Optical Engineering

SPIEDigitalLibrary.org/oe

Full-automatic recognition of various parking slot markings using a hierarchical tree structure

Jae Kyu Suhr
Ho Gi Jung

Full-automatic recognition of various parking slot markings using a hierarchical tree structure

Jae Kyu Suhr
Hanyang University
Research Institute of Automotive Electronics and Control
Department of Automotive Engineering
222 Wangsimni-ro, Seongdong-gu
Seoul 133-791, Republic of Korea

Ho Gi Jung
Hanyang University
Department of Automotive Engineering
222 Wangsimni-ro, Seongdong-gu
Seoul 133-791, Republic of Korea
E-mail: hogijung@hanyang.ac.kr

Abstract. A full-automatic method for recognizing parking slot markings is proposed. The proposed method recognizes various types of parking slot markings by modeling them as a hierarchical tree structure. This method mainly consists of two processes: bottom-up and top-down. First, the bottom-up process climbs up the hierarchical tree structure to excessively generate parking slot candidates so as not to lose the correct slots. This process includes corner detection, junction and slot generation, and type selection procedures. After that, the top-down process confirms the final parking slots by eliminating falsely generated slots, junctions, and corners based on the properties of the parking slot marking type by climbing down the hierarchical tree structure. The proposed method was evaluated in 608 real-world parking situations encompassing a variety of different parking slot markings. The experimental result reveals that the proposed method outperforms the previous semiautomatic method while requiring a small amount of computational costs even though it is fully automatic. © *2013 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.OE.52.3.037203]

Subject terms: driver assist system; parking assist system; parking slot marking recognition; hierarchical tree structure.

Paper 121603 received Nov. 2, 2012; revised manuscript received Dec. 27, 2012; accepted for publication Feb. 1, 2013; published online Mar. 7, 2013.

1 Introduction

Due to the rapidly growing interests in parking aid products,[1,2] automatic parking systems have been extensively researched in both academia[3] and industry.[4] These systems consist of three components: target position designation, path planning, and path tracking. Approaches to the target position designation can be categorized into four types: user-interface based,[2,5,6] infrastructure based,[7–10] free-space based,[11–18] and parking slot marking based.[19–24] Most of the parking aid products on the market utilize a user-interface based approach[2,5,6] or a free-space based approach using ultrasonic sensors.[25–27]

Compared with the other approaches, the parking slot marking based approach has the following advantages: (1) it can be combined with the user-interface based approach to reduce the inconvenience caused by repetitive driver operations, a dominant drawback of the user-interface based approach. (2) Unlike the free-space based approach, it is able to more correctly align parking spaces because its designation process does not depend on the poses of adjacent vehicles but parking slot markings. (3) It can effectively be used for slanted parking situations. The free-space based approach using ultrasonic sensors fails in slanted parking situations due to the limitations of the sensors. (4) It usually consumes a small amount of computational resources compared with the free-space based approach using binocular or motion stereo. (5) It does not require additional sensors, such as a stereo camera,[11] scanning laser radar,[14] or short range radar[17] since it utilizes the existing rear-view cameras. The parking slot marking based approach has a limitation in that it can only be used in cases where parking slot markings are present. However, this approach is still quite useful since drivers want to use an automatic parking system in complex urban situations where parking slot markings are usually present.

The parking slot marking based approach can be divided into semiautomatic and full-automatic methods. The semiautomatic method is likely to produce more reliable results and consume less computational resources compared with the full-automatic method because it has additional information from drivers. This method is also useful as a backup tool for handling cases when the full-automatic method fails. Jung et al.,[20] proposed a method which requests a driver to mark a dot inside the preferable parking slot, and then recognizes line segments using a directional intensity gradient. This method was evaluated with limited situations and can be applied to only one type of parking slot marking. To overcome this drawback, they proposed another method[19] which can recognize various types of parking slot markings with greater user cooperation. This method asks users to input the initial positions of the two junctions which construct the entrance of the parking slot. Based on these initial positions, two junctions are classified and localized based on neural network and genetic algorithms. It was quantitatively evaluated and achieved a 91% recognition rate.

Full-automatic methods are also developed in various ways. Xu et al.[21] proposed a method that recognizes parking slot markings based on the color segmentation technique. This method is able to recognize only red colored markings and the color information-based approach is known to be sensitive to illumination changes. Jung et al.,[22] utilized a binocular stereo technique to reconstruct the rear-view three-dimensional (3-D) structure and recognized parking slot markings by extracting the pixels on the ground plane using a homography constraint. This method requires a

stereo camera that is not commonly used as a rearview camera and needs a high computational cost for stereo matching. Tanaka et al.[23] proposed a method that recognizes parking slot markings by detecting a group of straight lines by applying random sample consensus (RANSAC) to the edge segments. Since this method is highly dependent on line detection, its performance could be degraded with occluded and partially damaged marking lines. Jung et al.[24] recognized parking slot markings with a fixed width by applying a special filter to the Hough space of the bird's-eye view edge image. This method could be sensitive to marking line width as well as occlusions and partial damage to marking lines due to its dependency on line detection. All these full-automatic methods have the same drawback that they are unable to deal with various parking slot markings.

To overcome this drawback of previous full-automatic methods, this paper proposes a novel method for automatically recognizing various types of parking slot markings. The proposed method models various types of parking slot markings as a hierarchical tree structure of type-slot-junction-corner and automatically recognizes them based on this structure. Figure 1 shows the flowchart of the proposed method. The proposed method mainly consists of two processes: bottom-up and top-down. Once a bird's-eye view image is generated immediately after a driver shifts into reverse gear to start parking, the bottom-up process generates corner-junction-slot candidates and determines the type of parking slot markings by climbing up the hierarchical tree structure. After that, the top-down process confirms the final parking slots by removing the falsely generated candidates based on the properties of the parking slot marking type while climbing down the hierarchical tree structure. In experiments, the proposed method was evaluated in 608 real world parking situations encompassing a variety of different parking slot markings, and the result reveals that the proposed method outperforms the previous semiautomatic method[19] and automatic method[24] while requiring less computational costs even though it is full-automatic.

The rest of this paper is organized as follows. Section 2 presents the structural analysis of parking slot markings. Section 3 briefly describes the bird's-eye view image generation. Section 4 explains the bottom-up process for corner-junction-slot generation and parking slot marking type selection. Section 5 describes the top-down process for final slot determination. Section 6 presents the experimental results and analyses. Finally, we conclude the paper with a summary and some suggestions for future work in Sec. 7.

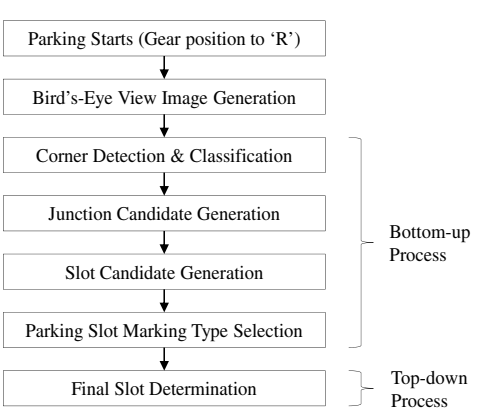

Fig. 1 Flowchart of the proposed method.

2 Structural Analysis of Parking Slot Markings

In this chapter, we analyze the structures of various parking slot markings and show how those structures can be modeled as a hierarchical tree structure of type-slot-junction-corner. This paper deals with the following four types of parking

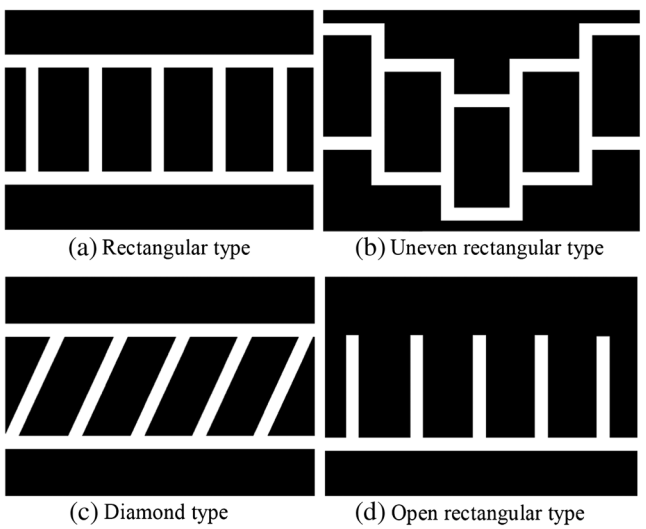

Fig. 2 Four types of parking slot markings.

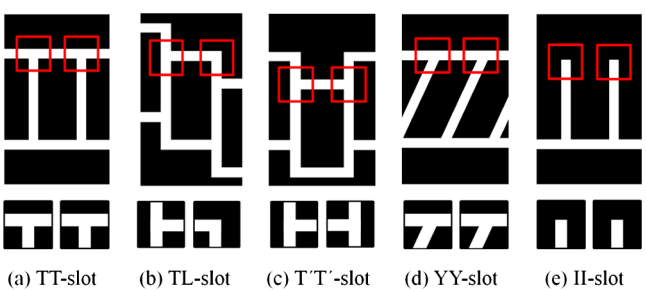

Fig. 3 Five types of slots (first row) and corresponding junction pairs (second row). Slots in (a), (d) and (e) form the rectangular, diamond, and open rectangular types, respectively, and slots in (b) and (c) form the uneven rectangular type.

