

# Parking Space Recognition for Autonomous Valet Parking Using Height and Salient-Line Probability Maps

Seung-Jun Han and Jeongdan Choi

**An autonomous valet parking (AVP) system is designed to locate a vacant parking space and park the vehicle in which it resides on behalf of the driver, once the driver has left the vehicle. In addition, the AVP is able to direct the vehicle to a location desired by the driver when requested. In this paper, for an AVP system, we introduce technology to recognize a parking space using image sensors. The proposed technology is mainly divided into three parts. First, spatial analysis is carried out using a height map that is based on dense motion stereo. Second, modelling of road markings is conducted using a probability map with a new salient-line feature extractor. Finally, parking space recognition is based on a Bayesian classifier. The experimental results show an execution time of up to 10 ms and a recognition rate of over 99%. Also, the performance and properties of the proposed technology were evaluated with a variety of data. Our algorithms, which are part of the proposed technology, are expected to apply to various research areas regarding autonomous vehicles, such as map generation, road marking recognition, localization, and environment recognition.**

**Keywords:** Autonomous valet parking, autonomous vehicle, height map, salient-line, straight-line model, dense motion stereo, obstacle detection, Bayesian classifier, parking slot recognition.

---

Manuscript received Aug. 9, 2014; revised June 16, 2015; accepted Aug. 21, 2015.

This work was supported by the Transportation & Logistics Research Program of the Ministry of Land, Infrastructure and Transport, Rep. of Korea (ID-79209, 14TLRP-B2078228-01).

Seung-Jun Han (corresponding author, joshua@etri.re.kr) and Jeongdan Choi (jdchoi@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

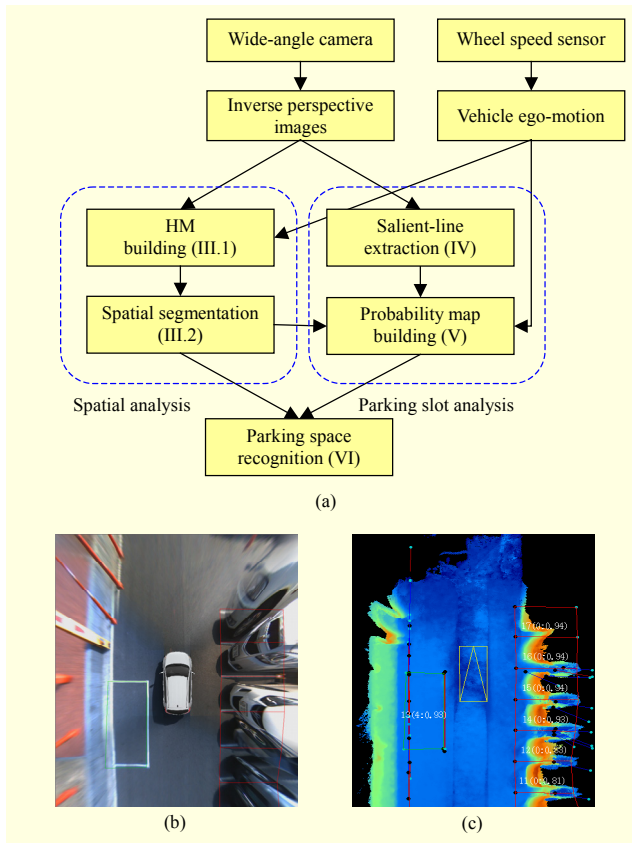
## I. Introduction

Over the past decade, autonomous driving technologies have made outstanding advancements. As a result, many companies including Google, Mercedes, BMW, and Volvo have presented autonomous driving demonstrations, with a plan to initiate their sales of autonomous vehicles by 2020. Also, they predicted that fully automated driving vehicles could be ready by 2025 [1]. Furthermore, in five states in the USA, including Nevada, experiments on autonomous vehicles have been legally approved, and many other states are also considering doing the same [2].

An autonomous valet parking (AVP) system is a complete autonomous unmanned vehicle system that drives the car to a safe parking lot and parks it there on behalf of the driver. Also, the system drives the car to a position where the driver can board it [3]. An AVP system drives a car at a low speed or parks it within a limited area, such as a parking area, or surrounding road. This is an advanced form of the currently available automatic parking system, and it is expected to become an integral part of the first generation of commercial autonomous vehicles.

The technologies necessary to realize an AVP system include technology to control the vehicle; explore and recognize the environment around the vehicle; and determine an appropriate parking maneuver for a designated space. Also, an AVP system must secure high reliability in its recognition and decision technologies since it has to find a parking space and park a car without the aid of a driver.

This study aims to develop a suitable parking space recognition technology that is sufficiently reliable for an AVP system. The target system must not only find an available parking space, but it must also decide whether the space would



**Fig. 1.** Proposed parking space recognition algorithm and its results: (a) block diagram of proposed method, (b) IPI and recognition results; green box is empty slot and red boxes are occupied spaces, and (c) saliient-line probability map overlaid with height map coded by color: blue, green, and red are road surface, curbs (or lower objects), and obstacles, respectively.

be suitable; that is, whether the space is for disabled persons only, whether the type of space matters, and whether there is sufficient space.

Figure 1 shows the vision-based parking space recognition technology proposed in this paper. The technology (algorithm) is primarily divided into three parts. The first step is a spatial analysis of the vehicle by using a height map (HM), which is obtained through dense motion stereo and map fusion techniques and provides information about the vehicle's height. This process is carried out by a background thread because the results are static information and require much computation. The next step is parking space detection and analysis using a new saliient-line extraction and its probability map. In particular, this paper proposes a new line extractor with non-separable kernel estimation and fast convolution for the saliient line extraction. The last step combines the spatial and road-marking data to identify a parking space based on a Bayesian classifier. The results of this paper show a real-time execution time of up to 10 ms and a reliability of over 99%.

This paper is organized as follows. Section II discusses related research trends. Sections III through VI describe the details of the proposed technologies. Section VII presents and discusses the experimental results. Finally, Section VIII provides some concluding remarks.

## II. Related Works

For parking space recognition, the following two technologies are required: recognition of an empty space and recognition of a parking lot. Thus, these technologies can be classified into space recognition, parking-space marking recognition, and recently, a combination of the two.

Space recognition technology is used to find an empty space for parking. Most parking-assist system products currently available in the market adopt this method. This is categorized by space-recognizing sensors; the most common of which is an ultra-sonic sensor [4]–[5], although range sensors such as LiDAR (light detection and ranging) sensors [6]–[7] and microwave radar sensors [8]–[9] are also used. Recently, research on vision-based space recognition techniques has been actively conducted. In the initial stage, this technique adopted stereo vision technology using two cameras [10], and nowadays, motion-stereo vision technology based on a mono camera is also used [11]–[12]. This technology is divided into feature-based sparse reconstruction [11] and dense reconstruction [12]. The dense reconstruction technology is known for its superior performance [12].

