Available Parking Slot Recognition based on Slot Context Analysis

Soomok Lee [1] and Seung-Woo Seo [2,*]

[1,2] Department of Electrical and Computer Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea.
[*]sseo@snu.ac.kr

Abstract: Vacant parking-slot recognition is one of the core elements for achieving a fully automated parking assistant system. The recognition of a vacant parking slot is typically executed in two stages: parking-slot recognition and slot-occupancy classification. In most previous works, considering the recognised slot results as a grid cell, the existence of objects is evaluated through the slot examination in slot-occupancy classification. Despite moderate performance, however, these methods cannot distinguish the parking-slot availability of moving or small objects and permit marks, since all previous methods only focus on the occupancy of typically parked vehicles. In this paper, we present a camera-based available parking-slot recognition algorithm based on slot context analysis. By utilising abundant visual features, we propose a new algorithm handling diverse available parking-slot conditions. For parking-slot recognition, we propose a new methodology of extracting and associating line markings. The proposed algorithm includes a slot-validation step that identifies multiple slot contexts in a probabilistic integration and has more flexibility on irregular patterns. In the slot-occupancy classification stage, reliable parking-availability detection is achieved through visual slot features, including the Histogram of Gradient and frequency-magnitude features, via a Support Vector Machine. Simulations and vehicle-level experiments demonstrate the robustness of the proposed algorithm in diverse conditions.

1. Introduction

So far, a large number of automated parking assistant systems (PASs) have been proposed to maximise both the safety and convenience of parking. The selection of a particular parking-slot among those available is a primary component in PASs. PASs have mainly been developed in three ways based on their scope: infrastructure-based methods, semi-automatic methods and fully automatic methods. The infrastructure-based methods suggest the utilisation of a prebuilt map and infra-level sensors [12][20]. Desirable parking spaces are selected based on the infrastructure, and the vehicle receives the information through vehicle-to-infrastructure (V2I) communication. Despite the methods having an advantage of managing all parking-slots, they are confined to parking lots with a pre-built infrastructure. On the other hand, in semi-automatic methods, the selection of a desirable parking space among available parking spaces is performed by a human driver via a touch screen [1]-[5]. Although this method is simple and reliable, it has limitations in being combined with fully autonomous driving technology. In fully automatic methods, however, a desirable parking space is selected automatically by PASs. Fully-automatic methods can be categorised into two types: free-space-based approaches [17]-[23] and marking-based approaches [1]-[11].

In free-space-based approaches, different types of sensors with different capabilities (e.g. laser scanners, ultrasonic sensors and depth cameras) have typically been employed. Except for visual-type sensors, these methods are highly dependent on the existence and position of adjacent vehicles [17]-[19]. For visual-type sensors [21]-[23], however, it is difficult to attain a real-time algorithm due to heavy computational load for depth-map generation. On the other hand, in the marking-based approach, slot markings are detected using visual sensors. By detecting the parking-lines of possible parking-slots, an accurate position and heading angle are provided. With this information, parking becomes more accurate within an available parking space. For these reasons, the marking-based approach has been the focus of recent research and development. In addition, the marking-based approach has become more influential in the fully-automatic method, since a wider field of view (FOV) has become available with multiple visual-sensor configurations, such as around-view monitoring systems (AVM), in recent years.

1.1. Motivation

Despite the benefits, the marking-based approach has one problem. Since marking-based slot recognition relies entirely on line marks, some occupied slots can be recognised as available parking-slots when they are actually unavailable for parking manoeuvring. Therefore, slot-occupancy classification is imperative for achieving a full-automatic parking system. In certain previous studies [8][9], ultrasonic sensors are jointly employed to determine slot occupancy. Meanwhile, other studies [28][29] have adopted a camera-based method for the entire process. However, all previous studies rely on feature occupancy ratio and overlook the existence of informative inner-marks or small objects, such as traffic cones, waste bags and pedestrians. To avoid such limitations, the proposed algorithm analyses multiple aspects of slot context, including appearance and frequency-based features, in the camera-only-based method. With slot context analysis, diverse slot availability conditions, such as vehicles, pedestrians, traffic cones and permit marks, can potentially be recognised.

1.2. Prior Work

The vacant parking-slot detection algorithm [8] leads to the following two main procedures: parking-slot recognition and slot-occupancy classification of the detected parking-slot. Concerning the first procedure, a variety of parking-slot detection methods has been proposed. Parking-line detection methods were initially proposed for slot recognition. For instance, colour segmentation [1] and line-edge detection [2]-[4], including the Hough transform [2][3], were proposed to extract parking-lines. Although these methods detect parking-lines moderately well, they were rather ineffective regarding shadows or faint lines and computationally heavy. More importantly, these parking-line detection methods considered neither multiple slot cases nor diverse slot patterns. Most recently, *Ho-gi et al.* [4] adopted pattern recognition steps to classify various slot patterns, such as open rectangular, diamond and slanted rectangular patterns. These patterns are recognised by a neural network with the data of predefined junction-pattern templates. *Jae-kyu et al.* [6]-[8] improved the pattern recognition format through a hierarchical tree-structure classifier. The classifier investigates the extracted Harris corners according to the hierarchical steps of corner angle, junction type and slot pattern. Despite decent performances, these approaches have numerous limitations. First, the slot recognition algorithm can malfunction if a new target does not belong to the predefined patterns. Second, the utilisation of corner features cannot always provide precise target positions and heading angles. Lastly, the algorithm has limitations due to excessive constraints in scenarios when two or more slots with an identical slot pattern need to be positioned or the entering side of slots must visible.

In terms of the second procedure, slot-occupancy classification is performed by two methods: ultrasonic fusion methods and image analysis methods. First, occupancy classification has been attempted by combining ultrasonic sensors. To avoid unreached regions caused by short-range measurement, accumulated sequential ultrasonic measurements are mapped towards the slots in the image. The slot occupancy is inferred by statistical distributions of accumulated points [8]-[9]. In these studies, researchers accumulate the ultrasonic measured points with certain period of times to compensate for sensor measurements' sparsity and short range. However, this approach has the following limitations. This inference model is improper for small objects, whereas moving objects are not properly analysed. Regarding moving objects, for example, if a pedestrian walks near the ego-vehicle and on the parking-slots, the methods have the potential to accumulate all the traces of the pedestrian and wrongly recognise the slots as occupied. Second, image analysis methods utilise edge features restrictively. *Wang et al.* [28] analyses the occupancy of the front slot using Sobel-edge features, whereas *Houben et al.* [29] adapts linear discriminant analysis (LDA) to determine slot occupancy based on Difference of Gaussian (DoG) edge distributions.

1.3. Our Contribution

In this paper, we present a camera-based algorithm for available parking-slot recognition based on visual slot context analysis. We seamlessly employ visual analysis throughout the procedures: both parking-slot recognition and slot-occupancy classification (Fig.1). First, the parking-slot recognition stage generates all possible parking-slots based on visible parking-lines. This stage consists of parking-line detection through marking extraction and clustering, slot-candidate generation with the detected lines and slot validation through slot context evaluation in a Bayesian Network (BN) framework. Second, the slot occupancy classification stage verifies the parking-slot availability based on the detected slots. For each detected slot, visual slot context information is collected by extracting a Histogram of Gradient (HOG) and frequency magnitude feature of a regularised slot representation. Through these visual features, each detected slot is classified as an un/available slot by the Support Vector Machine (SVM).