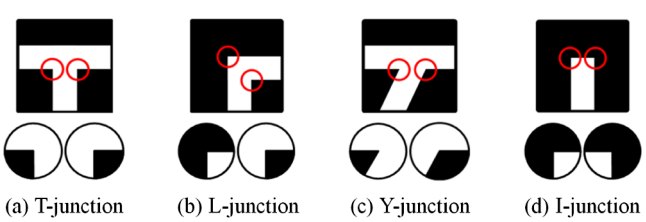

Fig. 4 Four types of junctions (first row) and corresponding corner pairs (second row).

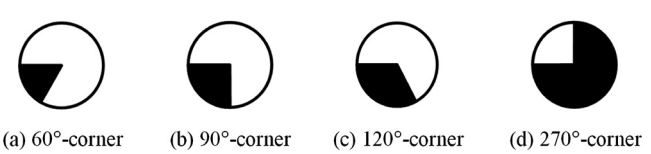

Fig. 5 Four types of corners.

slot markings: rectangular, uneven rectangular, diamond, and open rectangular as shown in Fig. 2. Four types of parking slot markings consist of five types of slots as shown in the first row of Fig. 3, and the entrance of each slot has two junctions as shown in the second rows of Fig. 3. In this figure, (a), (d), and (e) form rectangular, diamond, and open rectangular types, respectively, and (b) and (c) form the uneven rectangular type. Slots shown in Fig. 3(a)–3(e) are called TT-slot, TL-slot, T'T'-slot, YY-slot, and II-slot, respectively. Junctions presented in the second row of Fig. 3 can be categorized into four types as shown in the first row of Fig. 4. In junction categorization, rotated and reflected junctions are considered as the same shape. Four junctions in the first row of Fig. 4(a)–4(d) are called T-junction, L-junction, Y-junction, and I-junction, respectively, and each junction consists of two corners as shown in the second row of Fig. 4. Although a junction is basically composed of two corners, it could also be generated from a single corner by assuming that the other corner exits with proper type, location, and orientation. This improves the recognition performance since one of two corners in a junction might not be detected due to damage to parking slot markings. This will be explained in detail in Sec. 4. The corners presented in the second row of Fig. 4 can be categorized into four types as shown in Fig. 5 according to the proportion of the parking slot marking (white) and the ground plane (black). Arc angles of the ground plane (black) for the corners in Fig. 5(a)–5(d) are 60 deg, 90 deg, 120 deg, and 270 deg, respectively, and those corners are called the 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively. According to this structural analysis, various parking slot markings can be modeled as a hierarchical tree structure of a type-slot-junction-corner as shown in Fig. 6.

3 Bird's-Eye View Image Generation

Images taken by a rear-view camera should be transformed into bird's-eye view images before applying the proposed method. Since most of the rear-view cameras utilize a fisheye lens to maximize the view behind the vehicle, two-step transformation consisting of radial distortion correction and projective transformation should be applied to input fisheye images. In this paper, a fifth-degree polynomial model is used to remove radial distortion,[28] and a homography between the ground plane image and the real world ground plane is utilized to generate birds-eye view images from undistorted images.[29] Figure 7 shows an example of this two-step transformation. In this figure, (a) is an original fisheye image, and (b) and (c) are undistorted and bird's-eye view images, respectively.

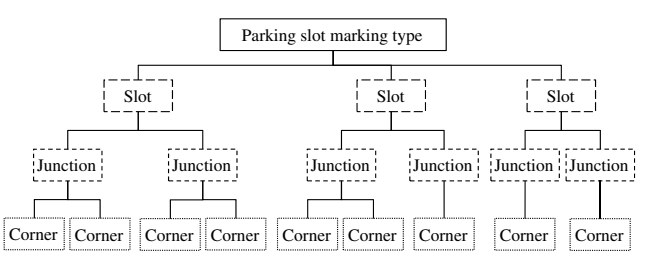

Fig. 6 Hierarchical tree structure of parking slot markings.

4 Bottom-Up Process for Parking Slot Candidate Generation

4.1 *Corner Detection*

This section explains the detection procedure of corners which are the lowest level features in the hierarchical tree structure of the parking slot markings in Fig. 6. The proposed method detects corners by applying the Harris corner detector[30] to bird's-eye view images. The Harris corner detector finds interest points by using a 2×2 matrix related to the local auto-correlation function which measures the intensity structure of the local neighborhood. A cornerness value is calculated by using the determinant and trace of this matrix at each pixel, and the pixel locations with the local maximum cornerness values are chosen as corners.

To appropriately detect the corners of parking slot markings, two parameters of the Harris corner detector should be properly determined. The first parameter is the standard deviation (σ) of the Gaussian windowing function which represents the scale of corner. If σ is inappropriately selected, the Harris corner detector fails to find corners or suffers from a localization error. The second parameter is the radius (d) used for the nonmaximum suppression process. If d is improperly determined, the corners of the parking slot markings are falsely eliminated. Figure 8 shows the corner detection results of a T-junction with different σ and d values. In this figure, a red cross, blue dotted circle, and green dashed circle indicate the detected corner, a circle with radius 3σ, and a circle with radius d, respectively. Figure 8(a) shows the case where only one corner is detected with a localization error because of too large σ. In the case of Fig. 8(b), σ is properly selected but d is set to too large a value. Thus, one corner is correctly detected at an appropriate location, but the other corner is falsely eliminated due to the large value of d for nonmaximum suppression. Figure 8(c) shows the detection results with the proper values of σ and d. It can be seen that two corners of the T-junction are correctly detected in the latter case. Considering the

Fig. 7 Two-step transformation of an input image taken by a rear-view camera with a fisheye lens. (a) Original fisheye image. (b) Undistorted image. (c) Bird's-eye view image.

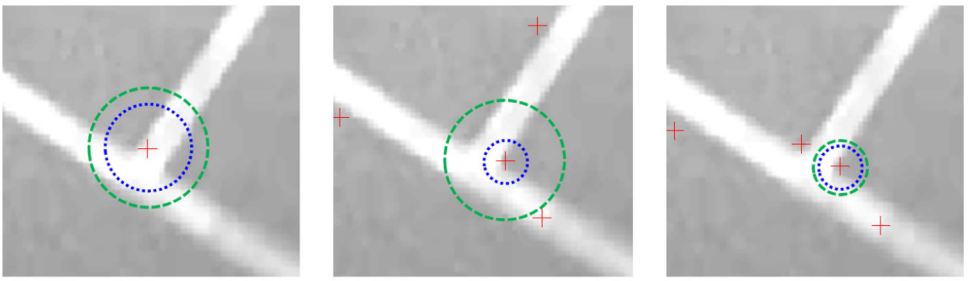

Fig. 8 Detection results of the Harris corner detector with different σ and d values. (a) Detection result with a large value of σ. (b) Detection result with proper σ and too large d. (c) Detection result with proper σ and d. Red cross, blue dotted circle, and green dashed circle indicate the detected corner, circle with radius 3σ, and circle with radius d, respectively.

properties of the two parameters, 3σ and d are set to three pixels which is approximately half the line width of the parking slot markings. Figure 9 shows the corner detection results when applying the Harris corner detector to images of five types of slots with an appropriate parameter setting. It can be noticed that two corners are correctly detected at each junction.

4.2 *Corner Classification and Orientation Estimation*

Once corners are detected by the Harris corner detector, the class and orientation of each corner should be estimated. To this end, this paper utilizes the template matching technique based on a circular intensity profile of each corner. Firstly, templates of four types of corners are generated by assuming an ideal situation where the intensity of the parking slot marking is 255 and the intensity of the ground plane is 0. In this ideal situation, the circular intensity profiles of the 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner in a clockwise direction are produced as shown in Fig. 10(a)–10(d), respectively. These circular intensity profiles are used as templates of four types of corners.