Parking space marking recognition technology is used to recognize the lines dividing a parking lot using vision technology. Usually, images taken by wide-angle cameras are converted into inverse perspective images (IPIs). Since parking lots consist of several straight lines, transforms may be used to locate such lines. For example, Hough [13] and Radon [14] transformations can be used. In addition, the characteristics of the corner lines [15] and a pattern matching technology [16] are also examined and used.

A combination of space and parking-space data is useful to determine an available parking space. Recently, the technology to combine range sensor and vision information was introduced [16].

Finding a parking space by recognizing space and parking spaces is essential for an AVP. In addition, legal matters, such as spaces dedicated to the disabled, are also considered; moreover, low obstacles, such as a parking lot stopper or curb, which partially block a vehicle's entrance, need to be considered when developing technologies to determine the appropriate parking method. The technologies proposed in this paper are fully vision-based and can handle a variety of parking space types and obstacles of various heights.

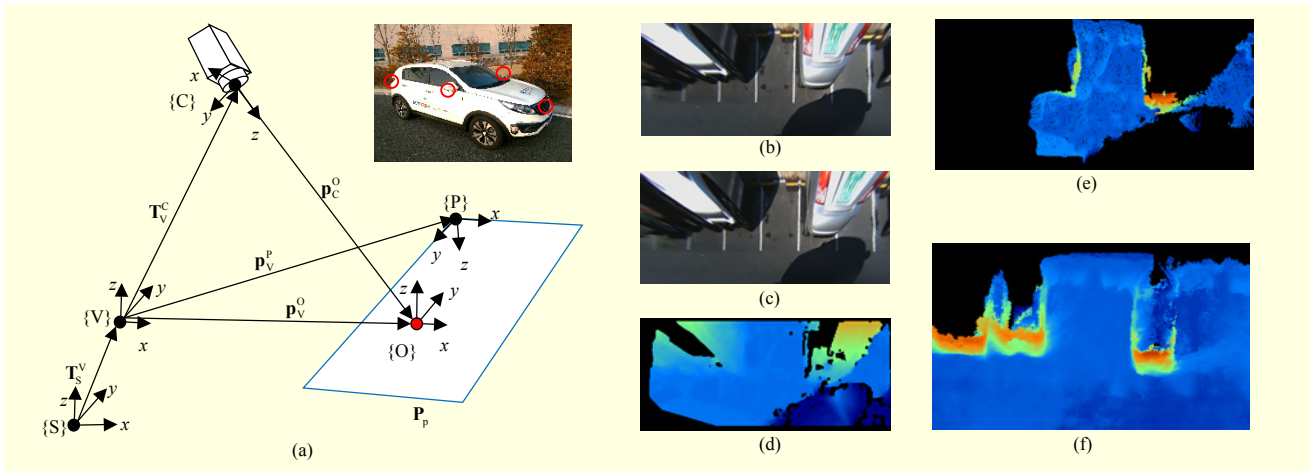


Fig. 2. HM building: (a) coordinate system used in this paper (attached picture shows experimental vehicle and camera mounting positions), (b) and (c) are rectified stereo pair from IPI, (d) estimated disparity map, (e) native HM, and (f) improved HM using modified TMF. Note that poor quality of original map has been significantly improved.

### III. Spatial Segmentation through HM

The height information from objects that are realized to be on the road surface is very important in building a perception of the related environment. This paper proposes a spatial analysis method that makes use of an HM. Our recent research [17] provided the technology for effective HM generation using dense-motion stereo from IPIs. Notably, the study proposed effective stereo rectification from IPIs and a fusion method using a modified temporal median filter (TMF). This section briefly describes how to make an HM using IPIs and dense-motion stereo efficiently, and then proposes a spatial segmentation method based on height information.

#### 1. HM Building

HM generation using dense-motion stereo technology commonly consists of the following four steps: obtaining a rectified stereo image pair at the previous and the present time using structure from motion (SFM), estimating a disparity map, mapping 3D points obtained by triangulation into a grid map, and map fusion to improve the quality of the map [12].

In particular, a high-precision HM can be efficiently obtained using an IPI, which is converted to remove the projective distortion of the road surface [17]. In general, an IPI can be calculated by a homographic and linear transformation. However, this approach may cause a blind area near the vehicle and is barely able to estimate camera parameters. Accordingly, it is appropriate to transform directly from a camera's intrinsic and extrinsic parameters [17].

Figure 2 illustrates the HM building process. First,  $\{V\}$ ,  $\{C\}$ , and  $\{P\}$  are the coordinate frames of the vehicle, camera, and projective plane, respectively. Second,  $\{O\}$  is a frame lain on

the intersection position of the projected plane  $\{P\}$  and the extended vector of the camera's optical center. Finally,  $\{S\}$  is the map's origin frame defined by the starting point of the vehicle's movement. The transformation matrix of an IPI,  $\mathbf{H}_{IPI}$ , is defined by the following expression [17]:

$$\mathbf{H}_{IPI} = F_w^1(\mathbf{T}_V^{C-1} \mathbf{P}_p). \quad (1)$$

Here,  $F_w^1(\cdot)$  is a non-linear function obtained from the camera's intrinsic calibration to convert the world coordinate into an image coordinate. Furthermore,  $\mathbf{P}_p$  is the set of all points of the projection area in the plane  $\{P\}$ ;  $\mathbf{T}_V^C \in \mathbb{R}^{4 \times 4}$  is the camera's transformation matrix with basis  $\{V\}$ , and can be calculated based on the camera's extrinsic calibration. Also, the camera parameters of an IPI for triangulation in stereo can be derived from this geometric information [17].

The stereo rectification ((b), (c)) of an IPI pair can simply be approximated by rotation about the optical center, and it can be determined using the SFM technique [17]. The disparity map (d) is estimated using the rectified image pair. Moreover, 3D coordinates are calculated by triangulation and mapped onto the grid map based on frame  $\{S\}$  through quantization. At this time, each cell of the grid takes only the maximum value of the height data to be mapped to each cell.

As shown in Fig. 2(e), the generated grid map is of poor quality due to the wide field-of-view. However, the use of a map fusion technique can dramatically improve the quality. This paper uses a grid-map fusion technology based on a modified TMF (f) [17]. The method is a simple but effective sorting-based TMF. The input data is inserted into a buffer in descending order to maintain the buffer's status. If the amount of inserted data reaches the buffer's limitation, then the maximum and minimum values are removed to retain data consistency [17].

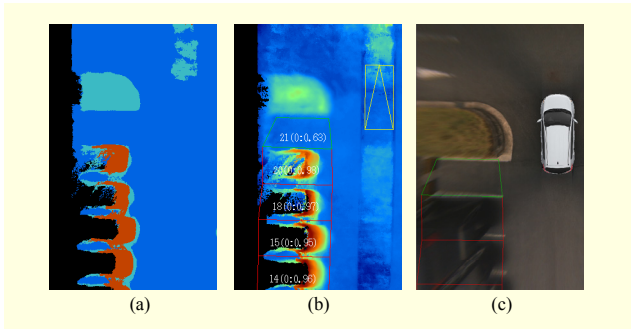


Fig. 3. Spatial segmentation: (a) spatial segmentation result, (b) HM overlaid with recognition results, and (c) successfully recognized parking space beside flowerbeds.