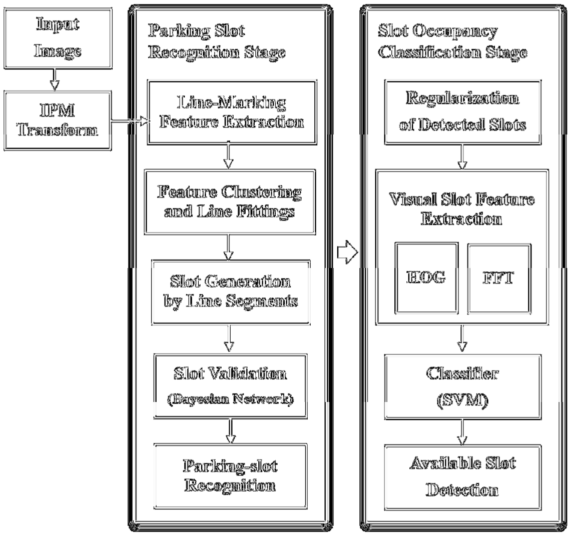

Input Image
IPM Transform
Parking Slot Recognition Stage
Line-Marking Feature Extraction
Feature Clustering and Line Fittings
Slot Generation by Line Segments
Slot Validation (Bayesian Network)
Parking-slot Recognition
Slot Occupancy Classification Stage
Regularization of Detected Slots
Visual Slot Feature Extraction
HOG
FFT
Classifier (SVM)
Available Slot Detection

Fig.1. *Two stage framework for available parking space recognition.*

The main contributions of this study can be summarized as follows:

- We propose the cone-hat marking extraction filter and entropy-based line-marking clustering for accurate parking-line detection. The cone-hat filter extracts line features robustly even in deteriorated visibility. Afterwards, the proposed entropy-based line-marking clustering assigns the extracted line marks to each parking-line segment rapidly and effectively. Compared with other line detectors, including the Hough transform, the proposed detection method is faster and guarantees more robust detection in the case of distorted or faint parking-lines. In addition, compared with corner detector-based methods, detected parking-lines provide more precise target positions, heading angles and occluded corner estimation.
- To improve slot recognition performance, we propose a slot validation step through slot context analysis based on the BN model theory. In slot validation, the proposed BN model evaluates whether the generated slot-candidates are properly formed by lines (e.g. line parallelism, slot angle and width). The model removes the cases of improperly generated slots accompanied by road markings or objects. Since these multiple slot-context cues are analysed in a probabilistic integration, the algorithm shows robustness in various slot conditions. Compared with conventional slot-pattern classification methods [6]-[9] with predefined slot profiles, our recognition can have more flexibility for coping with unexpected slot patterns, since our validation model is not confined to a particular predefined threshold.
- We illuminate parking-slot availability classification to solve diverse occupancy conditions by proposing visual analysis, including HOG and frequency magnitude features. We additionally identify parking availability with small objects, movable objects and parking-related marks.

Additionally, this visual analysis slightly reduces the false-positive slot recognition results that occur in the previous stage. This results from the proposed features, which also evaluate the border of the slot where line markings are normally positioned.

1.4. Paper Organization

The remainder of this paper is organised as follows. Section II presents the parking-slot recognition stage, which is comprised of line-segment detection, slot-candidate generation and validation. The parking-slot availability detection stage, which consists of visual slot transformation, slot feature extraction and classification, is introduced in Section III. Section IV provides the experimental results and an evaluation of the proposed algorithm. Section V draws conclusions.

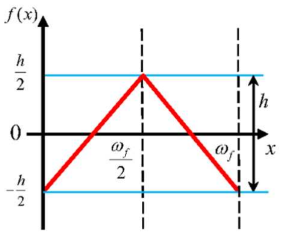
2. Parking-Slot Recognition

This section presents a novel parking-slot recognition method. Accurate parking-slot recognition is utilised not only to localise an automated vehicle directly using the slot position, but also to perform as input data for slot occupancy classification in the next stage. The parking-slot recognition algorithm consists of five steps: line-marking extraction using cone-hat filter, line-marking clustering using entropy-based clustering, line-segment detection, slot-candidate generation and slot verification. For inputs for the procedure, we utilise a transformed bird's-eye-view (BEV) image [26] to remove perspective effects assuming that the cameras are calibrated properly.

2.1. Parking-Line-Marking Feature Extraction Using a Cone-hat Filter

In parking-slot recognition, a robust line-marking extraction filter is a prerequisite for finding line segments. Conventional popular filters, such as the ridge, Gaussian, or edge filter, cannot cover all the harsh conditions of shadows, dirt or cracks on the road. Therefore, we propose a powerful filter named the cone-hat filter for line-marking extraction. The filter is defined as

$$f(x)=\begin{cases} -h/2+\dfrac{2h}{\omega_f-1}x & if\,(x\le \omega_f/2) \\ h/2-\dfrac{2h(x-(\omega_f-1)/2)}{\omega_f-1} & otherwise. \end{cases} \tag{1}$$



a

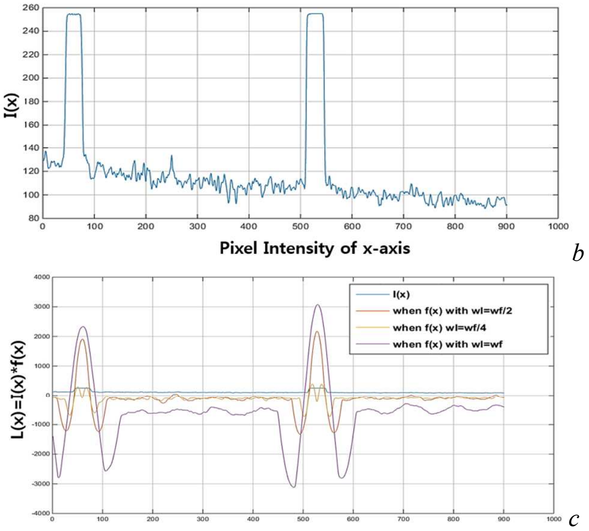


Fig.2. *Cone-hat Filter Response.*
a Shape of cone-hat filter
b Intensity values of x-axis on the image,
c Filter convoluted responses of cone-hat (Fig.2-a) with respect to varying filter sizes.

The cone-hat filter illustrated in Fig.2-a is a symmetrical filter whose width is designed to be twice that of the line-marking width. The filter shape gives more weight to the centre point than the side points and is computed with respect to one direction on the x-axis of the vertical lines and on the y-axis of the horizontal lines. Regarding line-marking extraction on the x-axis, the filter is convoluted to the image as

$$L[x] = I[x] * f[x]$$
$$(I * f)[n] = \sum^{img.width} I[n-m] f[m], \tag{2}$$

where $I(x)$ is the intensity of the image pixels. $f(x)$ and $L(x)$ indicate the filter and response function, respectively. The ridge points of the line-marking are shown effectively by the filter response. By inspecting the convoluted output, we collect the line-marking features as

$$x = (l_{ridge} + r_{ridge})/2, \quad \begin{cases} l_{ridge} & if \quad L[x]=0, \nabla_x L[x] > 0 \\ r_{ridge} & if \quad L[x]=0, \nabla_x L[x] < 0 \end{cases}, \quad \nabla_x L[x] = L[x] - L[x-1], \tag{3}$$

where a point x indicates the middle of the ridge points. To obtain the optimal filter, the filter width should be twice that of the line-marking width, while the filter magnitude does not influence the extraction performance, since it only changes the magnitude of the response. Compared with the Laplacian of Gaussian (LoG) filter's shape, the cone-hat filter has the highest and lowest symmetric values with a linear-polynomial shape and do not converge to zero when x goes infinite.

Fig.2-c illustrates the filter responses with varying filter widths. Even if the width is designed as slightly narrower or wider than the line markings, the filter still provides the exact points until the filter width is between half and twice that of the optimal filter size. This characteristic stabilise line-marking extraction and decreases dependency on the prior knowledge of accurate line-marking width.

2.2. Parking-Line-Marking Clustering

Proper clustering assigns the extracted marking features to each parking-line segment. There exist two challenging issues in clustering. First, there exist numerous outliers, such as symbolic road markings, white parts of the objects, or light reflections. Second, there are an unknown number of total line segments of multiple slots. The employment of popular clustering methods is not effective, since K-means clustering and hierarchical clustering need a predefined number of classes, whereas EM clustering is computationally heavy. Meanwhile, the proposed entropy-based marking-feature clustering is designed to fit in parking-slots' geometric characteristics to attain cost-efficient and decent clustering performance. For instance, the lateral lines are distinctively parallel, since the parking-slots are possibly positioned in a row.