A signal of each corner to be matched with the templates in Fig. 10 is produced by calculating a circular intensity profile around each corner with a radius (r). Since r should be smaller than the line width of the parking slot markings as shown in Fig. 4, it is set to half of the line width. This signal is sampled with a resolution of 1 deg, which means the dimension of the signal is 360. Furthermore, the circular intensity profile is not only calculated with a radius (r) but also with two more radiuses ($r-1$ and $r+1$), and those three profiles are averaged to produce the final signal to achieve the robustness against noise. After calculating a signal of each corner, the signal is matched with templates by using normalized cross correlation (NCC)[31] as a similarity measure. NCC $[\mathbf{c}(\tau)]$ between a signal circularly shifted by τ $[\mathbf{s}(\tau)]$ and a template $[\mathbf{t}(i)]$ is calculated as follows:

$$\mathbf{c}(\tau) = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \frac{[\mathbf{t}(i) - m_{\mathbf{t}}][\mathbf{s}(i+\tau) - m_{\mathbf{s}(\tau)}]}{\sigma_{\mathbf{t}}\sigma_{\mathbf{s}(\tau)}}, \tag{1}$$

where m_t, $m_{s(\tau)}$, $\sigma_{\mathbf{t}}$, and $\sigma_{s(\tau)}$ are means and standard deviations of $\mathbf{t}$ and $\mathbf{s}(\tau)$, respectively, and N is a signal dimension. For reducing the computational cost, this process is conducted in a frequency domain via Fourier transformation based on the correlation theorem.[31]

Once the signal of each corner is matched with the four templates presented in Fig. 10, the class of each corner is determined as a class of the template (j) which produces the maximum value of NCC as (2).

$$(\text{CLASS}, \text{W2BTD}) = \arg\max_{(j,\tau)} \{\mathbf{c}_j(\tau)\}, \quad \text{where } j = 1,2,3,4,\ \tau = 1,2,\ldots,N, \tag{2}$$

where CLASS and W2BTD indicate corner class and white-to-black transition direction, respectively, and $\mathbf{c}_j(\tau)$ is NCC between $\mathbf{s}(\tau)$ and j'th template. White-to-black transition direction (W2BTD), which indicates the direction where the circular intensity value is changed from white (parking slot marking) to black (ground plane) in a clockwise direction, is

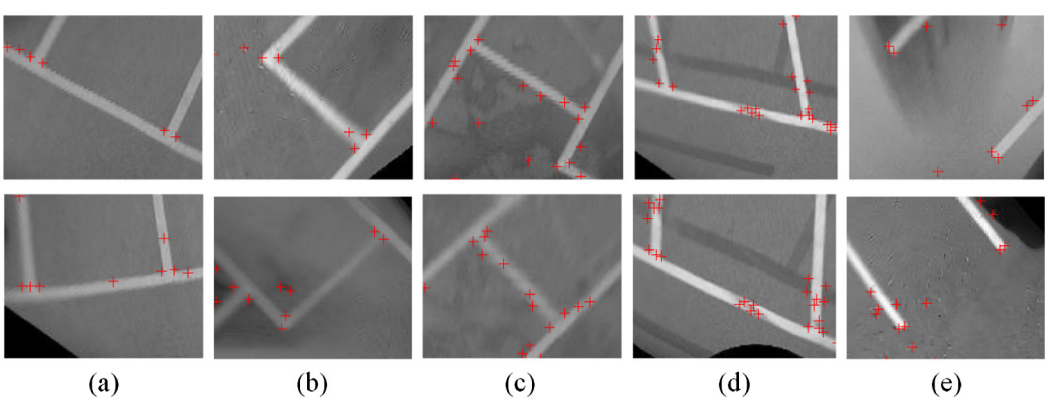

Fig. 9 Detection results when applying the Harris corner detector to images of five types of slots with appropriate parameter setting. (a)–(e) are the detection results of TT-slot, TL-slot, T'T'-slot, YY-slot, and II-slot, respectively.

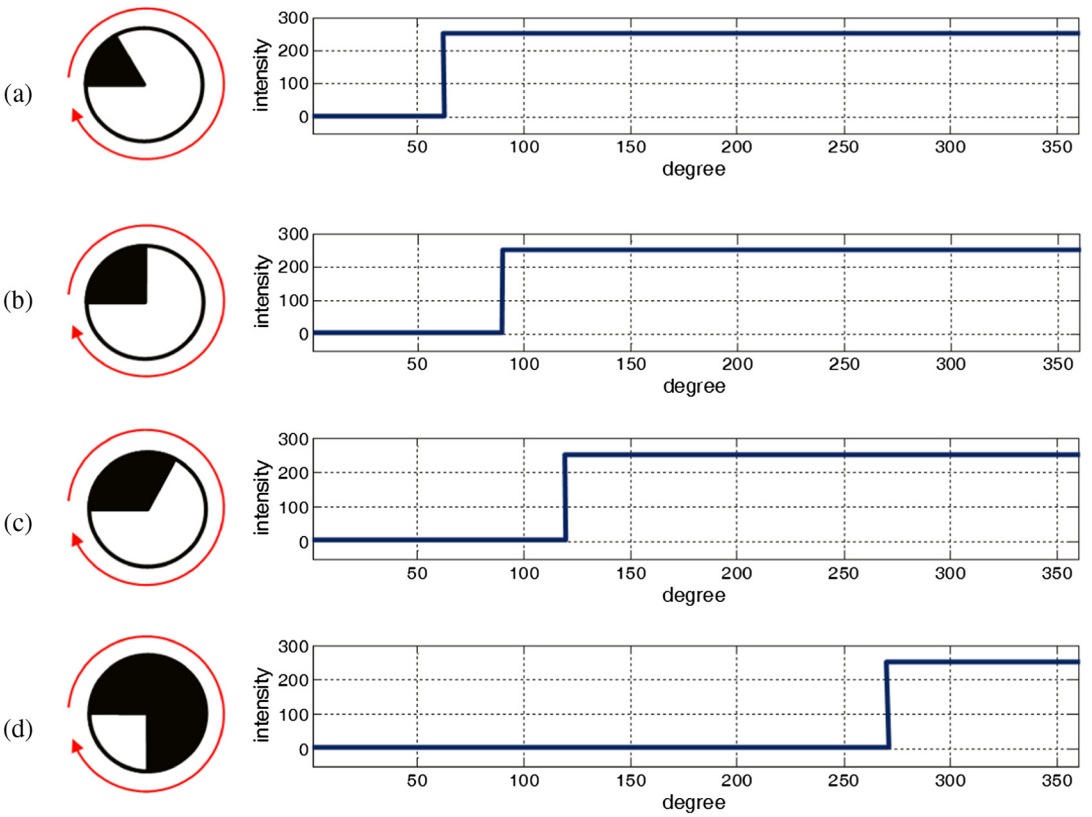

Fig. 10 Templates of four types of corners. (a)–(d) are templates of a 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively.

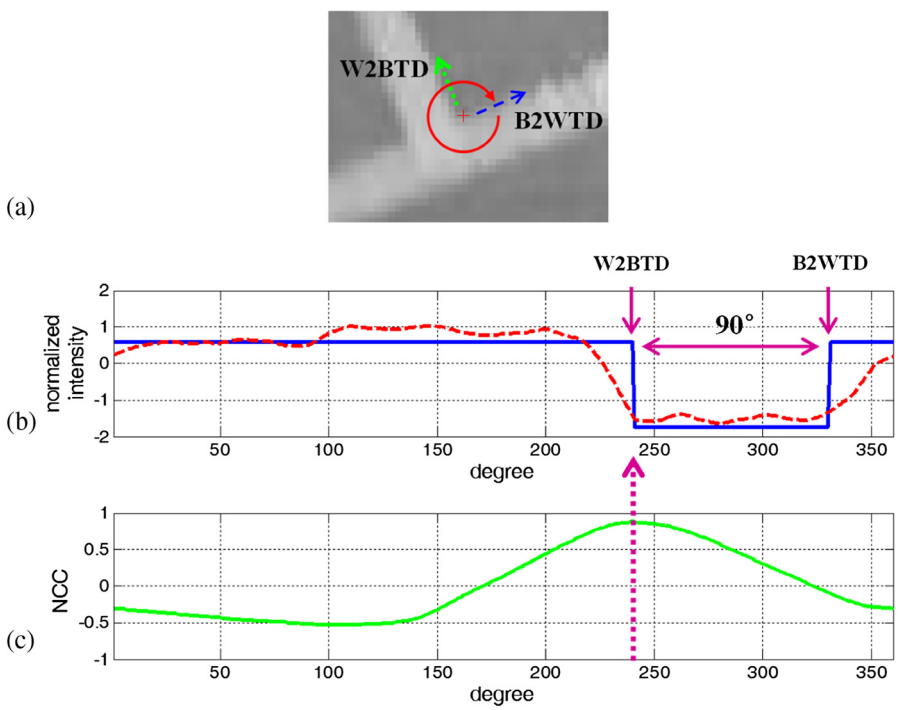

Fig. 11 An example of a template matching procedure. (a) A corner of T-junction where a red circular arrow indicates the radius and direction used to calculate a circular intensity profile, and the green dotted and blue dashed arrows are W2BTD and B2WTD, respectively. (b) Circular intensity profile of the corner in (a) (red dashed line) and the template of 90-deg corner (blue solid line). (c) NCC calculated between the circular intensity profile and the template of 90-deg corner.