## 2. Spatial Segmentation

The spatial segmentation of the surrounding space of a vehicle can be achieved by a relatively simple method thanks to the high quality of the HM previously described. In this paper, spatial segmentation is classified into three categories according to the heights of the road surface, low obstacles, and high obstacles. The road surface is defined to be the space in which the vehicle can travel freely; low obstacles form the border of the road, such as sidewalks, flowerbeds, curbs, and so on; and high obstacles are represented by objects such as parked vehicles and trees. A region growing method can effectively separate a space by using a seed value to represent the height of each category, because the map has only height information, as shown in Fig. 3. Next, the contour values of the low obstacles include the salient-line feature (Section IV) to process the road boundary. This process makes it possible to recognize parking spaces located next to road borders.

## IV. Salient-Line Extraction

Parking space markings are usually drawn on a road surface

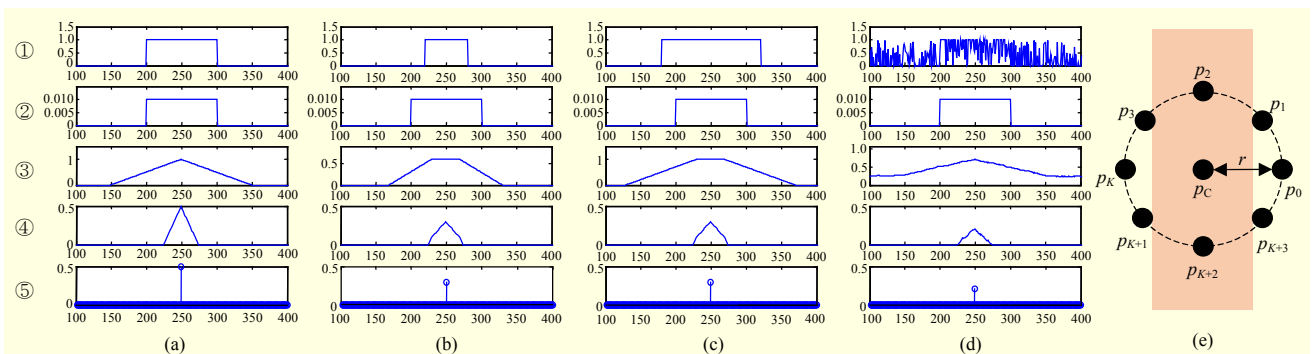


Fig. 4. Responses of proposed salient-line extractor: characteristics for variety of input signals, such as (a) idea, (b) narrow, (c) wide, and (d) high-intensity noise; here, each row is response of ① input signal, ② convolution kernel, ③ convolution, ④ MEF, and ⑤ local maxima. In particular, (e) MEF helps to find peak of convolution responses, and has sparse-kernel mask (shown in case of  $K = 4$ ).

in straight lines of a fixed thickness. This section proposes a method for efficiently searching for straight lines having a specific thickness. Figure 4 describes the motivation for this technique. As is well known, the convolution response of the symmetrical input signal (①) to the same signal (②) has a maximum peak at the center (③). Hence, the center position of the input signal (⑤) can be found by using local maxima detection (④). Since the most similar signal in all directions of a straight line is a circle with a diameter equal to the thickness of the line, in this paper, we propose to use a pillbox kernel having the same diameter as the thickness of the line. More particularly, this section introduces fast convolution techniques for the pillbox kernel and morphological extrema filter (MEF) for robust peak detection (Fig. 4(e)). The proposed method, as shown in Fig. 4, represents a very robust performance against the input signal error and high-intensity noise.

### 1. Pillbox Kernel Estimation and Fast Convolution

The pillbox (or disk) kernel (Fig. 5(a)) is mainly used as a smoothing filter and is defined as

$$\mathbf{K}_{\text{pillbox}}(u, v) = \begin{cases} \frac{1}{\pi r^2} & \text{if } \sqrt{u^2 + v^2} < r, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where,  $u$  and  $v$  are coordinates based on the center of the kernel, and  $r$  is the radius of the kernel. Approximately  $2(2r + 1)^2$  arithmetic operations per pixel are required for the convolution operation of such a 2D kernel. In a recent study, Elboher announced a very efficient Gaussian kernel filtering technique using a staked integral image and the separability property of the Gaussian kernel [18]. The separability property is that a 2D kernel,  $\mathbf{K}$ , is said to be separable if it can be decomposed as the convolution of two 1D kernels,  $\mathbf{v}$  and  $\mathbf{h}$ , such that  $\mathbf{K} = \mathbf{v} \otimes \mathbf{h}$ . Namely, a 2D kernel is a matrix of rank one.



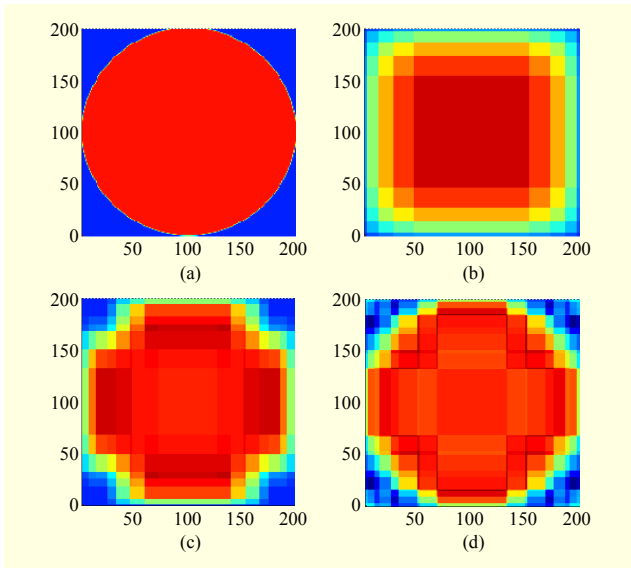


Fig. 5. Estimated pillbox kernel: (a) native pillbox kernel in case of  $r = 100$  (rank = 61), proposed estimated kernel in cases of (b)  $N = 1$ , (c)  $N = 2$ , and (d)  $N = 3$ ; these have  $M = 5$ .

In this paper, we propose a way to extend Elboher's method for a general non-separable 2D kernel. First, the symmetric 1D kernel can be approximated by the sum of the "weighted slices" [18] as follows:

$$\mathbf{h}(u) = \sum_{m=1}^M W_m(x), \quad W_m(x) = \begin{cases} w_m & \text{if } -p_m < x < p_m, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Here,  $W(x)$  is the "weighted slices" function,  $M$  is the order of the weighted slices,  $w$  is a weight, and  $p$  is a partition size. The optimum parameters,  $(w, p)$ , of the weighted slices can be determined by finding the value to minimize the  $l_2$  norm of the output error. In addition, the output is easily calculated by an integral sum [18].