In our approach, the clustering of vertical and horizontal line features is calculated independently. With the assumption of parallelism, we find an angle that is orthogonal to all parallel lines. The angle is defined as the "orientation difference." When the marking features are transformed by the acquired orientation difference, the feature dimension is reduced to one axis. This downsizes the algorithm complexity from $O(n^2)$ to $O(n)$. The orientation difference θ^* is obtained by rotations of the "projection line" l_θ from the image centre. Let $\Im_\theta(x', y') = I(x\cos\theta, y\cos\theta)$ be the transformed marking features by θ rotation. Let ε_θ be the entropy of $\Im_\theta$, which is computed using its histogram H_α. H_α is a histogram of accumulated marking features onto l_θ. The results of the energy function are obtained as

$$\varepsilon_\theta = -\sum_{i \in L} H_\theta(\Im)\log(H_\theta(\Im)), \tag{4}$$

where L is the number of bins of H_α. Then, the distribution of entropy values is obtained by varying the values of θ. In comparison with other projection lines with varying angles, the optimal gradient θ^*, which minimises the energy function, is acquired as

$$\theta^* = \arg\min_\theta \varepsilon_\theta = \arg\max_\theta \sum\nolimits_i^L H_\theta(i)\log(H_\theta(i)). \tag{5}$$

Fig.3-a depicts the feature accumulation histogram with varying l_θ to find the orientation difference of vertical line features. The feature histogram of $\theta = 5^\circ$ has more entropy than that of $\theta = -8^\circ$. Therefore, the projection line with $\theta = 5^\circ$ is closer to θ^*. Since the heading angle of the ego-vehicle is rarely changed dramatically in a short period, the orientation difference is updated intermittently. At the moment, the angle is only estimated with yaw angle. When updating θ^*_{t+1} with measurements, the scope of the orientation-difference calculation is confined around the previous angle, θ^*_t.

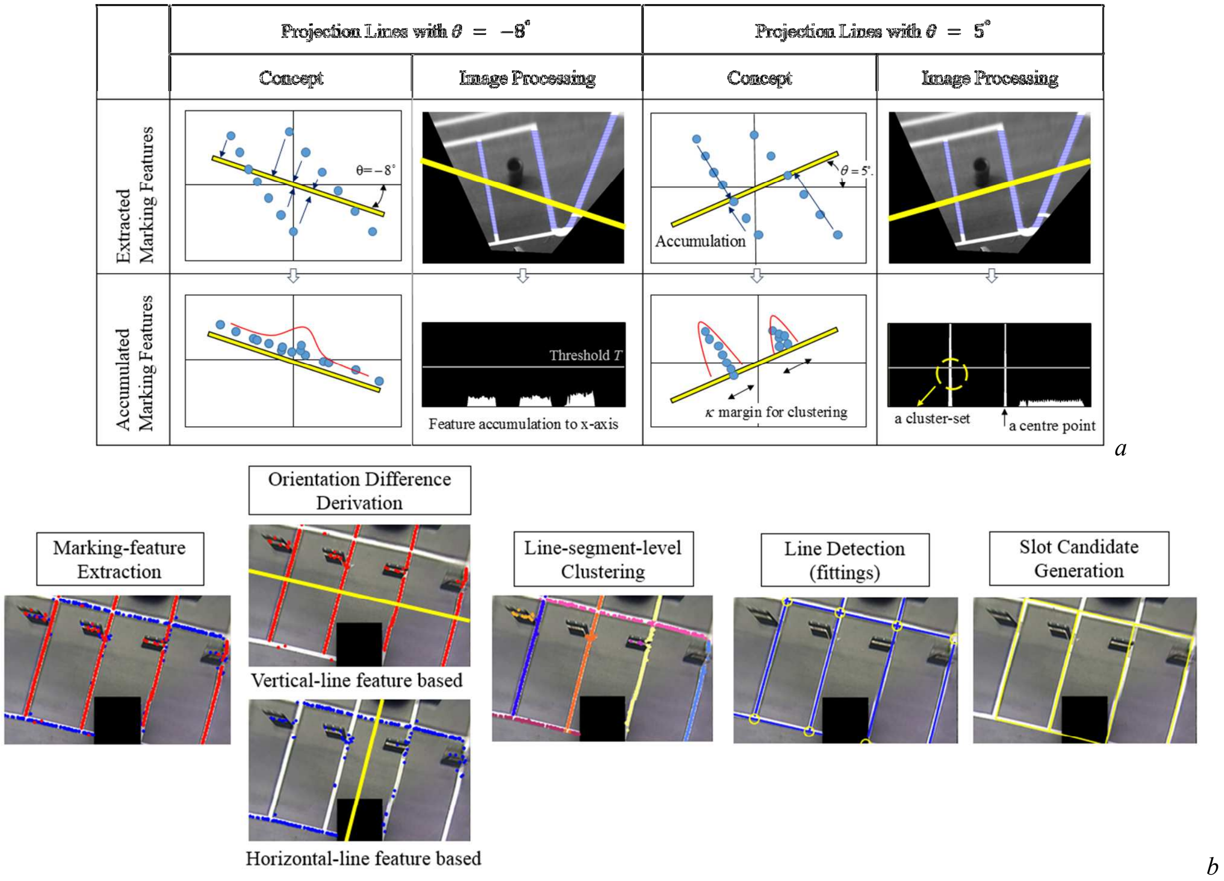

Projection Lines with θ = -8°
Projection Lines with θ = 5°
Concept
Image Processing
Concept
Image Processing
Extracted Marking Features
Accumulated Marking Features
θ=−8°
θ = 5°
Accumulation
Threshold T
Feature accumulation to x-axis
κ margin for clustering
a cluster-set
a centre point
a
Orientation Difference Derivation
Marking-feature Extraction
Line-segment-level Clustering
Line Detection (fittings)
Slot Candidate Generation
Vertical-line feature based
Horizontal-line feature based
b

Fig.3. *Parking-slot recognition procedures*
a Orientation difference derivation for the clustering method: feature (blue colour) projections to varying projection lines (yellow lines).
b Line-marking extraction (first), orientation difference derivations of two axes respectively (second), clustering marks (different colours) with respect to each parking-line group (third), parking-line detection using marking features (fourth) and slot-candidate generations (fifth).

Once the line feature set is projected to the projection line with respect to θ^*, the feature space considered is now only a histogram with a projected x-axis. The local maxima above certain threshold T of the projected histogram are considered to be the centre points of the clusters. The features, adjacent to a centre point within error margin κ, are collected into each parking-line group. The number of centre point considered is up to N_S, which is empirically set regarding the FOV. This procedure performs each line, vertical lines and horizontal lines respectively.

2.3. Parking-Line Detection and Parking-Slot Candidate Generation

Acquiring accurate parking-line information is important in that it provides the target heading angle, lateral offsets, and slot region. In the previous step, line-marking features are clustered into the line-segment level without meaningful information. We utilise Sequential-RANSAC [13] to extract parking-lines from individual marking feature clusters (Fig.3-b, third and fourth figures). The RANSAC model is a first-order polynomial, $f(x) = C_0 + C_1 x$, which gives the lateral offset C_0 and the heading angle $\tan^{-1} C_1$. In addition to line model parameters, each parking-line tip is collected.

Two or more line segments can be obtained for parking-slots. Slot-candidates are generated when horizontal and vertical lines exist and form convex polygons (Fig.3-b, fifth figure). In the case of open-type slots without entrance line, the entrance line is substituted by a hypothetical line by connecting the tips of the lateral lines and these tips are performed as crossing points. The minimum constraint for slot-candidate generation is the existence of two crossing points with lateral lines. If either the entrance or end-side slot line is invisible due to the FOV or occlusion, the invisible lines are approximated based on predefined geometry information, such as the width and height of the slot. With estimated or hypothetical line, four crossing points of the generated slot-candidate are always collected regardless of the line combination.

