25.1% (1718055)	0.7% (50972)	96.7% (6612579)	0.4% (25007)	2.2% (147869)	100% (6836427)
		O	4.4% (6783)	11.9% (18418)	83.7% (129014)	0.0% (0)	2.9% (4408)	7.5% (11584)	86.0% (132598)	3.6% (5625)	100% (154215)
	2nd driving (straight-forward)	S	81.0% (372154)	13.9% (63708)	0.6% (2717)	4.5% (20593)	90.3% (414448)	9.3% (42560)	0.3% (1547)	0.1% (617)	100% (459172)
		F	2.9% (158101)	70.6% (3801685)	1.4% (78176)	25.1% (1349741)	1.0% (51507)	95.9% (5168566)	0.2% (10065)	2.9% (157565)	100% (5387703)
		O	1.6% (1589)	20.1% (20203)	78.3% (78533)	0.0% (0)	1.3% (1304)	19.2% (19240)	79.5% (79781)	0.0% (0)	100% (100325)
	3rd driving (curved)	S	79.9% (946905)	13.2% (156652)	1.3% (15687)	5.5% (65760)	89.1% (1055605)	10.1% (120071)	0.7% (7918)	0.1% (1410)	100% (1185004)
		F	2.6% (418462)	70.6% (11390952)	2.2% (358126)	24.6% (3975564)	0.9% (140945)	97.8% (15780445)	0.3% (41063)	1.1% (180651)	100% (16143104)
		O	4.5% (9434)	15.3% (32379)	80.2% (169279)	0.0% (0)	3.0% (6351)	13.3% (28161)	83.3% (175802)	0.4% (778)	100% (211092)
	4th driving (parking)	S	72.8% (125079)	12.3% (21130)	0.8% (1424)	14.1% (24247)	89.9% (154551)	9.2% (15887)	0.5% (929)	0.3% (513)	100% (171880)
		F	2.8% (73186)	70.5% (1893717)	2.5% (68419)	24.2% (649307)	0.8% (22656)	92.3% (2478647)	1.2% (32496)	5.6% (150830)	100% (2684629)
		O	4.1% (2702)	31.3% (20851)	64.7% (43138)	0.0% (0)	1.4% (961)	19.4% (12936)	79.1% (52748)	0.1% (46)	100% (66691)
	5th driving (loop closing)	S	73.4% (681733)	12.0% (111951)	1.1% (10003)	13.5% (125476)	89.9% (835079)	9.6% (88920)	0.5% (4525)	0.1% (639)	100% (929163)
		F	2.9% (338916)	71.3% (8337429)	1.7% (199176)	24.1% (2824226)	1.1% (127390)	97.7% (11436114)	0.3% (32700)	0.9% (103543)	100% (11699747)
		O	5.3% (9110)	17.7% (30603)	77.0% (132977)	0.0% (0)	3.1% (5289)	13.2% (22715)	83.3% (143868)	0.5% (818)	100% (172690)
	Total	S	78.4% (2519483)	12.9% (414807)	1.3% (42742)	7.4% (236745)	89.5% (2874744)	9.8% (313911)	0.6% (20893)	0.1% (4229)	100% (3213777)
		F	2.8% (1179990)	70.7% (30238341)	1.9% (816386)	24.6% (10516893)	0.9% (393470)	97.0% (41476351)	0.3% (141331)	1.8% (740458)	100% (42751610)
		O	4.2% (29618)	17.4% (122454)	78.4% (552941)	0.0% (0)	2.6% (18313)	13.4% (94636)	83.0% (584797)	1.0% (7267)	100% (705013)

FIGURE 14. Confusion matrices of deep learning and evidence filter for 5 drivings.



Accuracy		PK + Probabilistic filter (94.99%)		DL + Probabilistic filter (96.86%)		DL + Evidence filter (98.39%)		Sum
		Slot marking	Non-slot marking	Slot marking	Non-slot marking	Slot marking	Non-slot marking	
Truth	Slot marking	1,009,334	2,204,443	1,952,084	1,261,693	2,874,744	339,033	3,213,777
	Non-slot marking	133,116	43,323,507	200,925	43,255,698	411,783	4,3044,840	43,456,623
Sum		1,142,450	45,527,950	2,153,009	44,517,391	3,286,527	43,383,873	46,670,400

FIGURE 15. Confusion matrices of probabilistic filter with pillbox kernel, probabilistic filter with deep learning, and evidence filter with deep learning.

have better performance than the previous work, but the filter also estimates the multiple states as shown in figure 16-(c).