The non-separable kernel with rank  $R$  is the sum of the separable kernel of as much as  $R$ . Therefore, the kernel can be approximated as the sum of a dominant separable kernel ( $N \ll R$ ) as follows:

$$\begin{aligned} \mathbf{K}_{\text{pillbox}}(u, v) &= \sum_{n=1}^R \mathbf{K}_n(u, v) \\ &\cong \sum_{n=1}^N \mathbf{v}_n(v) \mathbf{h}_n(u) \quad (N \ll R). \end{aligned} \quad (4)$$

Here,  $\mathbf{K}_n$  is an  $n$ th separable 2D kernel assumed to be sorted in descending order by the magnitude of each singular value, and  $N$  is the order of the dominant separable kernels. In (4),  $\mathbf{v}(v)$  and  $\mathbf{h}(u)$  are the vertical and horizontal 1D kernels, respectively. Substituting (3) into (4), it is possible to obtain a final approximate expression, such as the following:

1. Compute the horizontal cumulative sum of input image  $\mathbf{I}(u, v)$ .

$$\mathbf{H}(u, v) = \sum_{x=0}^u \mathbf{I}(x, v)$$

2. Compute the convolution of each horizontal separable kernel.

$$\mathbf{C}_{n \in N}(u, v) = \sum_{m=1}^M w_{nm} (\mathbf{H}(u + p_{nm}, v) - \mathbf{H}(u - p_{nm} - 1, v))$$

3. Compute the vertical cumulative sum of horizontal convolution.

$$\mathbf{V}_{n \in N}(u, v) = \sum_{y=0}^v \mathbf{C}_n(u, y)$$

4. Compute the sum of vertical convolution result.

$$\tilde{\mathbf{I}}(u, v) = \sum_{n=1}^N \sum_{m=1}^M z_{nm} (\mathbf{V}_n(u, v + q_{nm}) - \mathbf{V}_n(u, v - q_{nm} - 1))$$

Fig. 6. Fast convolution algorithm through estimated pillbox kernel.

$$\tilde{\mathbf{K}}_{\text{pillbox}}(u, v) \cong \sum_{n=1}^N \left( \sum_{m=1}^M V_{n,m}(v) \sum_{m=1}^M H_{n,m}(u) \right). \quad (5)$$

Here,  $V(v)$  and  $H(u)$  are the vertical and horizontal weighted slice functions, respectively. The examples of the estimated pillbox kernel obtained by (5) are shown in Fig. 5. Also, Table 1 shows the optimal parameters for the case shown in Fig. 5. In addition, the fast convolution algorithm using (5) is shown in Fig. 6. The proposed algorithm is an  $O(1)$  algorithm; that is, it takes only about  $6NM$  operations per pixel. For example, the operation efficiency is 40 times the native kernel in the case  $r = 15$ ,  $N = 2$ , and  $M = 4$ .

## 2. MEF and Straight-Line Modeling

This paper proposes a new MEF to effectively detect the peak values, as shown in Fig. 4(e). MEF gain is calculated from a mask with  $K$  pairs, and is determined using the following numerical expression:

$$\begin{aligned} \text{MEF} &= \max_{0 \leq k < K} \left\{ \min \left[ \pm(p_C - p_k), \pm(p_C - p_{K+k}) \right] \right\} \\ &\text{for } p_k = p_C + r e^{j(\pi k/K)}. \end{aligned} \quad (6)$$

Here,  $p_C$  is the value of the center of the kernel,  $p_k$  is the value of the  $k$ th kernel mask, and  $r$  is the kernel radius, which is the same as the pillbox. MEF has a sparse-circle mask whose radius is  $r$ , where each filter mask is symmetric with respect to the point  $p_C$ . A positive filter gain is applied to detect a bright marking (most road marking); whereas, a negative gain is applied in the case of a dark marking. As shown in Fig. 4(④), MEF helps to detect the peak accurately to make the pillbox convolution response more sharp. As the next step, to locate the peak, a local maxima approach is conducted followed by a hysteresis threshold [19].

A Radon transform is applied for straight-line modeling of detected peaks. A Radon transform consists of an integral of a function over straight lines, and can find the line parameters effectively and accurately [14]. The following steps are taken

Table 1. Optimal parameters of estimated pillbox kernel in case of  $M = 5$  and  $r = 100$ .

$N$	$(p, w)$ for $H(u)$	$(q, z)$ for $V(v)$	MSE*
1	(100, 0.169453), (97, 0.317627), (87, 0.258585), (73, 0.166367), (54, 0.129960)	(100, 0.167997), (97, 0.318749), (87, 0.258850), (73, 0.165857), (54, 0.129053)	0.2792
2	(100, 0.285734), (93, 0.182810), (87, 0.110683), (52, -0.032190), (27, -0.008107)	(82, 0.136443), (75, 0.153445), (68, 0.656909), (51, 0.636268), (32, 0.275134)	0.2012
	(77, -0.013927), (76, 0.382068), (67, 0.765396), (54, 0.692644), (40, 0.505191)	(99, 0.218989), (94, 0.180884), (68, -0.166268), (51, -0.169536), (32, -0.073390)	
3	(100, 0.062297), (87, 0.171360), (68, 0.267790), (51, 0.316682), (33, 0.223372)	(100, 0.379187), (91, 0.197857), (85, 0.155494), (97, 0.355243), (32, -0.023153)	0.1692
	(100, -0.430637), (97, -0.352791), (68, 0.325662), (51, 0.356790), (33, 0.223243)	(97, 0.231413), (66, -0.321408), (72, -0.242799), (51, -0.376739), (34, -0.329831)	
	(100, 0.415388), (91, -0.326408), (76, -0.559394), (49, 0.282192), (32, 0.422305)	(100, 0.520380), (91, -0.115296), (84, -0.939500), (48, 0.173419), (32, 0.443222)	

\*MSE is mean squared error between native kernel and estimated kernel.