2.4. Slot Validation Using Slot Context

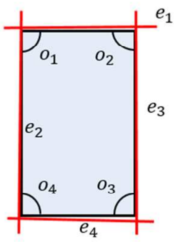
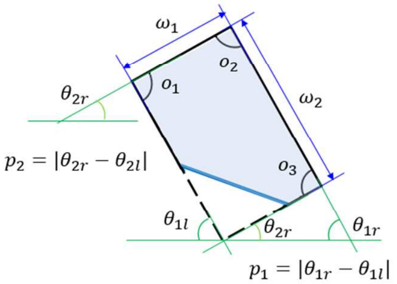
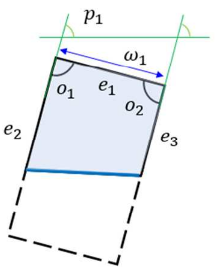
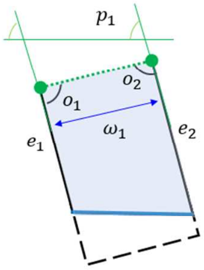
Every generated slot-candidate must be validated, since it is just a composition of adjacent line segments. To achieve slot validation, we adopt the BN model to analyse the geometric context of the generated slot-candidate. BN is one of the probabilistic graphical models that represent the mathematical principles of graph and probabilistic theory [27]. With the form of a directed acyclic graph $G = (V, E)$, a node implies that each random variable containing an observable factor and an edge represents a conditional probability of a parent node.

The proposed BN model contains observation nodes $\Im$ with a variety of geometric contexts such as corner incline O, line parallelism P, line-fitting error E and slot width W. Each observation node consists of individual observations. For example, O contains $o_1, ..., o_{N_O}$ where N_O represents the number of observed nodes. The numbers, N_O, N_P, N_E and N_W, can be changed according to the slot's FOV. Examples of node factor derivation are shown in the figure at Table 1. The corner incline o_i is obtained from the visible internal angles of the square. In the open-type case, the crossing angles between a hypothetical line and lateral lines are performed as o_i. Additionally, line parallelism p_i is calculated by the absolute difference orientation between facing lines, while the line-fitting error e_i represents the normalised least mean square error between detected lines and marking features. The number N_E of nodes e_i is the same as the number of detected lines (red line in the table). Lastly, slot width W consists of both horizontal ω_1 and vertical width ω_2.

Table 1 Parameter Representation and Examples of Individual Node Factors.

Node Factors	Parameters Representation
A Prior	$s \in \{Yes, No\},\ P(s \mid x) \sim Ber(\xi)$
Corner Incline	$O = \{o_1,..,o_{N_O}\},\ P(o_i \mid s) \sim \mathrm{N}\ (\mu_o, \sigma_o),\ N_O \in \{2,3,4\}$
Line Parallelism	$P = \{p_1,..,p_{N_P}\},\ P(p_i \mid s) \sim \mathrm{N}\ (\mu_p, \sigma_p),\ N_P \in \{1,2\}$
Line-fitting Error	$E = \{e_1,..,e_{N_E}\},\ P(e_i \mid s) \sim \mathrm{N}\ (\mu_e, \sigma_e),\ N_E \in \{2,3,4\}$
Slot Width	$W = \{\omega_1,..,\omega_{N_W}\},\ P(\omega_i \mid s) \sim \mathrm{N}\ (\mu_\omega, \sigma_\omega),\ N_W \in \{1,2\}$

Graphical Examples of Node Evidence

Cases	All Visible Corners	One Invisible Corner	Two Invisible Corners	Open-type
A Generated Slot in the Image	 e_1 o_1 o_2 e_3 e_2 o_4 o_3 e_4	 ω_1 o_2 o_1 θ_{2r} ω_2 $p_2 = \lvert\theta_{2r} - \theta_{2l}\rvert$ o_3 θ_{1l} θ_{2r} θ_{1r} $p_1 = \lvert\theta_{1r} - \theta_{1l}\rvert$	 p_1 ω_1 o_1 e_1 o_2 e_2 e_3	
# of nodes	$N_O = 4,\ N_P = 2,$ $N_E = 4,\ N_W = 2$	$N_O = 3,\ N_P = 2,$ $N_E = 4,\ N_W = 2$	$N_O = 2,\ N_P = 1,$ $N_E = 3,\ N_W = 1$	$N_O = 2,\ N_P = 1,$ $N_E = 2,\ N_W = 1$

▢ Visible Region ━ Visible Boundary ━ Visible Parking-lines ‒ ‒ Invisible Parking-lines ●·····● Hypothetical Line

Using the BN property, all these observation nodes are conditionally independent given the 'slot validity' layout s, which represents a prior. Based on the observation nodes, a prior s is classified whether the generated slot is proper. The joint distribution over observation $\Im$ and a prior is factorised:

$$P(s, \Im \mid \Theta) = p(s, O, P, E, W \mid \Theta) = P(s \mid \Theta) \prod_{i=1}^{N_O} P(o_i \mid s, \Theta) \prod_{i=1}^{N_P} P(p_i \mid s, \Theta) \prod_{i=1}^{N_L} P(e_i \mid s, \Theta) \prod_{i=1}^{N_W} P(\omega_i \mid s, \Theta), \quad (6)$$

where $\Theta = \{\mu_o, \sigma_o, \mu_p, \sigma_p, \mu_e, \sigma_e, \mu_\omega, \sigma_\omega, \xi\}$ represents the set of model parameters, which are shown in Table 1.

To establish our criterion, all parameters should be trained to perform as prior knowledge and dataset D is prepared for a set of n generated slots $X = \{x_1, ..., x_n\}$. For each $x \in X$, the dataset contains slot information $(x, o_x, p_x, e_x, \omega_x)$, i.e., $D = \{(s, o_x, p_x, e_x, \omega_x) \mid x \in X\}$. Each x has hidden variable information $s \in S = \{Yes, No\}$. The distributions of each variable follow Table 1. Given dataset D, our goal is to infer the model parameters by maximising the log-likelihood function (eq. 7).

$$\begin{aligned}
\text{maximise} \quad & \log P(D \mid \Theta) \\
& = \log \prod_{x \in X} P(x) P(\Im \mid x) \\
& = \log \prod_{x \in X} P(x) \prod_{x \in X} \sum_{s \in S} P(\Im \mid s) P(s \mid x) \qquad (7) \\
& = \sum_{x \in X} \log P(x) + \sum_{x \in X} \log \sum_{s \in S} P(\Im \mid s) P(s \mid x) \\
\text{such that} \quad & \sum_{s \in S} p(s \mid x) = 1.
\end{aligned}$$

This equation is rewritten as a Lagrangian equation.

$$L = \sum_{x \in X} \log P(x) + \sum_{x \in X} \log \sum_{s \in S} P(\Im \mid s) P(s \mid x) + \sum_{x \in X} \lambda_x (1 - \sum_{s \in S} P(s \mid x)) \qquad (8)$$

We adopt an expectation and maximisation (EM) algorithm to solve the equation above. In the EM algorithm, the parameters are iteratively calculated until they converge to the upper boundary. The E-step (eq. 9) and M-step (eq. 10) are derived as:

$$P(s \mid \Im, x) = \frac{P(o \mid s) P(p \mid s) P(e \mid s) P(\omega \mid s) P(s \mid x)}{\sum_{s' \in S} P(o \mid s') P(p \mid s') P(e \mid s') P(\omega \mid s') P(s' \mid x)} \qquad (9)$$

$$\begin{aligned}
& P(s \mid x) = P(s \mid \Im, x) = P(s \mid o, p, e, \omega, x) \\
& P(o \mid s) \sim (\mu_o, \sigma_o), \quad \mu_o = \frac{\sum_{x \in X} o \times p(s \mid \Im, x)}{\sum_{x \in X} p(s \mid \Im, x)}, \sigma_o{}^2 = \frac{\sum_{x \in X} (o - \mu_o)^2 p(s \mid \Im, x)}{\sum_{x \in X} p(s \mid \Im, x)} \qquad (10)
\end{aligned}$$

The parameter derivation of other observation nodes, such as $P(p \mid s)$, $P(e \mid s)$ and $P(\omega \mid s)$ in M-step follows the same form of $P(o \mid s)$ since they have the identical distribution. After training, a prior, slot validity, is determined with respect to the parameters when the test data of observations are applied.