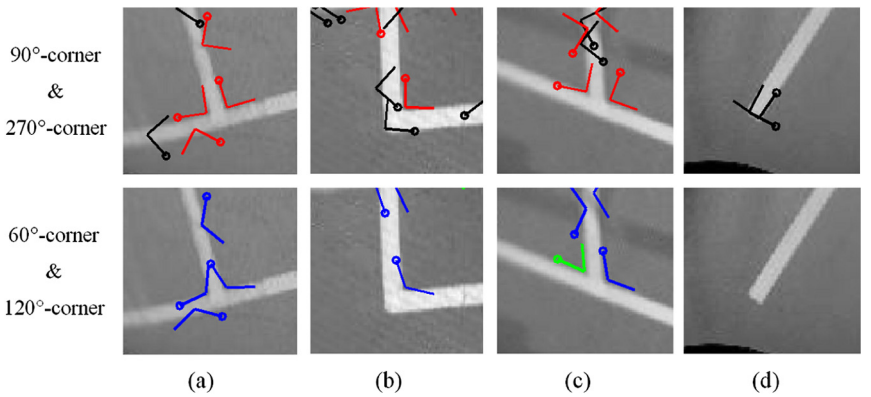

Fig. 12 Results of the corner classification and orientation estimation with images of four types of junctions. (a)–(d) are T-junction, L-junction, Y-junction, and I-junction, respectively. The first row shows the corners classified as a 90-deg corner and 270-deg corner, and the second row shows the corners classified as a 60-deg corner and 120-deg corner.

set to τ that gives the maximum value of NCC. Black-to-white transition direction (B2WTD) which has the opposite meaning of W2BTD is set to $\tau + 60$ deg, $\tau + 90$ deg, $\tau + 120$ deg, or $\tau + 270$ deg if the determined class is 60-deg corner, 90-deg corner, 120-deg corner, or 270-deg corner, respectively. Figure 11 shows an example case when the template matching procedure is applied to a corner of T-junction. In Fig. 11(a), a red circular arrow indicates the radius and direction used to calculate a circular intensity profile, and the green dotted and blue dashed arrows are W2BTD and B2WTD, respectively. A red dashed line in Fig. 11(b) indicates the circular intensity profile obtained from the red circular arrow in Fig. 11(a), and this signal produces the largest NCC with the template of 90-deg corner [blue solid line in Fig. 11(b)] when τ is 240 deg as shown in Fig. 11(c). Thus, W2BTD and B2WTD are set to 240 deg and 240 deg $+90$ deg, respectively.

Since the proposed template matching procedure utilizes simple feature vectors and similarity measure, classes and orientations of the corners are estimated with a low computational cost. On the other hand, this method may misclassify the classes of corners due to complex illumination, severe damage to markings, or low image degradation. In particular, this method often misclassifies 60-deg corners as 90-deg corners and 90-deg corners as 120-deg corners. To overcome this drawback, this paper utilizes an approach which assigns multiple hypotheses to a single corner according to its classification result. If a corner is classified as a 90-deg corner, two hypotheses (60-deg corner and 90-deg corner) are assigned to this corner since the possibility that a 60-deg corner is falsely classified as a 90-deg corner is not negligible. In the same way, a corner classified as a 120-deg corner will also have two hypotheses (90-deg corner and 120-deg corner). This approach is advantageous because at least one correct hypothesis can be assigned to each corner. Inevitably produced incorrect corner hypotheses in this procedure will be suppressed during junction and slot generation process and top-down process which will be explained in detail in the following sections and Sec. 5.

Figure 12 shows the results of the corner classification and orientation estimation with images of four types of junctions. Figure 12(a)–12(d) are images of T-junction, L-junction, Y-junction, and I-junction, respectively. Since it is hard to depict all assigned classes on a single image, corners classified as a 90-deg corner and a 270-deg corner are depicted in the first row and corners classified as a 60-deg corner and a 120-deg corner are depicted in the second row in Fig. 12. This is only for better visual understanding. Green, red, blue, and black line pairs indicate a 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively. Lines with and without circles at their ends are W2BTD and B2WTD, respectively, and an intersection of a line pair is a corner location.

4.3 *Junction Candidate Generation*

Junctions that compose the parking slot markings are categorized into four types as shown in Fig. 4. Different combinations of corners which generate four types of junctions are present in Fig. 13. In this figure, green, red, blue, and black line pairs indicate a 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively. Lines with and without circles at their ends are W2BTD and B2WTD, respectively, and a dashed arrow and its starting point indicate the direction and location of each junction, respectively.

To generate a junction using a pair of corners, four features of a corner (class, location, W2BTD, and B2WTD) are utilized. For instance, a T-junction consists of two 90-deg corners with a distance equal to the marking line width. The W2BTD and B2WTD of one corner are the same as the B2WTD and -W2BTD of the other, respectively. A vector connecting the two corners is opposite to the B2WTD of one corner and -W2BTD of the other. The other junctions are generated in a similar way. Figure 14 shows the results of junction generation in five types of slot situations. In this figure, red, black, green, and blue lines indicate a T-junction,

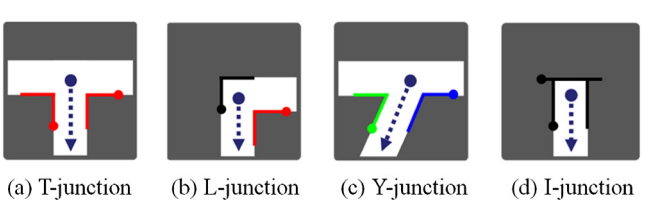

Fig. 13 Different combinations of corners which generate four types of junctions. Green, red, blue, and black line pairs indicate a 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively. Lines with and without circles at their ends are W2BTD and B2WTD, respectively, and a dashed arrow and its starting point indicate the direction and location of each junction, respectively.

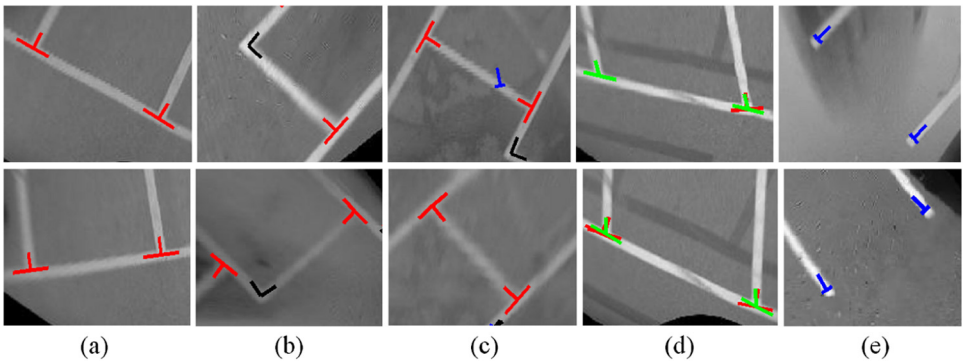

Fig. 14 Results of the junction generation procedure in five types of slot situations. (a)–(e) are TT-slot, TL-slot, T'T'-slot, YY-slot, and II-slot, respectively. Red, black, green, and blue line pairs indicate a T-junction, L-junction, Y-junction, and I-junction, respectively.

L-junction, Y-junction, and I-junction, respectively. Since multiple classes can be assigned to a single corner, multiple junctions are generated by the same corner pair as shown in Fig. 14(d). The junction generation procedure needs three parameters: minimum and maximum line widths of the parking slot markings, and the margins for determining the orientation coincidence.

Although a junction is basically constructed from two corners, we also generate a junction from a single corner by assuming that the other corner with proper type, location, and orientation are present. This approach increases the recognition rate of parking slot markings since one of two corners in a junction may not be detected due to damage and occlusions on parking slot markings or image degradations during the acquisition procedure. Figure 15 shows the ways to generate junctions from different kinds of single corners. In this figure, vivid and pale corners indicate the detected and assumed corners, respectively. Unlike the junction generation of two corners, a single corner generates multiple junction hypotheses due to the inevitable ambiguity. This approach prevents the possibility of missing correct junctions due to the fault of the corner detection, but it produces false junction hypotheses. These false junction hypotheses are suppressed during the slot generation process and top-down process which will be explained in detail in the following sections and chapter 5.

4.4 *Slot Candidate Generation*

Five types of slots can be categorized based on the properties of the junction pairs which compose the entrance of parking slots as shown in Fig. 16. In this figure, a red rectangle and red solid arrow indicate the parking slot and its orientation, respectively. The slot candidate generation procedure is similar to the junction generation procedure. Three features of a junction (type, location, and orientation) are utilized for this task. For example, a TT-slot consists of two T-junctions with a distance equal to the slot width. The orientations of the two junctions are the same, and a vector connecting the two junctions is perpendicular to the orientations of both junctions. The other slots are generated in a similar way. Figure 17 shows the results of the slot candidate generation in five types of slot situations. In this figure, red, black, magenta, green, and blue lines indicate the TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. Since multiple junctions are constructed by the same corner pair, multiple slots are generated by the same junction pair as shown in Fig. 17(d). In this slot candidate generation procedure, two constraints are utilized to reduce the number of false slots which affect the final result and require more computational cost. One is that one of two junctions should be supported by two corners, and the other is that slot orientation should not be toward the camera. The slot candidate generation procedure requires three parameters: minimum and maximum widths