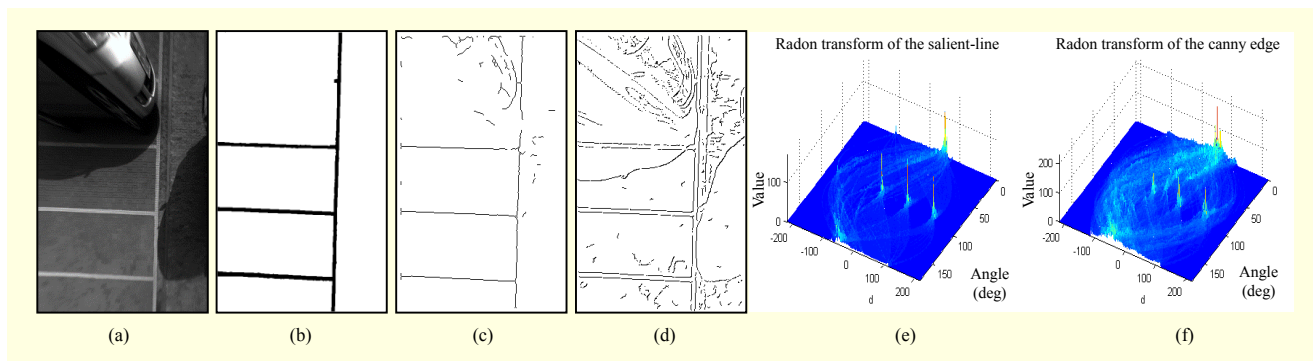


Fig. 7. Salient-line extractor characteristics: (a) input image for brick road with strong shadow, (b) ground truth generated by retouching, (c) proposed salient-line extractor result, (d) Canny edge detector result, in which optimal parameters were found manually, (e) Radon space of salient-line (c), and (f) Radon space of edge (d). Note that salient-line extractor has successfully extracted road marking in difficult case. Also it was represented by small data set.

to improve the straight-line model obtained by the Radon transform process: finding and integrating lines that have a similar slope and that are within a certain distance, and then applying straight-line fitting to find optimal line parameters.

Figure 7 shows the characteristics of the proposed detector in a situation where there is a brick road with a strong shadow. The salient-line extractor is compared with a Canny edge detector that has its parameters manually tuned ( $\sigma = 5.1$ ). Many of the existing algorithms found the space marking by using edge detection techniques [13]–[14], [16]. Despite using a Canny edge detector and optimum parameters, which is known to be robust, the result of the edge detection shows that much noise is generated. On the other hand, the proposed algorithm shows not only that it is able to dramatically remove such noise, but that it is also able to find a significantly improved peak in Radon space; moreover, the relatively small number of pixels it detects could benefit its computational

efficiency several times over.

## V. Salient-Line Probability Map (SPM) Building

The salient-line extraction technique can effectively detect road markings, as described in Section IV. However, as shown in Fig. 8, the extraction results may also contain noise due to objects such as parked vehicles, curbs, flowerbeds, and so on.

This section describes a method to remove such noise. In an IPI, since distortion of a road surface is removed, any remaining lines representing road markings keep their shape. However, those lines representing non-road markings vary in shape (length and slope) relative to the camera position (in our study, there were four cameras attached to the test vehicle). It is from such a property that we devise a method to determine those lines that represent road markings.

Assuming lines are extracted while the vehicle is moving, we can attempt to form groups of lines that are representative

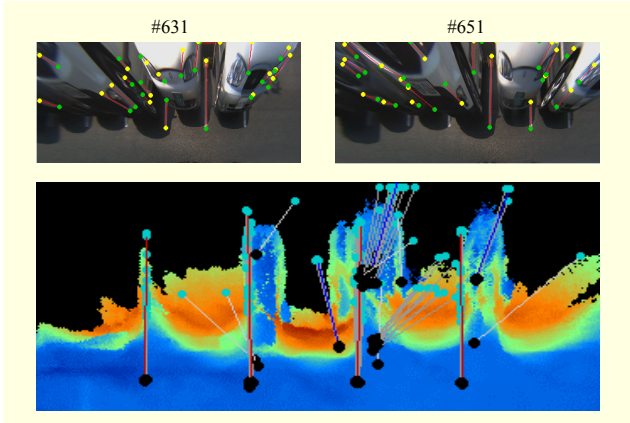


Fig. 8. SPM building example: upper row is input images with detected salient-lines at difference times; lower row is SPM overlaid with HM; red lines are determined as road marking; grey lines are matched line history; and blue lines are its mean value.

of a single object or marking. Considering a single such *line group*, we can then investigate the distribution of the shapes of the lines within this line group, and from this, we can determine how likely (in terms of a probability) it is that this particular line group is representative of an actual road marking. The distribution of the shapes of lines within a line group is defined to be an SPM.

The following describes the SPM procedure. First, we take the coordinates of the end points of a line and map them onto  $\{S\}$  using the following equation:

$$\mathbf{x}_s = \mathbf{T}_s^V \mathbf{T}_V^I \mathbf{x}_I. \quad (7)$$

Here,  $\mathbf{x}_s$  and  $\mathbf{x}_I \in \mathbb{R}^{4 \times 2}$  are homogeneous vectors of the end points of a straight line in  $\{S\}$  and in  $\{I\}$  (image coordinate frame), respectively;  $\mathbf{T}_s^V$  is the relative position of the vehicle obtained from the vehicle's wheel speed sensors; and  $\mathbf{T}_V^I$  is a mapping from  $\{V\}$  onto  $\{I\}$ .

In the next step, the end points of each line segment are compared with information contained within the HM, and any line found not to be a part of the road surface is removed. The remaining lines are then grouped, and it is at this point that only the last line to have entered a group is then used for comparison against any subsequent new line attempting to join the same group. The following discriminant is used to determine whether a new line should be accepted into a line group:

$$D = \begin{cases} \text{True} & \text{if } \min_{i,k \in \{1,2\}} \|\mathbf{L}_M^i - \mathbf{L}_S^k\|_2 < \tau_x \wedge \|\angle \mathbf{L}_M - \angle \mathbf{L}_S\| < \tau_\theta, \\ \text{False} & \text{otherwise.} \end{cases} \quad (8)$$

Here,  $\mathbf{L} \in \mathbb{R}^{2 \times 2}$  is a matrix containing two end point vectors of a line. Namely,  $\mathbf{L}_M^i$  is the  $i$ th column vector on the SPM, and  $\mathbf{L}_S^k$  is the  $k$ th column vector obtained by (7). In addition,

$\angle \mathbf{L}_M$  and  $\angle \mathbf{L}_S$  denote the slope of the line in the SPM and the image, respectively; and  $\tau_x$  and  $\tau_\theta$  are the threshold values for the distance between the end points and slope of the line, respectively.

Lastly, when a candidate line for a line group is matched to the group, its line information is added and the probability of the line group is updated. Here, the probability in question is derived from the standard deviation of the slope and length of a line. Lines are determined to constitute a road marking if this probability is within a given error range ( $\tau_\sigma$ ) and the matching count is greater than a given number ( $\tau_N$ ). The relevant discriminant is as follows:

$$D_{\text{road}} = \begin{cases} \text{True} & \text{if } \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{S}_i - \boldsymbol{\mu})^2} < \tau_\sigma \wedge N > \tau_N, \\ \text{False} & \text{otherwise.} \end{cases} \quad (9)$$

Here,  $N$  is the matching count;  $\mathbf{S}_i \in \mathbb{R}^{2 \times 1}$  is the  $i$ th vector of the length and slope of the line in the group; and  $\boldsymbol{\mu} \in \mathbb{R}^{2 \times 1}$  is the mean of vectors  $\mathbf{S} \in \mathbb{R}^{2 \times N}$ .

In addition, the road marking information in the probability map retains the same properties as in the digital map [20]. Therefore, in this study, a digital map-based localization method is used to locate the vehicle position at the parking stage after determining a place to park.