3. Slot Occupancy Classification

This section presents slot-occupancy classification for examining the parking availability of detected slots. The slot availability of each detected slot should be tested, since some of the detected slots could be unavailable because of occupied objects or permit marks. This section is comprised of three steps: slot-image regularisation, HOG and frequency magnitude feature extraction and parking availability classification using SVM. We proceed with these steps as long as at least half of the detected slot is visible.

3.1. Regularisation of Detected Parking Slots

In section 2.3, the detected parking-slot always collects four crossing points by the detected or estimated parking-lines even when only two junctions are visible. The crossing points form a square, which is rotated with respect to the vehicle heading and distortion with calibration errors. A regularised form is required to both compensate for these issues and maximise the recognition performance to perform as the basis for adequate feature extraction. The warping technique is adjusted for regularisation. The four crossing points of the recognised slot serve as source points, whereas the destination points are the corners of the designated rectangular feature size (i.e., 64x128 pixels). If the slot has an invisible region, zero intensity is assigned. At the moment, the warping technique is designed to include border slot lines, as shown in Fig.4-a. The regularised images are applied to both training and testing data.

3.2. Visual Slot Feature Extraction

Slot features are used to analyse the slot context and recognise slot occupancy. Two major features are applied: the HOG descriptor and the frequency magnitude feature. First, the HOG descriptor [14][15] is commonly applied for human detection. It is a window-based descriptor that scans all image regions to detect an aiming region. However, in this study, a region of interest is already fixed to the detected slots only; thus, the procedure is much simpler. To compute the HOG descriptor, the image window is divided into small spatial rectangular cells (8x8). Within each cell, a frequency histogram is computed that represents the distribution of edge orientations. The edge orientations are computed as $\arctan(\frac{df}{dV}/\frac{df}{dH})$ and quantised into an n-bin histogram, where $\frac{df}{dV}$ and $\frac{df}{dH}$ indicate the vertical and horizontal gradients, respectively. After that, normalisation can reduce the illumination variability. The combination of every computed histogram is put to a feature vector set.

Second, the frequency magnitude feature is extracted with the same regularised input. The image is transformed by the Fast Fourier transform (FFT), and the magnitude response is derived as follows:

$$F(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)*e^{-j*2\pi(ux+vy)/N}=\mathrm{Re}(F)+j\mathrm{Im}(F)$$
$$Mag(F(u,v))=\sqrt{\mathrm{Re}(F)^2+\mathrm{Im}(F)^2} \tag{11}$$

The magnitude of the frequency response is the main concern, since the response varies with respect to the slot context, as shown in the third row of Fig.4-b. For example, the frequency magnitude of the outer lines is likely to be positioned along the x and y axes, whereas the frequency magnitude of the object is distributed sparsely in an oval shape. To collect a feature vector set from the response, a line rotates with the centre and extracts the intensity distribution of the frequency magnitude. Consequentially, the HOG descriptor is beneficial for conducting the shape analysis of permit marks and object patterns, while the frequency magnitude feature is robust for the analysis of object existence, road and border line markings. By composing these two features in a single feature vector set, individual weaknesses are mutually compensated for.

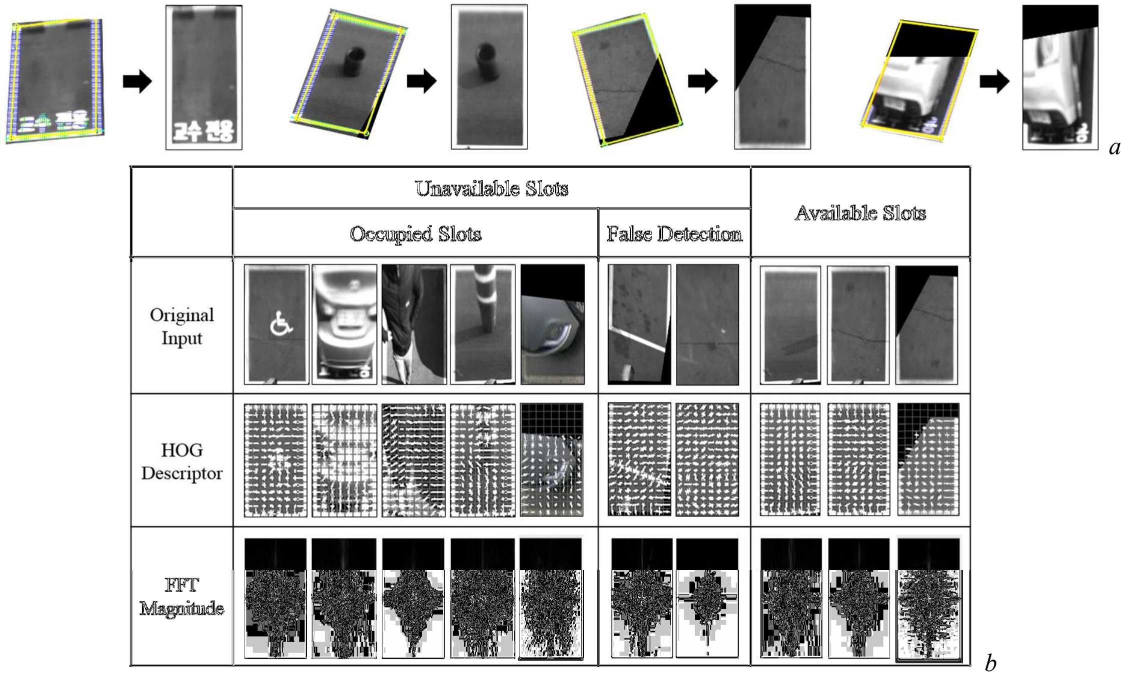


Fig.4. *Visual-slot feature extraction.*
a Image Regularization Transform of the detected parking-slot.
b Examples of visual-slot feature.

3.3. Slot Occupancy Classification by Support Vector Machine

A classifier, which is a perfect match for analysing the visual-slot features, should be selected to evaluate the slot availability. One of the classifiers for superbly analysing the descriptor is SVM. SVM is a discriminative and supervised learning technique for categorising the classes by predefined parameters.

The parameters represent the coefficient of a hyper-plane, which divides the classes in the multi-dimensional (number of feature dimensions) space. These parameters are determined by the training dataset. Given the labelled training data, $(x_i, y_i), i = 1,\dots,N, y_i \in \{-1,1\}, x_i \in \Re^d$, SVM aims to find an decision boundary $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$. The decision boundary is designed to separate the individual classes by maximising a support vector, $\gamma = \frac{1}{\|\mathbf{w}\|_2} = \frac{1}{\mathbf{w}^{\mathrm{T}}\mathbf{w}}$. The optimisation problem of SVM is following the 2-norm soft-margin objective as follows:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimise}} \quad \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}\zeta_i^2 \\ &\text{such that} \quad y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \geq 1 - \zeta_i, \text{i=1}, \dots, N \end{aligned} \tag{12}$$

A dual problem of equation (12) using the Lagrangian equation is derived as

$$\begin{aligned} &\text{maximise} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{n,m} y_n y_m \alpha_n \alpha_m (K(x_n, x_m) + \frac{1}{C}\delta_{n,m}) \\ &\text{such that} \quad \sum_n y_n \alpha_n = 0, \qquad \alpha_n \geq 0 \end{aligned} , \tag{13}$$

where α_n and α_m are dual variables and the radial basis function (RBF) kernel $K(x_n, x_m)$ is applied [16].

As shown in Fig.4-b, we divide the dataset between the positive and negative samples of an available and unavailable slot, since the parking-availability detection is the goal in this stage. We collect positive data of vacant parking-slots without permission marks, whereas the parking-slots contain permission marks, or various objects are designated as the negative data. All datasets are acquired by the regularised representation of a detected slot, which is the same representation for the testing. Classification is performed after the trained parameters are obtained.