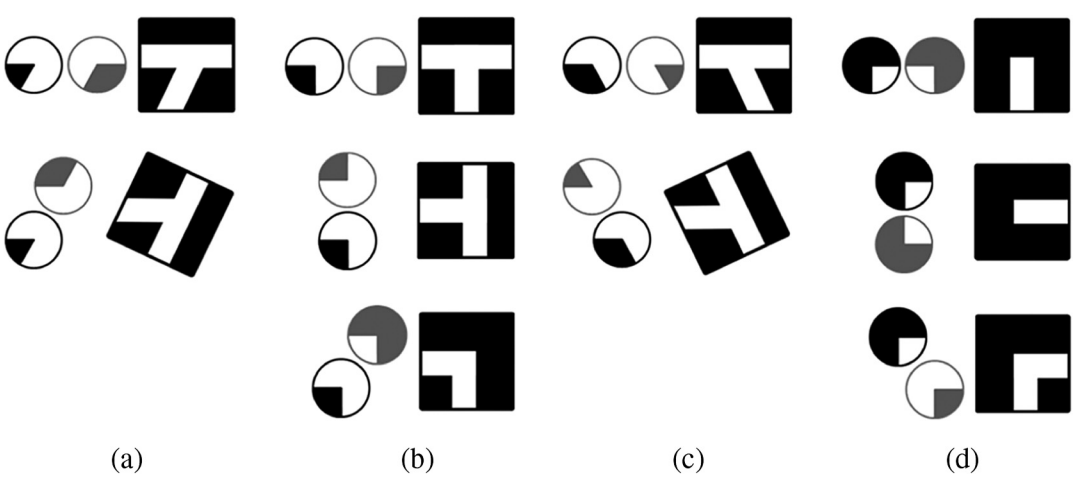

Fig. 15 Junction generation from a single corner. (a)–(d) are junctions generated by a single corner classified as a 60-deg corner, 90-deg corner, 120-deg corner, and 270-deg corner, respectively. Vivid and pale corners indicate the detected and assumed corners, respectively.

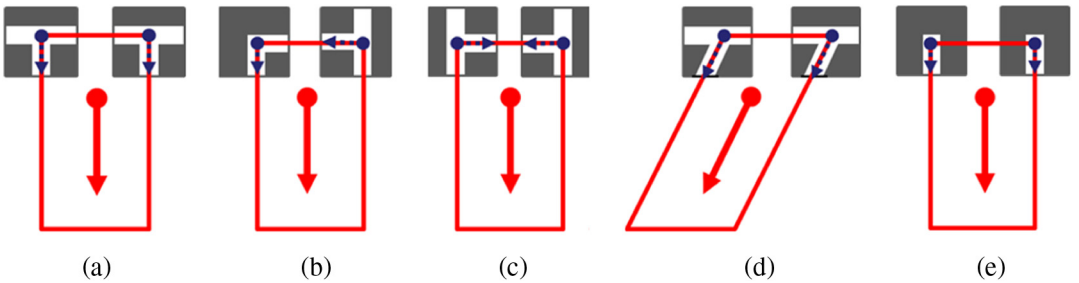

Fig. 16 Five types of slots categorized based on the properties of the junction pairs. (a)–(e) are TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. A red rectangle and red solid arrow indicate the parking slot and its orientation, respectively.

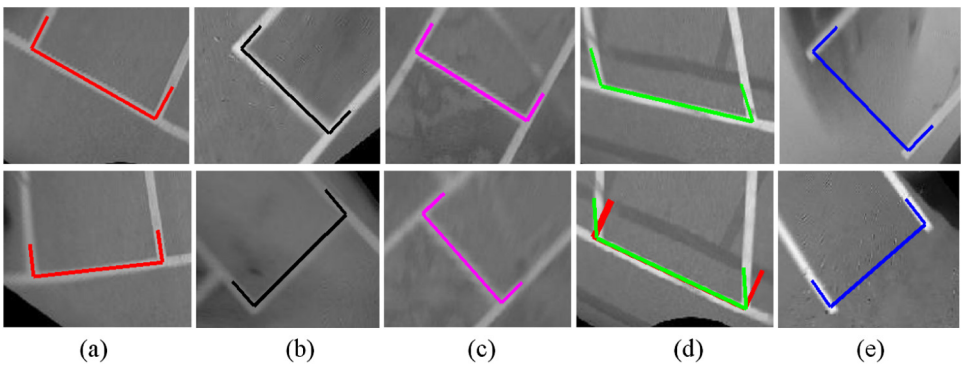

Fig. 17 Results of the slot candidate generation procedure in five types of slot situations. (a)–(e) are TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. Red, black, magenta, green, and blue lines indicate TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

of parking slots in a lateral direction, and a margin to determine orientation coincidence.

4.5 *Parking Slot Marking Type Selection*

Once the slot candidates are generated, the parking slot marking type located at the highest level of the hierarchical tree structure in Fig. 6 should be selected by using the generated slot candidates. For this task, this paper utilizes a method which finds the most appropriate slot called the "best slot" and selects the parking slot marking type by considering which type of parking slot marking includes the best slot. For instance, the parking slot marking is determined as the rectangular type if the type of the best slot is TT-junction.

To select the best slot, this paper calculates the average intensity value (AIV) of the skeleton of a parking slot's entrance as depicted by red lines in Fig. 18.

$$\mathrm{AIV} = \frac{1}{M} \sum_{(x,y) \subset S} I(x,y), \tag{3}$$

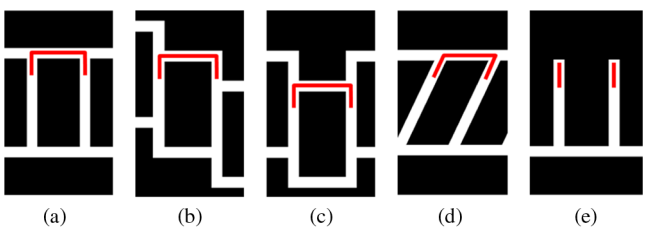

Fig. 18 Skeletons of five types of slots used to calculate average intensity values. Red lines in (a)–(e) are skeletons of a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

where $I(x,y)$ indicates the intensity value of the pixel at (x,y) on the skeleton of the parking slot, S, and M is the total number of pixels on S. The AIV is used as the properness measure of the parking slot. Generally, parking slot markings are much brighter than the ground plane, and pixels near the center of the parking slot marking line tend to have larger intensity value than the boundary pixels.[20] Therefore, the slot located at the most appropriate position has the largest AIV compared with the other candidates.

Figure 19 shows the best slots selected from the initial slot candidates in Fig. 20 via the AIV measure. In this figure, red, black, magenta, green, and blue lines indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. It can be seen that the AIV based approach can effectively select the most appropriate slots in various real world situations despite its computational simplicity. Once the best slot is selected, the parking slot marking type is determined according to the type of best slot. In Fig. 19, the parking slot marking of (a) is recognized as a rectangular type because the type of best slot is a TT-slot, and both (b) and (c) are recognized as uneven rectangular types since their best slots are a TL-slot and T′T′-slot, respectively. Figure 19(d) and 19(e) are recognized as diamond and open rectangular types because their best slots are a YY-slot and II-slot, respectively.

5 Top-Down Process for Final Parking Slot Determination

Section 4 explains the bottom-up process that generates corners, junctions, and slots, and determines the type of parking slot marking by climbing up the hierarchical tree structure. This chapter describes the top-down process that prunes falsely generated slots, junctions, and corners according to the properties of the selected parking slot marking type

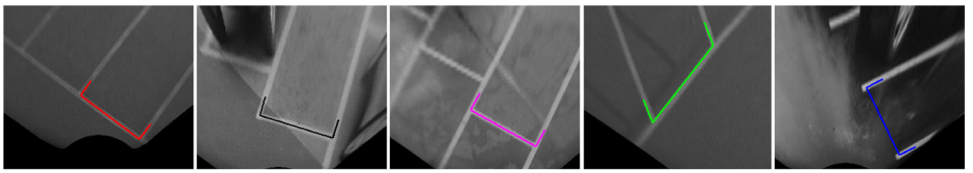

Fig. 19 Results of the best slot selection procedure. Red, black, magenta, green, and blue lines indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

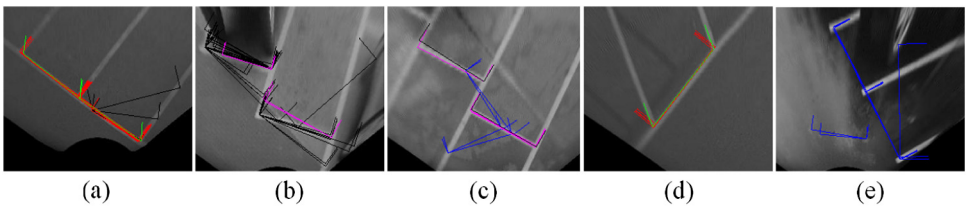

Fig. 20 Initial slot candidates generated from four type of parking slot markings. (a) Rectangular type. (b) and (c) Uneven rectangular type. (d) Diamond type. (e) Open rectangular type. Red, black, magenta, green, and blue lines indicate TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

by climbing down the tree structure. The top-down process consists of two steps. The first step rejects the slot-junction-corner candidates that are discordant with the recognized type of parking slot marking, and the second step selects the most appropriate slot from the overlapped slot candidates.