## VI. Parking Space Recognition

This section describes how an AVP system identifies and confirms a parking space. Figure 9 shows various parking spaces and a simplified model for recognition purposes. Parking patterns are to be classified into three types — perpendicular, angled, and parallel. In addition, legal restrictions, such as the dimensions of a parking space for the disabled, should be considered. Therefore, the parking spaces are classified into five categories — perpendicular, perpendicular-disabled, angled, angled-disabled, and parallel. Moreover, it must ensure that the parking space is empty.

As shown in Fig. 9(b), the parking space recognition model proposed in this paper is based on a parallelogram. Namely, the space consists of pairs of parallel segments, A-B and C-D. Segments A and B must exist, but C and D may not. Note that neither A nor B has to necessarily be an actual road marking, as shown by the green area in Fig. 9. If this is indeed the case, then either line segment can be determined by including the boundary information of obstacles in HM using a contour tracing technique, as described in Section III. This paper proposes a Bayesian classifier to recognize a parking space. The Bayesian classifier used in this study is defined based on the following classifiers using a maximum a posteriori (MAP)

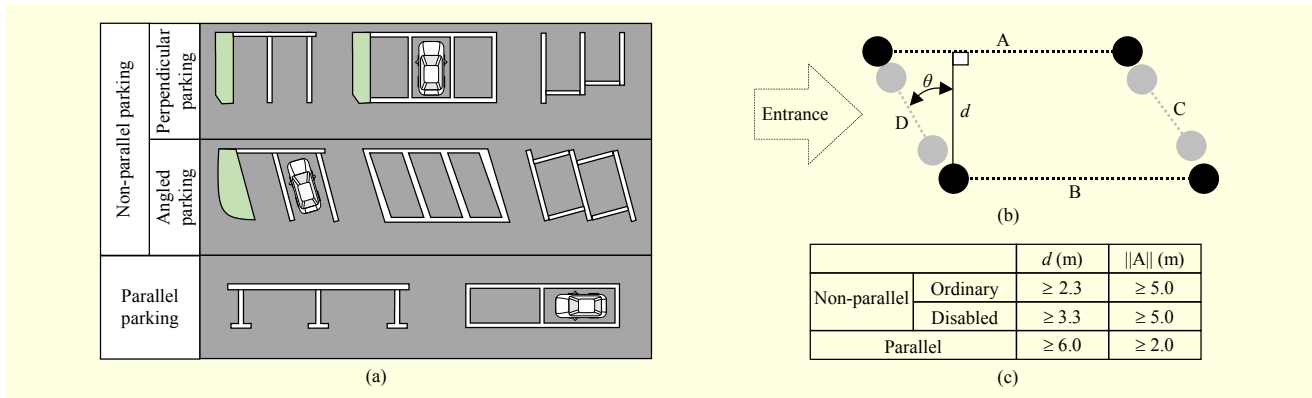


Fig. 9. Category of parking pattern and simplified model: (a) parking spaces can be classified roughly into three types by form — perpendicular, angled, and parallel — and two types by users — disabled or not disabled. Note that parking space includes not only markings but also flower beds, sidewalks, and so on (green regions), (b) proposed simple parking space model for recognition, and (c) legal regulations of parking laws in Rep. of Korea.

estimate [21]:

$$\mathcal{D}_{\text{MAP}} = \arg \max_{\mathbf{z} \in \mathcal{A}} P(\mathbf{x}|\mathbf{z})P(\mathbf{z}). \quad (10)$$

Here,  $\mathbf{x} = [x_L \ d \ \theta]^T \in \mathbb{R}^{4 \times 1}$  is an observation vector;  $x_L$  is the average length of segments A and B;  $d$  is the distance between segments A and B;  $\theta$  is the slope angle of a parking space's entrance (see Fig. 9(b));  $\mathbf{z} \in \mathcal{A}$  is a set to be classified according to the five categories described above.

To solve this MAP problem, the likelihood and prior should be known. The likelihood,  $P(\mathbf{x}|\mathbf{z})$ , is determined by a probability density function taken as a multivariate Gaussian function. A Gaussian kernel is known to be effective in ellipsoidal clustering [22]. In this case, the observation vector has two dimensions, and the equation is as follows [21]:

$$P(\mathbf{x}|z_n) = P(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \quad (11)$$

Here,  $\boldsymbol{\mu} = E[\mathbf{x}] \in \mathbb{R}^{4 \times 1}$  is a mean vector, and  $\boldsymbol{\Sigma} = \text{cov}[\mathbf{x}] \in \mathbb{R}^{4 \times 4}$  is the covariance matrix. To minimize numerical errors, we substitute (11) for (10), take the natural logarithm, and exclude the constant. A final decision function is then arrived at, as follows:

$$\begin{aligned} \hat{\mathcal{D}}_{\text{MAP}} &= \arg \max_{z_n \in \mathcal{A}} \ln [P(\mathbf{x}|\mathbf{z})P(\mathbf{z})] \\ &\cong \arg \max_{z_n \in \mathcal{A}} \left[ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) + \ln P(\mathbf{z}) - \frac{1}{2} \ln |\boldsymbol{\Sigma}| \right]. \end{aligned} \quad (12)$$

Here,  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  can be readily gained from the learning data. In contrast, it is almost impossible to find prior  $P(\mathbf{z})$  in the general case. However, it is possible to infer  $P(\mathbf{z})$  because parking spaces are defined by type. Initially,  $P(\mathbf{z})$  sets a normalized vector with an equivalent probability if the nearest neighbor space is of the same type; then  $P(\mathbf{z})$  is increased by a certain

interval (for example, 0.1). Lastly,  $P(\mathbf{z})$  is normalized.

When a parking space is recognized, a final decision is carried out on the basis of the height information in the space. If there is an obstacle in the recognized parking space, then the space is determined to be unavailable. To increase the reliability of recognition, the updating of (12) and the decision process are repeated every frame to correct a wrong recognition result.

## VII. Experimental Experience

### 1. Experimental Environment

The AVP vehicle adopted in this research uses a 2.6 GHz Intel Xeon E5-2670 as the main controller and was remodeled as an unmanned unit. The test vehicle is equipped with four 1.3 megapixel Pointgrey blackfly cameras with fisheye lenses. The cameras were installed in the front, rear, left, and right sides of the vehicle (see Fig. 2(a)). All algorithms were optimized using C/C++ and an Intel single instruction multiple data intrinsic functions. As shown in Fig. 10, to verify the results, various scenarios were collected into a database containing information on 4,024 parking spaces, stored, and synchronized with in-vehicle network data. The vehicle ran freely up to 30 km/h in various types of indoor and outdoor parking spaces. The outdoor environment included various weather conditions (sunny, cloudy, rainy, and snowy). Also, it was confirmed to have successfully implemented AVP services and that the unmanned vehicle performs parking space navigation, driving, and parking through a combination of proposed algorithms and AVP control technology [3].