4. Experiments and Evaluation

4.1. Experimental Setup

Because the experiments were conducted with a vehicle platform for the 2014 Hyundai Motors Autonomous Vehicle Competition, the proposed algorithm was implemented in real-time using C++/OpenCV. The algorithm runs at a frame rate of 21.9 fps on a normal PC (Intel Core i5-Q720 CPU @ 1.60GHz/4.00GB RAM) with an image size of 752x480 pixels. The cameras were mounted both on the rear side (190-degree fish-eye lens) and on the roof of the vehicle, projecting the side (Fig.5). Image sequences were acquired when the vehicle searched for available parking-slots and reversed to park backwards.

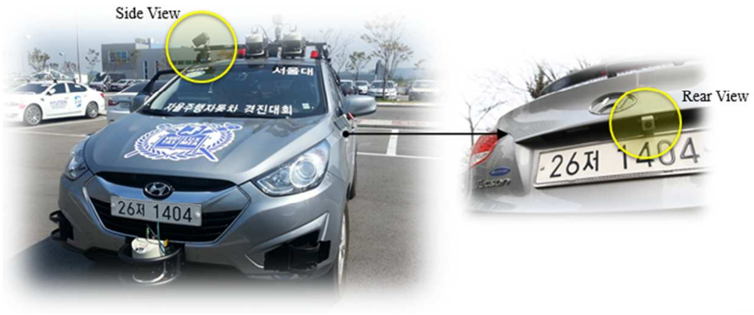


Fig.5. *Camera configurations of Autonomous Vehicle Platform.*

We precede the parameter training for slot-validation and slot-occupancy classification stages using a dataset (Table 2) before the test. We discriminate between the training and test set using different images. For slot recognition, the learning samples consist of diverse slot widths, parallelism (even some distortions) and corner angles, but do not include irregular patterns. To flourish the evaluation, we additionally include certain undefined patterns (defined patterns following [8]) and somewhat distorted images for the test set. Meanwhile, for parking-slot availability classification, a dataset is collected from the results of the parking-slot recognition stage. The training set includes inner-marks, vehicles, pedestrians and traffic cones captured with varying camera angles. Not only did we test different images of vehicles, inner-marks and objects, but we also included objects such as small tires, different shapes of traffic cones, motorcycle and waste bags. The test samples contain 2408 parking-slots in 1,184 frames in total.

Table 2 Total number of Training Datasets for BN and SVM classifier

Dataset	BN (Slot Recognition)	SVM (Occupancy Classification)
Positive samples	2602 (Slots)	1312 (Available slots)
Negative samples	1280 (None slots)	1290 (Unavailable slots)

4.2. Evaluation

The contributions of this study include the evaluation of four factors: the line-marking extraction filter, feature-clustering, parking-slot recognition and parking-slot availability classification performances. First, the proposed cone-hat filter is compared with other popular line-marking extraction filters (Fig.6). We evaluate 150 images containing different illuminance conditions, including shiny and rainy conditions. We utilise a graphic software tool and label the line markings manually to build ground truth. Three classic metrics [24] are calculated as

$$\begin{aligned}
&\text{True positive rate (TPR)} = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False negative (FN)}},\\
&\text{False positive rate (FPR)} = \frac{\text{False positive (FP)}}{\text{False positive (FP)} + \text{True negative (TN)}},\\
&\text{Dice similarity coefficient } (DSC) = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}.
\end{aligned} \qquad (14)$$

The performance of the ridge, global and local threshold filters are highly dependent on a certain intensity threshold. These threshold-based approaches deteriorate in cases where a certain threshold works well in one scenario, but does not work well in others. We additionally compare the LoG filter, which is independent of the intensity threshold. The performance of the cone-hat filter extracts line marks robustly in most cases.

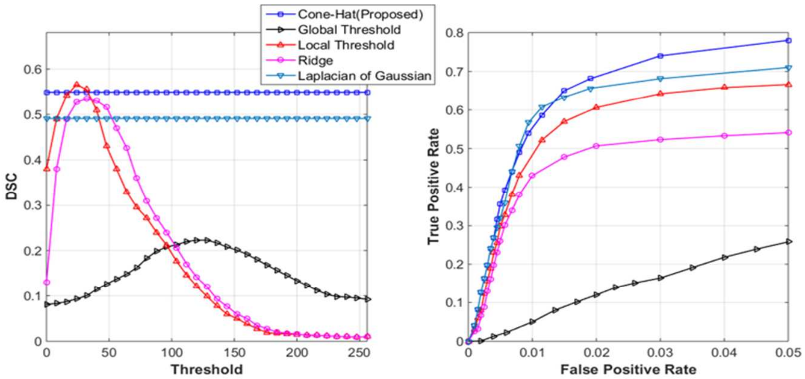


Fig.6. *Performances of line-mark extraction filters.*

Second, clustering performance is evaluated by comparing other methods, such as K-means clustering, hierarchical clustering, density-based spatial clustering of applications with noise (DBSCAN) and the EM clustering algorithm, in Table 3. The measured processing time is for clustering six clusters with 3,483 line-marking features. K-means and hierarchical clustering assume the knowledge of the cluster number and their performances is unreliable. EM and DBSCAN do not require the number of clusters and are more robust. However, the processing time is not attainable for real-time processing. Contrary to the processing time in these methods, the proposed clustering algorithm is reliable and faster.

Table 3 Clustering Performances of Diverse Clustering Algorithms.

Algorithms	Ours	Kmeans	Hierarchical	DBSCAN	EM
Time(s)	0.052	0.06	17.49	0.71	720.3
Accuracy	0.962	0.736	0.874	0.987	0.977

In addition, the parking-slot recognition stage is evaluated from three aspects: detection performance, accuracy and the effectiveness of slot validation. For the detection performance, the algorithm is evaluated to detect parking-slots properly using metrics [8], which are calculated as

$$\text{Recall} = \frac{\text{No. of correctly detected slots}}{\text{No. of existing slots}},\ \text{Precision} = \frac{\text{No. of correctly detected slots}}{\text{No. of detected slots}}. \qquad (15)$$

If more than two slot junctions are visible, we regard the parking-slot as existing slots (ground truth). If the detected slot is distorted 15 percent away from the span, we manually count it as incorrect detection.

Table 4 Parking-slot Recognition Performances.

Cases	Normal Cases		Parking-line Occluded		Undefined Patterns		Overall	
(# of slots)	(1266)		(961)		(182)		(2408)	
Performance	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Proposed Method	0.965	**0.938**	**0.919**	0.920	**0.906**	**0.916**	**0.942**	**0.930**
Jae-kyu *et al.* [6]–[9]	**0.979**	0.927	0.874	**0.925**	0.834	0.904	0.926	0.926

We evaluate the performance in varying circumstances and compare the results with junction based-methods in Table 4. A junction-based method [6]-[9] shows decent performances in most normal cases, whereas our method is less robust in a full parking lot. In this case, warped images of vehicle's wheel or license plate occasionally have similar intensities and widths to the line-markings, thus preventing the effective detection of parking-lines. However, junction-based methods are negatively affected by occlusion in entrance lines and undefined patterns due to their high dependency on the slot-pattern classifier. If a corner angle is distorted and wrongly classified, the higher hierarchical stage of their classifier contains defects in inferring slot patterns. On the other hand, our method has more flexibility. Once the lines compose a rectangle, the BN model examines their geometric features within probabilistic margins. Even if the certain observation belongs away from a proper span, other proper observations compensate in a probabilistic integration. Unfortunately, for open-type slots, this method is less effective because of the lower number of observational nodes than closed-type slots.

The accuracy of the recognised parking-slot is measured regarding errors in slot's orientation and corner location (Fig.7-a) with manual assessment. This is measured in a sequential manner when the vehicle searches for and parked in a single target slot. Most intermittent peak noises are ignorable if we adopt a simple tracking method. Meanwhile, slot validity using the BN model is verified through a comparison with the slot-width thresholding method. Fig.7-b illustrates a receiver-operating characteristics (ROC) curve with varying parameters. The figure is demonstrated with a series of margins ranging from lenient to strict. Although the detection performance of the BN model can occasionally be lower than its width-size validity, the BN model remains mostly effective in terms of safety by sustaining a high TPR with a low FPR.