D. PROCESSING PERIOD OF PROPOSED ALGORITHM

In order to process the proposed algorithm, a computer with the Intel (R) Core i7-8750H CPU at 2.20 GHz and NVIDIA GTX 1050TI GPU was used. The proposed algorithm is composed of three huge computing major modules: semantic segmentation, evidential merging, and GraphSLAM. In order to process the algorithm in real-time, the three modules are distributed to heterogeneous platforms. The semantic segmentation based on the deep learning was processed on the CUDA platform under the NVIDIA GTX1050TI GPU to accelerate the TensorFlow library. The evidential merging algorithm was performed on the OpenCL platform, supporting parallel processing, under the Intel (R) UHD Graphics 630 GPU to boost the evidence theory in each pixel. Finally, the GraphSLAM algorithm was optimized under the Intel (R) Core i7-8750H CPU. As shown in figure 17, the semantic segmentation, evidential merging, and GraphSLAM took

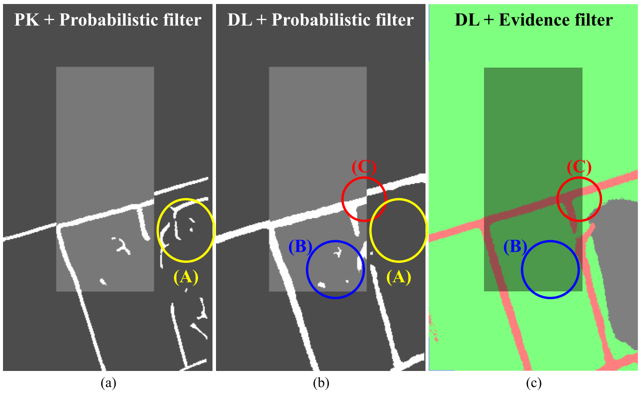


FIGURE 16. Result images derived from three algorithms : (a) probabilistic filter with pillbox kernel, (b) probabilistic filter with deep learning, and (c) evidence filter with deep learning.

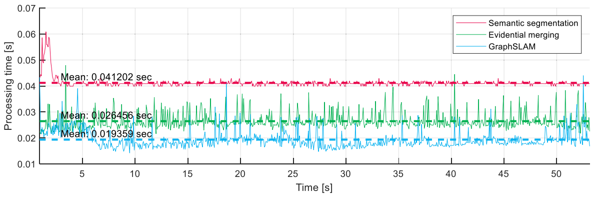


FIGURE 17. Processing time of three major parts in the proposed algorithm.

0.041 *sec*, 0.026 *sec*, and 0.019 *sec* on average, respectively. Since the three computing modules were performed on heterogeneous platforms, the longest processing time determines the processing period of the proposed algorithm. Accordingly, the average of the frame rate of the algorithm is 24.27 *FPS*.

VIII. CONCLUSION

AVM based semantic segmentation based on deep learning have five inherent problems: occlusion, misclassification, geometric distortion, calibration error, and stretched obstacle. The problems cause the poor performance of slot marking and obstacle detection. For overcoming the problems, we propose the evidence filter. The evidence filter estimates precise states around the ego-vehicle by accumulating the multiple semantic measurements based on the evidence theory and the GraphSLAM algorithm. In order to determine the reliability of the measurements, the confidence map based on the five models (ego-vehicle model, misclassification model, geometric distortion model, calibration error model, and stretched obstacle model) is considered in the filter. The evidence filter has two remarkable benefits compared with the previous works such as the deep learning [9] and the probabilistic filter [11].

1) Different from the probabilistic filter estimating only slot markings, the evidence filter can estimate the multiple states including slot markings and obstacles based on the evidence theory.

2) The evidence filter can estimate precise states in the regions, which are misclassified by measurements or occluded by the ego-vehicle, by applying the GraphSLAM algorithm and the confidence map based on the five confidence models. The improvement of the performance by the evidence filter is validated in the experiments compared with the deep learning and the probabilistic filter.

The evidence filter has some limitations. The filter has better performance at the low vehicle speed than the high vehicle speed because more measurements in the same region are accumulated. Accordingly, the evidence filter is suitable for parking lot environments where the vehicles are driven at a low speed. In addition, the evidence filter requires high computing powers for processing the deep learning, the evidence merging, and the GraphSLAM algorithm. It is difficult for a commercial vehicle to install high-performance computing units. In order to solve this problem, the authors plan to research edge computing based on the wireless network within the parking lot.

REFERENCES

[1] A. Miller. (Dec. 2016). *Automated Driving: The Technology and Implications for Insurance*. Thatcham Research. [Online]. Available: http://www.brakepro.org/assets/docs/event-resources/invehicle-tech-Dec16/Thatcham_Research_Webinar.pdf

[2] H. Banzhaf, D. Nienhüser, S. Knoop, and J. M. Zöllner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1827–1834.