### 2. Performance Evaluation

In this experiment, the IPI size used was  $356 \times 800$ ; this was



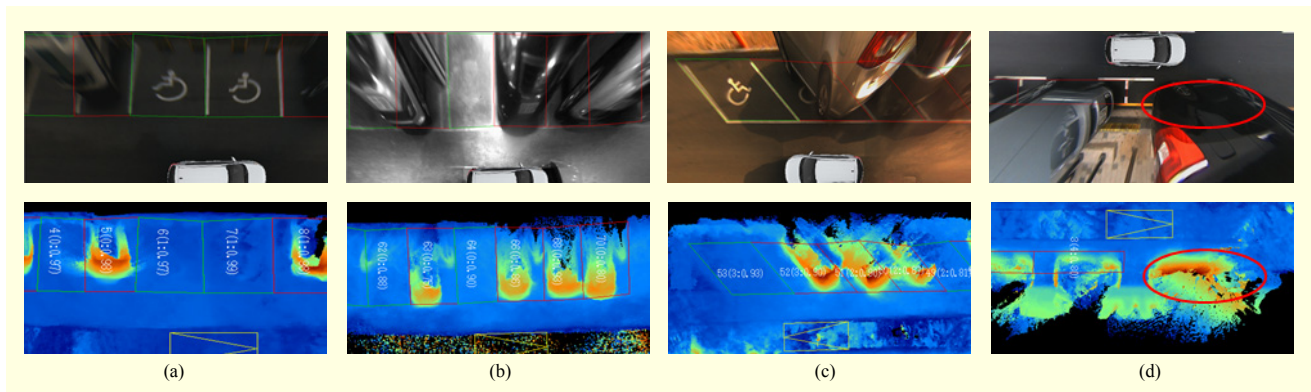


Fig. 10. Sampled database and results: upper row is input image and lower row is recognition results overlapped by HM. Red rectangles are occupied slots, green rectangles are vacant slots, and text in slots is slot ID, type (0: perpendicular, 1: perpendicular-disabled, 2: angled, 3: angled-disabled, 4: parallel), and its probability. (a) Snowy day with low contrast, (b) indoor parking lot with heavy reflection, (c) angled slots at sunset, and (d) parallel slots on rainy day, if parked vehicle covers slot markings (as in red circle), then proposed algorithm could not recognize parking space.

resampled by half for the building of the HM. The main evaluation criteria are the recall and precision [21]:

$$\text{Recall} = \frac{t_p}{t_p + f_n} \times 100, \quad \text{Precision} = \frac{t_p}{t_p + f_p} \times 100. \quad (13)$$

Here,  $t_p$  denotes “true positive,”  $f_n$  “false negative,” and  $f_p$  “false positive.” Table 2 shows the region separation performance of various disparity estimation algorithms. The segmentation results are compared with the manual ground truth data. For comparison, SNCC [23], HEBF [24], and ELAS [25] were selected from the local optimization algorithms. HEBF was originally a GPU-based algorithm, but it was implemented for use by a CPU. As the results show, ELAS demonstrated the best performance because ELAS can more effectively handle the non-textured areas.

Table 3 shows the parking space marking detection performance. The ground truth, as shown in Fig. 7(b), was detected using S3R [20] and then retouched. The proposed technique was compared with state-of-the-art road marking detectors, an SLT [26], and the newest ESLT and S3R detectors

[20]. As the results show, the proposed detector demonstrated the best performance, thanks to robust salient-line detection

Table 2. Performance of static obstacle detection.

Algorithm	Road surface	Low obstacles	High obstacles	Execution time (ms)
SNCC [23]	68.1	65.1	87.1	55.8
HEBF [24]	78.8	77.1	90.5	63.2*
ELAS [25]	95.8	88.5	91.2	53.24

\*CPU version, GPU version is reported to be less than 6 ms [24].

Table 3. Performance of parking space marking detection.

Algorithm	Recall (%)	Precision (%)	Execution time (ms)
SLT [26]	75.8	72.1	0.5
ESLT [20]	85.3	78.8	0.7
S3R [20]	90.1	82.5	4.6
Ours	100	92.6	5.2

Table 4. Evaluations of parking space recognition.

	Perpendicular				Perpendicular-disabled				Angled		Angled-disabled		Parallel			Indoor	Total
	Sunny	Cloudy	Rainy	Snowy	Sunny	Cloudy	Rainy	Snowy	Sunny	Cloudy	Sunny	Cloudy	Sunny	Cloudy	Rainy		
TP	1,452	1,230	292	182	92	30	20	13	223	82	132	22	35	68	32	82	3,987
FN	2	0	0	0	1	0	0	0	4	2	0	0	8	10	9	1	37
FP	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2
Recall (%)	99.86	100	100	100	98.92	100	100	100	98.23	97.61	100	100	81.40	87.18	78.05	98.80	99.08
Precision (%)	100	100	100	100	100	100	100	100	100	100	100	100	97.22	100	96.97	100	99.95

Table 5. Evaluations of execution time.

Algorithm	Step	Execution time (ms)	Total (ms)
HM building*	Stereo rectification	3.01	58.69
	Disparity estimation	53.24	
	Grid mapping	1.55	
	Map fusion	0.89	
Line feature extraction	Pill-box convolution	1.73	5.71
	MEF	0.54	
	Radon transform	1.47	
	Local maxima detection	0.85	
	Straight-line fitting	1.11	
Parking space recognition	Map building	0.21	0.42
	Bayesian classification	0.05	
	Decision process	0.16	
N/A	Other processing	N/A	3–4

\*HM building is running in background thread.

and probability-based road-marking selection technology.

Table 4 shows the results of the parking space recognition performance in a variety of environments. Of the 4,024 parking spaces used in this study, recognition failure occurred in 37 cases and incorrect recognition in only 2. As shown in Fig. 10(d), recognition failure occurred only when a parked vehicle covered the space markings; however, there was no recognition failure if the space was empty. The proposed technology is not affected by adverse environmental conditions, as shown in the results. Therefore, these algorithms are considered to be appropriate to be used as a recognition technique for AVP systems.

Lastly, the evaluation results of the execution time in each step are shown in Table 5. Each execution time was a result of a single core implementation. All algorithms are guaranteed real-time execution except the HM building process, which is run on a background thread. In actual vehicle experiments, it was confirmed that the system successfully operates up to 30 f/s capture rate and 50 km/h velocity.

## VIII. Conclusion

As the competition in autonomous vehicle technology development becomes more and more fierce, autonomous valet parking (AVP) technology is expected to be commonly used in the near future. This study proposes an efficient parking space recognition technology that can be applied directly to an AVP system. In particular, this paper proposes new algorithms consisting of space analysis, road marking detection, and

parking space recognition that combines space and road marking information. Also, the algorithm performance and usability were evaluated in various ways. More research is needed to develop a symbol recognition technology for the disabled parking space and a map optimization technology to improve the accuracy of both the SPM and HM maps. Because the proposed algorithms are the foundation techniques for autonomous vehicles, the proposed techniques can be applied to various other areas.