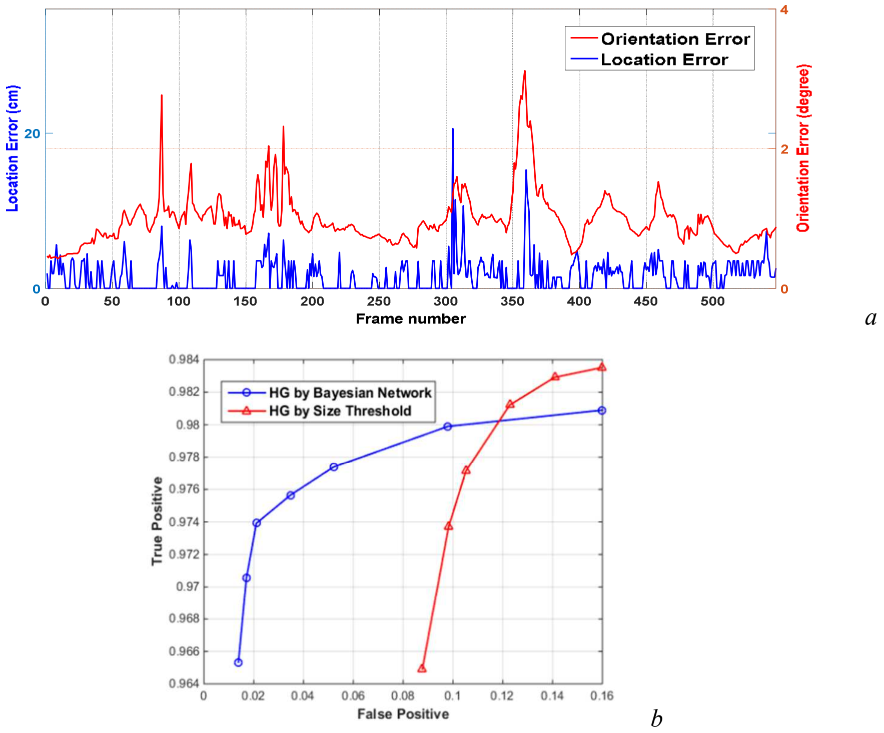


Fig.7. Parking-slot-recognition evaluations
a Errors of detected parking-slot.
b ROC curve of slot-recognition performance.

Lastly, the performance of parking-slot availability classification is evaluated quantitatively. We calculate the FPR, false negative rate (FNR) and error rate, which are defined [12] as

$$\begin{aligned} \text{FPR} &= \frac{\text{No. of available slots being detected as unavailable}}{\text{No. of available slots}}, \\ \text{FNR} &= \frac{\text{No. of unavailable slots being detected as available}}{\text{No. of unavailable slots}}, \\ \text{Error rate} &= \frac{\text{No. of incorrect detection in both un/available slots}}{\text{No. of tested slots}}. \end{aligned} \qquad (16)$$

In these conditions, the performance of the proposed method is compared to those of previous methods in Table 5. The previous methods target only vehicles and analyse only the edge feature distribution or occupancy ratio. Since Wang [28] only analyses the front-slot-region occupancy, small objects are barely recognised. Meanwhile, it is negatively affected by the existence of inner-marks, because Houben's method [29] distinguishes the existence of edge features. In comparison, our method can handle more diverse types of parking-slot availability. Like these previous approaches, however, our method also has weaknesses on object's warping effect, such as a warped vehicle image that overflows to vacant slots. In the top-left image of Fig.8-d, the leftmost slot is wrongly recognised as unavailable, since a large portion of the warped vehicle image overflows. Meanwhile, invisible portions in regularised slots are negligible until two-thirds are visible. However, a half-visible slot has a higher error rate of approximately 4%. Therefore, if a certain application's FOV mostly covers less than a half of the slot, the region of regularised inputs can be alternatively adjusted to the corresponding portion.

Table 5 Parking-slot Availability Classification Performances

Performance ratio	Available Slots (FPR)		Unavailable Slots (FNR)			Error rate
Category (# of tested slots)	Normal Vacant (1072)	Informative mark (117)	A Parked car (667)	Small object (149)	Permit mark (97)	Total (2360)
Proposed Method	4.67 %	**14.53%**	3.56 %	**8.22%**	**8.63 %**	**5.59%**
Wang *et al.* [28]	8.78 %	84.60 %	3.34 %	90.96%	20.86 %	24.43%
Houben *et al.* [29]	**4.45%**	55.56 %	**2.99 %**	15.34%	12.80 %	8.87%

Fig.8 presents snap-shots of the evaluated results with side-view and rear-view images. The original image and BEV image are both displayed simultaneously. While the BEV image indicates intermediate procedures, the front image shows the results. The parking-slot recognition stage shows a decent performance (Fig.8-a) including diverse slot patterns (Fig.8-b). Meanwhile, Fig.8-c shows that improperly generated slot-candidates are filtered by the BN model-based slot validation. In Fig.8-d, various kinds of un/available slots are classified.

We did not directly compare performance between the rear-view (wide) and side-view (normal) images, since they targets different regions. However, there is no big performance difference between them, because we analyse the overall procedures in BEV images. The rear-view images tend to perform slightly decently in parking-slot recognition with less parking-line occlusions, whereas their parking-slot availability classification performance is slightly lower with more distortions.

5. Conclusion

This paper proposes an available parking-slot recognition algorithm that contains fewer constraints through visual sensors. The proposed slot-recognition stage makes the algorithm more cost-effective and reliable, while the occupancy classification stage of the recognised parking-slots solves parking-slot availability classification issues. Since temporal information is not included, there is room for the improvement of slot-recognition. Moreover, other sensor configurations, including AVM system, can be demonstrated to show the robustness of our proposed algorithm.