In the first step of the top-down process, the slot candidates that cannot be produced from the selected parking slot marking type are rejected. For instance, the TL-slots and YY-slots in Fig. 20(a) should be eliminated since the parking slot marking is recognized as a rectangular type that can include only TT-slots. After this type based rejection, slot candidates whose sizes and orientations do not coincide with the size and orientation of the best slot are eliminated. After the first step, there might be some overlapped slots with a similar size and orientation since multiple slots can be generated at similar locations as shown in Fig. 20. These overlapped slots are generated because of the approach that produces parking slots with the junctions generated from not only two corners but also a single corner. These overlapped slots are pruned in the second step of the top-down process since it is impossible for multiple parking slots to be overlapped. In this step, only one slot, the most appropriate, remains of the overlapped slots. To evaluate the slot candidates, AIV used to measure the properness of parking slots is utilized again: the slot with the largest AIV survives out of overlapped slots. Since AIVs of the slot candidates are calculated during the best slot selection procedure, they can be re-used without additional calculation. Figure 21 shows the results of the top-down process in various parking slot marking types. In Fig. 21(a), 21(d), and 21(e) are rectangular, diamond, and open rectangular types, respectively, and (b) and (c) are uneven rectangular types. Red, black, magenta, green, and blue lines indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. It can be noticed that the proposed top-down process can effectively reject the falsely generated slots while retaining the correct slots.

6 Experimental Results

6.1 *Description of Database*

The proposed method was evaluated using the database described in Table 1. Images in the database were acquired by a rear-view camera with a fisheye lens whose resolution and field of view in horizontal and vertical directions are 640×480 pixels and 105 deg $\times$80 deg, respectively. The database includes 608 images which consist of 191 images presented in Ref. 19 and 417 images newly acquired in this paper. The first row of Fig. 22 shows example images of the database. Input images (640×480 pixels) are transformed into bird's-eye view images (600×300 pixels) which include a rear-view within a radius of 6 m. The second row of Fig. 22 shows the bird's-eye view images corresponding to the original images in the first row of Fig. 22.

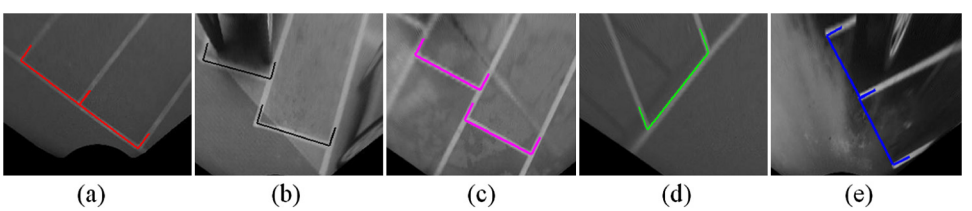

Fig. 21 Results of the top-down process in four types of parking slot markings. (a) Rectangular type. (b) and (c) Uneven rectangular type. (d) Diamond type. (e) Open rectangular type. Red, black, magenta, green, and blue lines indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

Table 1 Number of images in test database.

Type	Rectangular	Uneven rectangular	Diamond	Open rectangular	Total
Database in Ref. 19	65	54	45	27	191
Supplementary database	150	122	40	105	417
Combined database	215	176	85	132	608

6.2 *Performance Evaluation and Comparison*

This section presents the evaluation results of the proposed method and comparison results with the previous semiautomatic method[19] and automatic method.[24] The previous methods[19,24] are called Jung's semiautomatic method and Jung's automatic method, respectively. Two metrics are used: recall and precision as Eqs. (4) and (5).

$$\text{recall} = \frac{\#\text{of true positive (TP) slots}}{\#\text{of slots}}, \tag{4}$$

$$\text{precision} = \frac{\#\text{of true positive (TP) slots}}{\#\text{of true positive (TP) slots} + \#\text{of false positive (FP) slots}}. \tag{5}$$

Jung's semiautomatic method received initial positions of two junctions which compose the entrance of the nearest parking slot and its performance was evaluated by judging whether two junctions are correctly recognized and paired. The performance of the proposed method and Jung's automatic method were evaluated in the same way except they did not receive any initial position of the junction. We refer to this as a nearest-slot recognition performance. Table 2 shows the comparison result between the proposed method and Jung's semiautomatic method in terms of the nearest-slot recognition using the database used in Ref. 19.

On average, the proposed method gives a recall of 95.3% and a precision of 96.3% while Jung's semiautomatic method gives 91.1% for both recall and precision. This reveals that the proposed method outperforms Jung's semiautomatic method by 4.2% and 5.2% in terms of recall and precision, respectively, even though it is full-automatic. There are two main reasons that make the recall of the proposed method higher than that of Jung's semiautomatic method. First, Jung's semiautomatic method segments the junctions of parking slot markings by using intensity clustering and image binarization based on the assumption that the intensity histogram around a junction is three-modal. As mentioned in Ref. 19 the three-modal assumption and the approach directly using intensity values may be sensitive to illumination conditions. For the same task, the proposed method utilizes corners detected based on the distribution of local gradients. Generally, it is known that the gradient based approach is more robust against illumination conditions compared to the intensity-based approach. Second, Jung's semiautomatic method categorizes junctions into eight classes and classifies them via a three-layered neural network. Since the classifier should deal with too many classes, it is inevitable to produce many misclassifications. Contrarily, the proposed method categorizes corners into only four classes and assigns multiple hypotheses to a single corner so as not to lose the correct class label. This approach is supposed to make the junction classification procedure of the proposed method more reliable than that of Jung's semiautomatic method. The proposed method gives a higher precision compared with Jung's semiautomatic method. This is because Jung's semiautomatic method is designed to output one parking slot per trial. This means Jung's semiautomatic method generates a false slot whenever it fails. Contrarily, the proposed method produces no slot when it cannot find any reliable slots. Consequently, it can reduce the number of false positives.

Jung's automatic method was applied to the rectangular type parking slot markings since it is designed to deal with only this type. Jung's automatic method gives a recall of 84.6% and a precision of 85.9%. This reveals that the proposed method also outperforms Jung's automatic method by

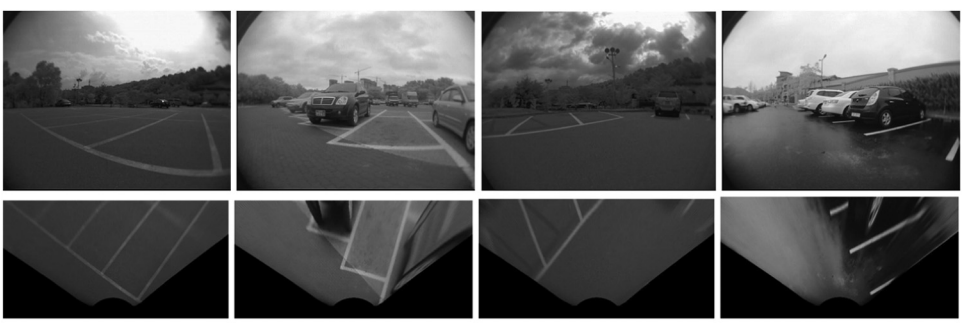

Fig. 22 Original images of four types of parking slot markings (first row) and their bird's-eye view images (second row). (a) Rectangular type. (b) Uneven rectangular type. (c) Diamond type. (d) Open rectangular type.

Table 2 Performance comparison in terms of nearest-slot recognition using the database in Ref. 19.

Type	# of slots	Jung's semiautomatic method				Proposed full-automatic method			
		# of TPs	# of FPs	Recall	Precision	# of TPs	# of FPs	Recall	Precision
Rectangular	65	60	5	92.3%	92.3%	63	2	96.9%	96.9%
Uneven rectangular	54	49	5	90.7%	90.7%	53	3	98.1%	94.6%
Diamond	45	39	6	86.7%	86.7%	41	2	91.1%	95.3%
Open rectangular	27	26	1	96.3%	96.3%	25	0	92.6%	100%
Total	**191**	**174**	**17**	**91.1%**	**91.1%**	**182**	**7**	**95.3%**	**96.3%**

12.3% and 11.0% in terms of precision and recall, respectively. The main reason that causes Jung's automatic method to give a low accuracy is that it is sensitive to marking line width as well as partial occlusions and damage to marking lines due to its dependency on line detection.

Additionally, the proposed method is able to recognize multiple parking slots as far as the image quality is retained. Thus, the proposed method was also evaluated in terms of how many visible slots are recognized. This measure is called the visible-slot recognition rate. In this evaluation, the visible slots are defined as the slots whose junctions are located within 6 meters (300 pixels) of the rear-view camera. Table 3 shows the performance of the proposed method in terms of the nearest-slot recognition and visible-slot recognition using the combined database. The proposed method gives a recall of 92.9% and a precision of 95.9% in terms of nearest-slot recognition. The recall of the combined database is lower than that of the database in Ref. 19 by 2.4% while the precisions of both databases are almost similar. This is probably because the combined database includes harsher conditions compared with the database in Ref. 19. In terms of visible-slot recognition, the proposed method gives 88.5% and 97.2% for recall and precision, respectively. Recall of the visible-slot is lower than that of the nearest-slot by 4.4% since an image of parking slot marking is dramatically degraded when its distance from the camera is larger.