## References

- [1] G Meyer and S. Deix “Research and Innovation for Automated Driving in Germany and Europe,” in *Road Vehicle Autom.*, Berlin, Germany: Springer, 2014, pp. 71–81.
- [2] B.W. Smith, “Automated Vehicles are Probably Legal in the United States,” Preprint, submitted Aug. 1, 2013. <http://ssrn.com/abstract=2303904>
- [3] K.W. Min and J. Choi, “A Control System for Autonomous Vehicle Valet Parking,” *Int. Conf. Contr., Autom. Syst.*, Gwangju, Rep. of Korea, Oct. 20–23, 2013, pp. 1714–1717.
- [4] W. Park et al., “Parking Space Detection Using Ultrasonic Sensor in Parking Assistance System,” *IEEE Intell. Veh. Symp.*, Eindhoven, Netherlands, June 4–6, 2008, pp. 1039–1044.
- [5] S.H. Jeong et al., “Low Cost Design of Parallel Parking Assist System Based on an Ultrasonic Sensor,” *Int. J. Autom. Technol.*, vol. 11, no. 3, June 2010, pp. 409–416.
- [6] H.G. Jung et al., “Scanning Laser Radar-Based Target Position Designation for Parking Aid System,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, Sept. 2008, pp. 406–424.
- [7] J. Zhou, L.E. Navarro-Serment, and M. Hebert, “Detection of Parking Spots Using 2D Range Data,” *IEEE Int. Conf. Intell. Transp. Syst.*, Anchorage, AK, USA, Sept. 16–19, 2012, pp. 1280–1287.
- [8] S. Gömer and H. Rohling, “Parking Lot Detection with 24 GHz Radar Sensor,” *Int. Workshop Intell. Transp.*, Hamburg, Germany, Mar. 14, 2006, pp. 1–6.
- [9] M.R. Schmid et al., “Parking Space Detection with Hierarchical Dynamic Occupancy Grids,” *IEEE Intell. Veh. Symp.*, Baden-Baden, Germany, June 5–9, 2011, pp. 254–259.
- [10] N. Kaempchen, U. Franke, and R. Ott, “Stereo Vision Based Pose Estimation of Parking Lots Using 3D Vehicle Models,” *IEEE Intell. Veh. Symp.*, Versailles, France, June 17–21, 2002, pp. 459–464.
- [11] J.K. Suhr et al., “Automatic Free Parking Space Detection by Using Motion Stereo-Based 3D Reconstruction,” *Mach. Vis. Appl.*, vol. 21, no. 2, Feb. 2010, pp. 163–176.
- [12] C. Unger, E. Wahl, and S. Ilic, “Parking Assistance Using Dense Motion-Stereo,” *Mach. Vis. Appl.*, Apr. 2014, vol. 25, no 3, pp. 561–581.

- [13] H.G. Jung et al., "Parking Slot Markings Recognition for Automatic Parking Assist System," *IEEE Intell. Veh. Symp.*, Tokyo, Japan, June 13–15, 2006, pp. 106–113.
- [14] C. Wang et al., "Automatic Parking Based on a Bird's Eye View Vision System," *Adv. Mech. Eng.*, vol. 2014, 2014, pp. 1–3.
- [15] J.K. Suhr and H.G. Jung, "Full-Automatic Recognition of Various Parking Slot Markings Using a Hierarchical Tree Structure," *Opt. Eng.*, vol. 52, no. 3, Mar. 2013, pp. 1–14.
- [16] 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, Feb. 2014, pp. 21–36.
- [17] S.-J. Han, J. Kim, and J. Choi, "Effective Height-Grid Map Building Using Inverse Perspective Image," *IEEE Intell. Veh. Symp.*, Seoul, Rep. of Korea, June 28–July 1, 2015, pp. 549–554.
- [18] E. Elboher and M. Werman, "Efficient and Accurate Gaussian Image Filtering Using Running Sums," *Intell. Sys. Des. Appl.*, Kochi, India, Nov. 27–29, 2012, pp. 897–902.
- [19] L. Gorman, M. Sammon, and M. Seul, "Practical Algorithms for Image Analysis," 2nd Ed., New York, USA: Cambridge University Press, 2008, pp. 118–125.
- [20] S.-J. Han and J. Choi, "Real-Time Precision Vehicle Localization Using Numerical Maps," *ETRI J.*, vol. 36, no. 6, Dec. 2014, pp. 968–978.
- [21] K.P. Murphy, "Machine Learning: A Probabilistic Perspective," Cambridge, MA, USA: MIT Press, 2012, pp. 151–186.
- [22] H.S. Lee, J.H. Yoo, and D.H. Park, "Data Clustering Method Using a Modified Gaussian Kernel Metric and Kernel PCA," *ETRI J.*, vol. 36, no. 3, June 2014, pp. 333–342.
- [23] N. Einecke and J. Eggert, "A Two-Stage Correlation Method for Stereoscopic Depth Estimation," *IEEE Int. Conf. Digital Image Comput. Techn. Appl.*, Sydney, Australia, Dec. 1–3, 2010, pp. 227–234.
- [24] Q. Yang, "Hardware-Efficient Bilateral Filtering for Stereo Matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, May 2014, pp. 1026–1032.
- [25] A. Geiger, M. Roser, and R. Urtasun, "Efficient Large-Scale Stereo Matching," *Asian Conf. Comput. Vis.*, Queenstown, New Zealand, Nov. 8–12, 2010, pp. 25–38.
- [26] T. Veit et al., "Evaluation of Road Marking Feature Extraction," *IEEE Int. Conf. Intel. Transp. Syst.*, Beijing, China, Oct. 12–15, 2008, pp. 174–181.



**Seung-Jun Han** received his BEng degree in control and instrumentation engineering from Pukyong National University, Busan, Rep. of Korea, in 1998. He received his MS degree in electronics engineering from Pusan National University, Rep. of Korea, in 2000. From 2000 to 2010, he worked as a principal researcher for Sane System Co., Ltd., Anyang, Rep. of Korea. In 2011, he worked as a researcher at the Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea. Since 2012, he has been working as a senior researcher at ETRI. In addition, he has won several awards, including the 6th Samsung Humantech Thesis prize in 2000, the outstanding paper award at the 17th ITSWC in 2010, and the outstanding employee award at ETRI in 2014. His main research interests are structure from motion; simultaneous localization and mapping; and machine learning.



**Jeongdan Choi** received her PhD in image processing from Chungnam National University, Daejeon, Rep. of Korea, in 2006 and her BS and MS degrees in computer graphics from Chung-Ang University, Seoul, Rep. of Korea, in 1993 and 1995, respectively. Since 1995, she has been working as a principal researcher at ETRI. She is also a director of the Autonomous Vehicle Infrastructure Research Section, Smart Mobility Research Department, ETRI. Her research interests are in automotive image processing and recognition; 3D modeling; and rendering.