[3] *Park4u: Parking Assistance System*. Accessed: Jun. 21, 2019. [Online]. Available: https://www.valeo.com/en/park4u-automated-parking/

[4] (2018). *Ford FUSION*. [Online]. Available: http://www.ford.com/cars/fusion/

[5] (2018). *BMW, Stress-Free Parking*. [Online]. Available: https://www.bmw.com.my/en/topics/discover/connected-drive/driver-assistance.html

[6] (2018). *Toyota, Enhanced Parking Support*. [Online]. Available: https://www.toyota.com/prius/prius-features/

[7] (2018). *Hyundai, New Technology*. [Online]. Available: http://org-www2.hyundai.com/eu/en/Innovation/Technology/NewTechnology/index.html

[8] C. Jang, C. Kim, S. Lee, S. Kim, S. Lee, and M. Sunwoo, "Re-plannable automated parking system with a standalone around view monitor for narrow parking lots," *IEEE Trans. Intell. Transp. Syst.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8624598/

[9] C. Jang and M. Sunwoo, "Semantic segmentation-based parking space detection with standalone around view monitoring system," *Mach. Vis. Appl.*, vol. 30, pp. 309–319, Oct. 2018. doi: 10.1007/s00138-018-0986-z.

[10] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[11] M. Lee, S. Kim, W. Lim, and M. Sunwoo, "Probabilistic occupancy filter for parking slot marker detection in an autonomous parking system using AVM," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2389–2394, Jun. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8424185/

[12] P. Degerman, J. Pohl, and M. Sethson, "Hough transform for parking space estimation using long range ultrasonic sensors," in *Proc. SAE World Congr. Exhib.*, Detroit, MI, USA, Apr. 2006.

[13] H. Satonaka, M. Okuda, S. Hayasaka, T. Endo, Y. Tanaka, and T. Yoshida, "Development of parking space detection using an ultrasonic sensor," in *Proc. 13th World Congr.*, Oct. 2006, pp. 8–12.

[14] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 1039–1044.

[15] S. H. Jeong, C. G. Choi, J. N. Oh, P. J. Yoon, B. S. Kim, M. Kim, and K. H. Lee, "Low cost design of parallel parking assist system based on an ultrasonic sensor," *Int. J. Automot. Technol.*, vol. 11, no. 3, pp. 409–416, 2010.

[16] S. Görner and H. Rohling, "Parking lot detection with 24 GHz radar sensor," in *Proc. 3rd Int. Workshop Intell. Transp.*, 2006, pp. 1–6.

[17] M. R. Schmid, S. Ates, J. Dickmann, F. Von Hundelshausen, and H.-J. Wuensche, "Parking space detection with hierarchical dynamic occupancy grids," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 254–259.

[18] R. Prophet, M. Hoffmann, M. Vossiek, G. Li, and C. Sturm, "Parking space detection from a radar based target list," in *IEEE MTT-S Int. Microw. Symp. Dig. Int. Conf. Microw. Intell. Mobility (ICMIM)*, Mar. 2017, pp. 91–94.

[19] H. G. Jung, Y. H. Cho, P. J. Yoon, and J. Kim, "Scanning laser radar-based target position designation for parking aid system," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 406–424, Sep. 2008.

[20] J. Zhou, L. E. Navarro-Serment, and M. Hebert, "Detection of parking spots using 2D range data," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2012, pp. 1280–1287.

[21] D. A. Thornton, K. Redmill, and B. Coifman, "Automated parking surveys from a lidar equipped vehicle," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 23–35, Feb. 2014. doi: 10.1016/j.trc.2013.11.014.

[22] C. Vestri, S. Bougnoux, R. Bendahan, K. Fintzel, S. Wybo, F. Abad, and T. Kakinami, "Evaluation of a vision-based parking assistance system," in *Proc. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2005, pp. 56–60.

[23] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Mach. Vis. Appl.*, vol. 21, no. 2, pp. 163–176, 2010.

[24] C. Unger, E. Wahl, and S. Ilic, "Parking assistance using dense motion-stereo: Real-time parking slot detection, collision warning and augmented parking," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 561–581, 2014.

[25] N. Kaempchen, U. Franke, and R. Ott, "Stereo vision based pose estimation of parking lots using 3D vehicle models," in *Proc. IEEE Intell. Vehicle Symp.*, vol. 2, Jun. 2002, pp. 459–464.

[26] H. G. Jung, D. S. Kim, P. J. Yoon, and J. H. Kim, "3D vision system for the recognition of free parking site location," *Int. J. Automot. Technol.*, vol. 7, no. 3, pp. 351–357, 2006.