Figure 23 shows the example cases where the proposed method successfully finds the parking slot markings. In Fig. 23(a)–23(d) are the example results of rectangular, uneven rectangular, diamond, and open rectangular types, respectively. Red, black, green, and blue solid lines indicate a T-junction, L-junction, Y-junction, and I-junction, respectively, and red, black, magenta, green, and blue dotted rectangles indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively. This figure shows that various types of parking slot markings presented in diverse weather (rainy, cloudy, sunny, etc.), lighting (strong backlight, shadows, etc.), and ground (mottled, bricked, handicap marked, etc.) conditions are correctly recognized by the proposed method. In particular, this method succeeded in case of the damaged marking slot marking [the third column of (a)]. The robustness against this situation is mainly because of the following two reasons. First, since parking slots are recognized using local shapes (junctions) of them, damage to other parts of the parking slot marking except junctions do not severely affect performance. Second, this method can detect junctions if they are partially damaged because one of two corners composing a junction can also generate a correct junction hypothesis.

6.3 *Computation Time Comparison*

We compared the computation time of three methods: the proposed, Jung's semiautomatic,[19] and Jung's automatic methods.[24] All methods were run in MATLAB using a 3.40 GHz Intel Core i7-2600 CPU with 4G RAM. Table 4 shows the computation time comparison among

Table 3 Performance of the proposed method using combined databases.

Type	Nearest-slot recognition					Visible-slot recognition				
	# of slots	# of TPs	# of FPs	Recall	Precision	# of slots	# of TPs	# of FPs	Recall	Precision
Rectangular	215	204	4	94.9%	98.1%	362	322	4	89.0%	98.8%
Uneven rectangular	176	165	13	93.8%	92.7%	275	248	13	90.2%	95.0%
Diamond	85	76	4	89.4%	95.0%	107	88	4	82.2%	95.7%
Open rectangular	132	120	3	90.9%	97.6%	211	187	3	88.6%	98.4%
Total	**608**	**565**	**24**	**92.9%**	**95.9%**	**955**	**845**	**24**	**88.5%**	**97.2%**

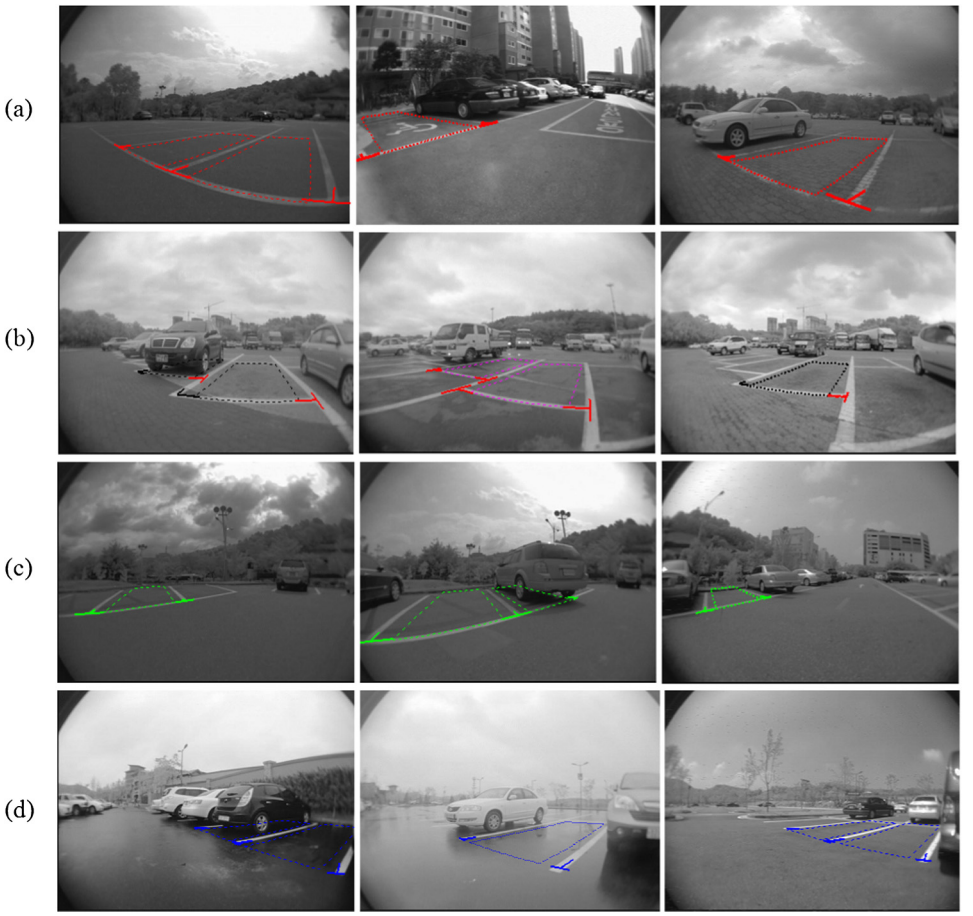

Fig. 23 Example cases where the proposed method successfully finds the parking slot markings. (a) Rectangular type. (b) Uneven rectangular type. (c) Diamond type. (d) Open rectangular type. Red, black, green, and blue solid lines indicate a T-junction, L-junction, Y-junction, and I-junction, respectively, and red, black, magenta, green, and blue dotted lines indicate a TT-slot, TL-slot, T′T′-slot, YY-slot, and II-slot, respectively.

the three methods. For a more detailed comparison, the computation times were measured by dividing the methods into several stages according to their flowcharts. On average, the proposed, Jung's semiautomatic, and Jung's automatic methods require 667 ms, 2148 and 4480 ms for a single image, respectively. This means that the proposed method is faster than Jung's semiautomatic and automatic methods by 3.2 times and 6.7 times, respectively, even if it is fully automatic. The most time consuming stage of the proposed method is corner detection and classification. This stage consumes 69.7% of the processing time since it must calculate cornerness values for corner detection and match the four templates to all corners for class and orientation estimation. In the case of Jung's semiautomatic method, junction localization stage

Table 4 Computation time comparison using the database in Ref. 19.

Proposed method	Procedure	Corner detection and classification	Junction generation	Slot generation	Type selection	Final slot determination	Total
	Computing time (ms)	465	31	169	1	1	**667**
Jung's semiautomatic method	Procedure	Initial direction establishment	Junction type recognition	Junction localization	Target position designation	—	Total
	Computing time (ms)	2	9	2136	1	—	**2148**
Jung's automatic method	Procedure	Edge detection and Hough transform	Marking line recognition	Guideline recognition	Marking line division	—	Total
	Computing time (ms)	1512	2966	1	1	—	**4480**

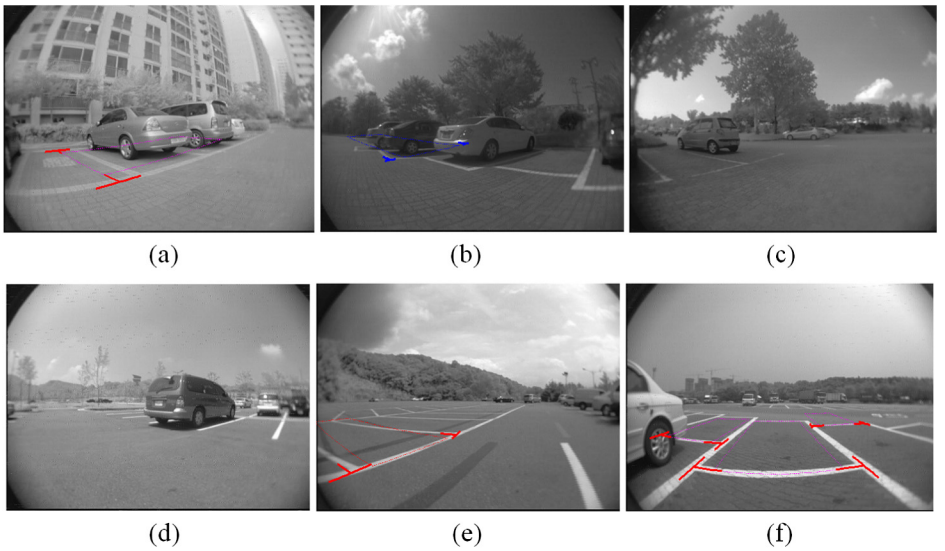

Fig. 24 Failed cases of the proposed method. (a)–(d) Corner detection procedure failed. (e) Type selection procedure failed. (f) Two false slots were generated.

takes most of the processing time (99.4%) because it should search for parameters in wide ranges due to the lack of corner information. Jung's automatic method consumes 66.2% of the computation time for the marking line recognition stage as it requires Hough space filtering. This processing time excludes the time needed to generate a bird's-eye view image since that is supposed to be implemented with specific hardware such as a field-programmable gate array (FPGA).[32]

6.4 *Limitations of the Proposed Method*

The fail cases of the proposed method can be divided into three categories. The first case occurs when the corner detection procedure fails. Corners cannot be extracted when the intensity contrast between parking slot markings and the ground is too low due to a bright ground plane [Fig. 24(a)] and strong shadow on a junction [Fig. 24(b)]. Also, corners are difficult to find in cases where parking slot markings are severely damaged [Fig. 24(c)] or image areas on the ends of distant junctions are degraded [Fig. 24(d)]. The second case occurs due to a fault of the best slot selection procedure. Figure 24(e) shows a situation where a YY-slot is misclassified as a TT-slot. This causes the correctly generated YY-slot to be eliminated. The third case occurs because of the falsely generated junctions. Figure 24(f) includes two false slots. One false slot on the left side is detected at the proper location by chance but one junction of this slot is falsely generated on the image of the vehicle wheel. The other false slot on the right side is detected outside of the parking slot marking since the T-junction is falsely generated on the image of the L-junction. These three fail cases occurred in the combined database by 35, 8, and 16 times, respectively.