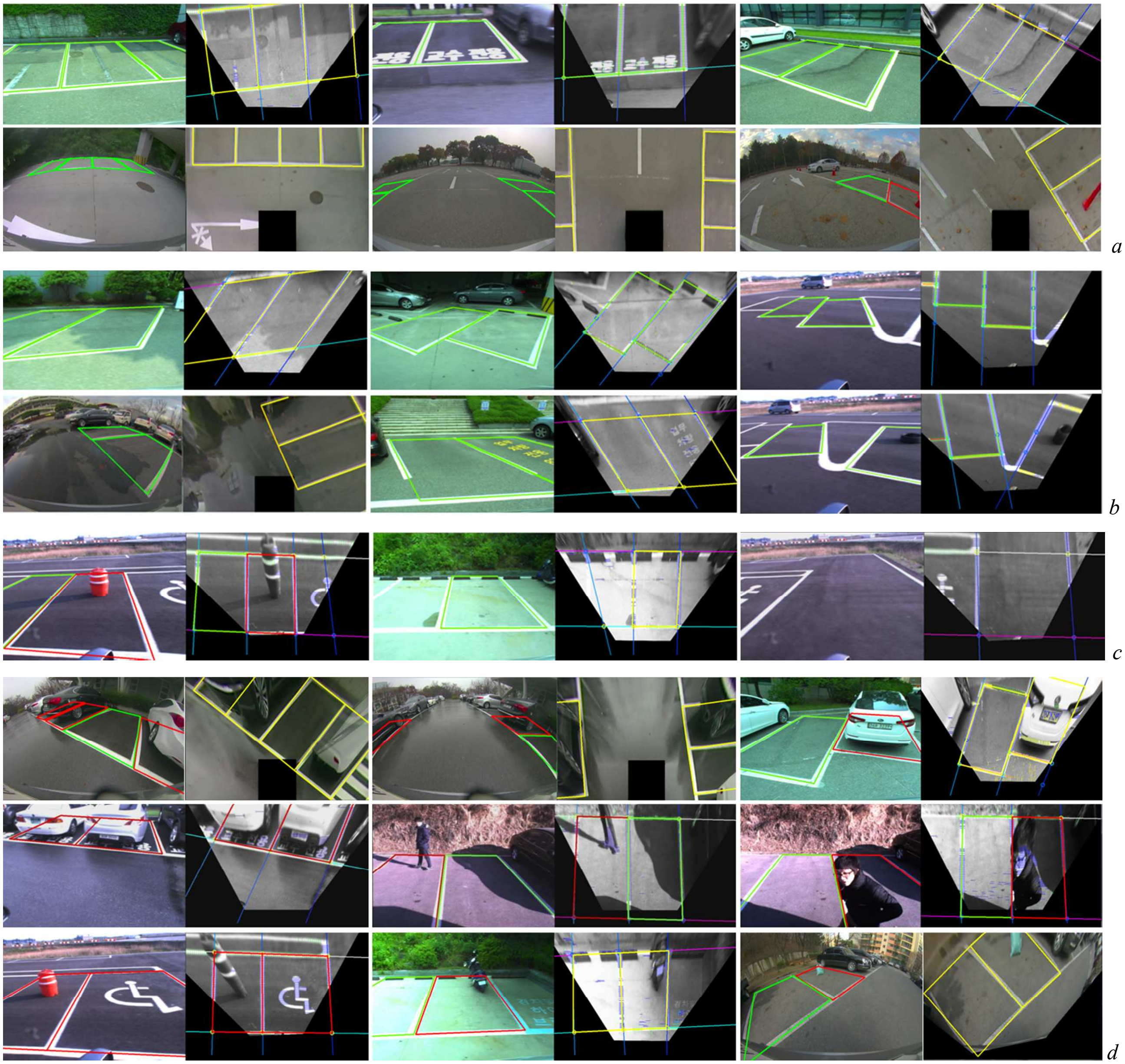


Fig.8. *Snapshots of parking-slot recognition (a-c) and parking-slot availability classification (d).*
a Detection results of side-view images (1st row) and rear-view images (2nd row).
b Diverse slot patterns (diamond, slanted rectangular, open, semi-open slanted, and irregular)
c Improperly composed parking-slots caused by the false positives of line detections are filtered by BN model-based slot validation.
d The green colour indicates available slots, and the red colour indicates unavailable slots. Occupancy cases with cars, pedestrians, permit marks and small objects (traffic cones, a motorcycle and waste bags).

6. References

[1] Xu, J., Chen G., Xie, M.: 'Vision-guided automatic parking for smart car', IEEE Intelligent Vehicle Symposium, June 2000, pp. 725-730

[2] Jung, H., Kim, D., Yoon, P., Kim, J.: 'Parking slot markings recognition for automatic parking assist system', IEEE Intelligent Vehicles Symposium, June 2006, pp. 106-113

[3] Hamada, K., Hu, Z., Fan, M. and Chen, H.: 'Surround View based Parking Lot Detection and Tracking', IEEE Intelligent Vehicles Symposium, Seoul, June 2015, pp. 1106-1111

[4] Jung, H., Lee, Y.: 'Uniform User Interface for Semi-automatic Parking Slot Markings Recognition', IEEE Transactions on Vehicular Technology, 2010, 59, (2), pp 616-626

[5] Jung, H.: 'Semi-automatic parking slot marking recognition for intelligent parking assist systems', The Journal of Engineering, 2014, pp 1-8

[6] Suhr, J., Jung, H.: 'Fully-automatic recognition of various parking slot markings in Around View Monitor image sequences', IEEE Conf. Intelligent Transportation Systems, Alaska, Sep. 2012, pp. 1294-1299

[7] Suhr, J., Jung, H.: 'Full-automatic recognition of various parking slot markings using a hierarchical tree structure', Optical Engineering, 2013, 52, (3), pp 1-14

[8] Suhr, J., Jung, H.: 'Sensor Fusion-based Vacant Parking Slot Detection and Tracking', Transactions on Vehicular Technology, 2014, 15, (1), pp 21-36

[9] Choi, J., Chang, E., Yoon, D., Ryu, S.: 'Sensor Fusion-Based Parking Assist System', SAE Technical Paper, Michigan, 2014, pp 1-5

[10] Suzuki,Y., M. Koyamaishi, Yendo,T., *et al.*: 'Parking assistance using multi-camera infrastructure', IEEE Intelligent Vehicle Symposium, Jun. 2005, pp. 106-111

[11] Tanaka, Y., Tanaka, M., Saiki, M., *et al.*: 'Development of image recognition for a parking assist system', IEEE Conf. Intelligent Transportation System, Japan, Oct. 2006, pp. 1-7

[12] Huang, C., Tai, Y., Wang, S.: 'Vacant Parking Space Detection Based on Plane-based Bayesian Hierarchical Framework', IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23, (9), pp. 1598-1610

[13] Nieto, M., Salgado, L.: 'Robust multiple lane road modelling based on perspective analysis', Int. Conf. Image Processing, San Diego, Sep 2008, pp. 2396-2399

[14] Dalal, N., Triggs, B.: 'Histograms of Oriented Gradients for Human Detection', Computer Vision and Pattern Recognition, San Diego, June 2005, pp. 886-893

[15] Felzenszwalb, P., McAllester, D., Ramanan, D.: 'A Discriminatively Trained, Multiscaled, Deformable Part Model', Computer Vision and Pattern Recognition, Alaska, June 2008, pp. 1-8

[16] Barber, D.: 'Bayesian Reasoning and Machine Learning'(Cambridge University Press, 1st edn. 2012), pp. 325-330

[17] Degerman, P., Pohl, J., Sethson, M.: 'Hough transform for parking space estimation using long range ultrasonic sensors', SAE Paper, Michigan, 2006, pp. 1-7

[18] Satonaka, H., Okuda, M., Hayasaka, S., Endo, T., Tanaka, Y., Yoshida, T.: 'Development of parking space detection using an ultrasonic sensor', The World Congress on Intelligent transportation Systems and Services, London, 2006, pp.1-8

[19] Jung, H., Cho, Y., Yoon, P., Kim, J.: 'Scanning Laser Radar-Based Target Position Designation for Parking Aid System', IEEE Transactions on Intelligent Transportation Systems, 2008, 9, (3), pp. 406-424

[20] Timpner, J., Rottmann, S., Wolf, L.: 'Vehicular Communications in the V-Charge Project', Proc. of GI/ITG KuVS Fachgesprach Inter-Vehicle Comm., Luxembourg, Feb 2015, pp. 41-44

[21] Suhr, J., *et al.*: 'Automatic free parking space detection by using motion stereo-based 3D reconstruction', Machine Vision and Applications, 2011, 21, (2), pp. 163-176

[22] Jung, H., Kim, D., *et al*: '3D vision system for the recognition of free parking site location', Int. Journal of Automotive Technology, 2007, 7, (3), pp. 351-357

[23] Llorca, D., Alvarez, S.: 'Vision-based parking assistance system for leaving perpendicular and angle parking lots', IEEE Intelligent Vehicles Symposium, Gold Coast, June 2013, pp 437-442

[24] Pollard, E., Gruyer, D., Cord, A.: 'Lane Marking Extraction with Combination Strategy and Comparative Evaluation on Synthetic and Camera Images', IEEE Conf. Intelligent Transportation Systems, Washington DC, Oct. 2011, pp. 1741-1746

[25] Su, B., Lu, S.: 'A System for Parking Lot Marking Detection', Springer Adv. in Multimedia Information Processing-PCM, 2014, 8879, pp. 268-273

[26] Kang, S, Lee, S., Seo, S.: 'Multi-lane detection based on accurate geometric lane estimation in highway scenarios', IEEE Intelligent Vehicles Symposium, Michigan, June 2014, pp. 221-226

[27] Lee, S., Kim, S., Seo, S.: 'Accurate Ego-Lane Recognition utilizing Multiple Road Characteristics in a Bayesian Network Framework', IEEE Intelligent Vehicles Symposium, Seoul, June 2015, pp. 543-548

[28] Wang, C., Zhang, H., *et al.*: 'Automatic Parking Based on a Bird’s Eye View Vision System', Adv. in Mechanical Engineering, 2014, 6, (847406), pp. 1-13

[29] Houben, S., Komar, M., *et al.*: 'On-Vehicle Video-based Parking Lot Recognition with Fisheye Optics', IEEE Conf. Intelligent Transportation Systems, Hague, Oct 2013, pp 7-12