[27] D. Kim, J. Choi, H. Yoo, U. Yang, and K. Sohn, "Rear obstacle detection system with fisheye stereo camera using HCT," *Expert Syst. Appl.*, vol. 42, nos. 17–18, pp. 6295–6305, Oct. 2015. doi: 10.1016/j.eswa.2015.04.035.

[28] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of road marking feature extraction," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2008, pp. 174–181.

[29] F. Zhang, H. Stähle, C. Chen, C. Buckl, and A. Knoll, "A lane marking extraction approach based on random finite set statistics," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1143–1148.

[30] C. Li, B. Dai, R. Wang, Y. Fang, X. Yuan, and T. Wu, "Multi-lane detection based on omnidirectional camera using anisotropic steerable filters," *IET Intell. Transport Syst.*, vol. 10, no. 5, pp. 298–307, 2016.

[31] C. Li, N. Qi, X. Yang, and B. Dai, "Road markings feature extraction from omni-directional image," in *Proc. Chin. Automat. Congr. (CAC)*, Oct. 2017, pp. 1223–1228.

[32] P. Mongkonyong, C. Nuthong, S. Siddhichai, and M. Yamakita, "Lane detection using randomized Hough transform," in *Proc. 8th TMSE Int. Conf. Mech. Eng.*, Dec. 2017, pp. 647–656.

[33] G. Bacchiani, M. Patander, A. Cionini, and D. Giaquinto, "Parking slots detection on the equivalence sphere with a progressive probabilistic Hough transform," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[34] J. Xu, G. Chen, and M. Xie, "Vision-guided automatic parking for smart car," in *Proc. IEEE Intell. Vehicles Symp.*, Oct. 2000, pp. 725–730. [Online]. Available: http://ieeexplore.ieee.org/document/898435/

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[36] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand, "Parking-stall vacancy indicator system, based on deep convolutional neural networks," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2017, pp. 655–660. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015181563&doi=10.1109%2FWF-IoT.2016.7845408&partnerID=40&md5=f9bd5fcabd29c0897a93b2581014e103

[37] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong, "Vision-based parking-slot detection: A benchmark and a learning-based approach," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2017, pp. 649–654.

[38] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5350–5364, Nov. 2018.

[39] B. Li, L. Yang, J. Xiao, R. Valde, M. Wrenn, and J. Leflar, "Collaborative mapping and autonomous parking for multi-story parking garage," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1629–1639, May 2018.

[40] J. K. Suhr and H. G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 21–36, Feb. 2014.

[41] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Feb. 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5681215/

[42] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[43] C. Jang. (2017). *AVM (Around View Monitoring) System Datasets for Auto Parking*. [Online]. Available: https://github.com/ChulhoonJang/avm_dataset

[44] K. Sentz and S. Ferson, *Combination of Evidence in Dempster-Shafer Theory*, vol. 4015. Pennsylvania, PA, USA: Citeseer, 2002.

[45] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, "Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3377–3392, Dec. 2015.

[46] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, nos. 5–6, pp. 403–429, 2006.

[47] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007. doi: 10.1002/rob.20204.

[48] S.-J. Han and J. Choi, "Parking space recognition for autonomous valet parking using height and salient-line probability maps," *Electron. Telecommun. Res. Inst. J.*, vol. 37, pp. 1220–1230, Dec. 2015.

CHANSOO KIM (S'16) received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, where he is currently pursuing the Ph.D. degree with the Automotive Control and Electronics Laboratory (ACE Lab). His main fields of interests are object detection and recognition, deep learning, localization, mapping, simultaneous localization, and mapping, map updating, real-time systems for autonomous cars, and systems of the autonomous vehicle. His current research activity includes precise map generation for an autonomous car.

SUNGJIN CHO (S'17) received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree with the Automotive Control and Electronics Laboratory. His main fields of interests include precise positioning and localization, information fusion theories, and real-time systems for an autonomous car. His current research activity includes the design of parallel computing architecture for vehicle localization.

CHULHOON JANG (S'13–M'17) received the B.S. degree in mechanical engineering and the Ph.D. degree in automotive engineering from Hanyang University, Seoul, South Korea, in 2011 and 2017, respectively. His main fields of interests are autonomous driving systems, image processing, and deep learning. His current research activities include object detection and classification using multiple sensor fusion for urban autonomous driving.

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, USA, in 1985, and has worked in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he worked on the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has led research activities as a Professor with the Department of Automotive Engineering, Hanyang University. His work 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).

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 (ACE Lab), Department of Automotive Engineering, Hanyang University, doing research on system design and implementation of autonomous cars. Since 2015, he has been with the Valeo Driving Assistance Research, Bobigny, France, working on 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 activities include hardware and software platform design for autonomous cars based on distributed real-time embedded systems, and in-vehicle networks.