6.5 *Parameter Setting*

The proposed method requires the following five parameters for corner, junction, and slot generation procedures: minimum and maximum widths of the parking slot marking line, minimum and maximum widths of the parking slot, and the orientation coincidence margin. The first four parameters can be set according to government regulations, and only the last one is experimentally set. Among them, minimum line width is the most important parameter since it determines the scale of the corner detector (σ), the radius of nonmaximum suppression (d), and the radius of circular intensity profile calculation (r). In short, 3σ, d, and r are set equal to half of the minimum line width. The effect of σ and d are mentioned with Fig. 8. If r is improperly set, circular intensity profiles are wrongly calculated and this causes the corner classification to fail. Minimum and maximum line widths and slot widths are used for junction generation and slot generation, respectively. Incorrect setting of these parameters may miss true junctions and slots or produce false junctions and slots. The orientation coincidence margin determines if the directions of the two corners or junctions coincide. A large margin increases the number of junction and slot candidates via loose pairing and a small margin decreases this number via strict pairing. This parameter is experimentally set to 25 deg in this paper. It may be retuned if the camera configuration is changed.

7 Conclusion

This paper models various parking slot markings as a hierarchical tree structure of a type-slot-junction-corner, and shows that the parking slot marking can be effectively recognized based on this structure. The proposed method climbs up the hierarchical tree structure to excessively generate corners, junctions, and slots in a bottom-up process, and climbs down the tree structure to eliminate falsely generated slots, junctions, and corners in a top-down process. This method gives a recall of 92.9% and a precision of 95.9% in terms of nearest-slot recognition and a recall of 88.5% and a precision of 97.2% in terms of visible-slot recognition while requiring a computation time of 667 ms for a single image. In the future, we are planning to develop a method that can recognize the occupancies of the detected parking slots.

Acknowledgments

This work was supported by Hyundai Motor Company.

References

1. R. Frank, "Sensing in the ultimately safe vehicle," in *Proc. Convergence International Congress and Exposition on Transportation Electronics*, Detroit, Paper No 2004-21-0055 (2004).
2. Y. Kageyama, "Look, No Hand! New Toyota Parks Itself," *CNN*, http://www.cnn.com/ (14 January 2004).
3. H. G. Jung, "Target position designation method for parking assistant system," Ph.D. Thesis, Yonsei University (2009).
4. J. Snyder, "Increasingly, cars get software upgrades to allow automatic parking," *Automotive News*, http://www.autonews.com/ (2 November 2009).
5. S. Hiramatsu et al., "Rearview camera based parking assist system with voice guidance," in *Proc. SAE 2002 World Congress and Exhibition*, Detroit, Paper No 2002-01-0759 (2002).
6. M. Furutani, "Obstacle detection systems for vehicle safety," in Proc. Convergence International Congress and Exposition on Transportation Electronics, Detroit, Paper No 2004-21-0057 (2004).
7. M. Wada, K. S. Yoon, and H. Hashimoto, "Development of advanced parking assistance system," *IEEE Trans. Indust. Electron.* **50**(1), 4–17 (2003).
8. C.-C. Huang and S.-J. Wang, "A hierarchical Bayesian generation framework for vacant parking space detection," *IEEE Trans. Circuits Syst. Video Technol.* **20**(12), 1770–1785 (2010).
9. K. Sung, J. Choi, and D. Kwak, "Vehicle control system for automatic valet parking with infrastructure sensors," in *Proc. IEEE International Conference on Consumer Electronics*, Las Vegas, pp. 567–568 (2011).
10. G. Yan et al., "SmartParking: a secure and intelligent parking system," *IEEE Intell. Transport. Sys. Mag.* **3**(1), 18–30 (2011).
11. N. Kaempchen, U. Franke, and R. Ott, "Stereo vision based pose estimation of parking lots using 3D vehicle models," in *Proc. IEEE Intelligent Vehicle Symposium*, Versailles, pp. 459–464 (2002).
12. H. Satonaka et al., "Development of parking space detection using an ultrasonic sensor," in *Proc. 13th World Congress on Intelligent Transportation Systems and Services*, London (2006).
13. U. Scheunert et al., "Free space determination for parking slots using a 3D PMD sensor," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, pp. 154–159 (2007).
14. H. G. Jung et al., "Scanning laser radar-based target position designation for parking aid system," *IEEE Trans. Intell. Transport. Syst.* **9**(3), 406–424 (2008).
15. J. K. Suhr et al., "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Mach. Vis. Appl.* **21**(2), 163–176 (2010).
16. S. H. Jeong et al., "Low cost design of parallel parking assist system based on an ultrasonic sensor," *Int. J. Autom. Technol.* **11**(3), 409–416 (2010).
17. M. R. Schmid et al., "Parking space detection with hierarchical dynamic occupancy grids," in *Proc. IEEE Intelligent Vehicle Symposium*, Baden-Baden, pp. 254–259 (2011).
18. C. Unger, E. Wahl, and S. Ilic, "Parking assistance using dense motion-stereo: real-time parking slot detection, collision warning and augmented parking," *Mach. Vis. Appl.*, Published online (2011).
19. H. G. Jung, Y. H. Lee, and J. Kim, "Uniform user interface for semi-automatic parking slot marking recognition," *IEEE Trans. Vehic. Technol.* **59**(2), 616–626 (2010).
20. H. G. Jung et al., "Structure analysis based parking slot marking recognition for semi-automatic parking system," *Lect. Notes Comput. Sci.* **4109**, 384–393 (2006).
21. J. Xu, G. Chen, and M. Xie, "Vision-guided automatic parking for smart car," in *Proc. IEEE Intelligent Vehicles Symposium*, Dearborn, pp. 725–730 (2000).
22. H. G. Jung et al., "3D vision system for the recognition of free parking site location," *Int. J. Autom. Technol.* **7**(3), 361–367 (2006).
23. Y. Tanaka et al., "Development of image recognition for a parking assist system," in *Proc. 13th World Congress on Intelligent Transportation Systems and Services*, London, pp. 1–7 (2006).
24. H. G. Jung et al., "Parking slot markings recognition for automatic parking assist system," in *Proc. IEEE Intelligent Vehicles Symposium*, Tokyo, pp. 106–113 (2006).
25. Ford 2012 Escape, http://www.ford.com/suvs/escape/features/#page=Feature11 (2011).
26. BMW 5 Series Sedan, http://www.bmw.com/com/en/newvehicles/5series/sedan/2010/showroom/comfort/parking_assistant.html (2011).
27. Lexus LS, http://www.lexus.com/models/LS/features/exterior/advanced_parking_guidance_system.html (2011).
28. H. G. Jung et al., "Radial distortion refinement by inverse mapping-based extrapolation," in *Proc. 18th International Conference on Pattern Recognition*, Hong Kong, pp. 675–678 (2006).
29. P. H. Batavia, D. A. Pomerleau, and C. E. Thorpe, "Detecting overtaking vehicles with implicit optical flow," Technical Report CMU-RI-TR-97-28, Robotics Institute, Carnegie Mellon University (1997).
30. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, Manchester, pp. 147–151 (1988).
31. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Pearson Prentice Hall, New Jersey (2008).
32. F. Cancare et al., "A bird's eye view of FPGA-based evolvable hardware," in *Proc. NASA/ESA Conference on Adaptive Hardware and Systems*, Los Alamitos, pp. 169–175 (2011).

Jae Kyu Suhr received his BS degree in electronic engineering at Inha University, Incheon, Korea, in 2005, and the MS and PhD degrees in electrical and electronic engineering at Yonsei University, Seoul, Republic of Korea, in 2007 and 2011, respectively. He is currently working as a postdoctoral researcher in Research Institute of Automotive Electronics and Control, Department of Automotive Engineering, Hanyang University, Seoul, Republic of Korea. His current research interests include computer vision, image analysis, and pattern recognition for intelligent vehicle and visual surveillance.

Ho Gi Jung received the BE, ME, and PhD degrees in electronic engineering from Yonsei University, Seoul, Republic of Korea, in 1995, 1997, and 2008, respectively. He was with MANDO Corporation Global R&D H.Q., from 1997 to April 2009. He developed environmental recognition systems for intelligent parking assist system, collision warning and avoidance system, and active pedestrian protection system. From May 2009 to February 2011, he was with Yonsei University as a full-time researcher and research professor. He researched computer vision applications for intelligent surveillance systems and biometric systems. Since March 2011, he has been with Hanyang University as an assistant professor. He is researching recognition systems for intelligent vehicles. His interests are recognition system for intelligent vehicle, next generation vehicle, computer vision applications, and pattern recognition applications.