2012 15th International IEEE Conference on Intelligent Transportation Systems
Anchorage, Alaska, USA, September 16-19, 2012

Fully-automatic Recognition of Various Parking Slot Markings in Around View Monitor (AVM) Image Sequences

Jae Kyu Suhr and Ho Gi Jung

Abstract— **This paper proposes a novel fully-automatic method for recognizing various parking slot markings in image sequences acquired by an Around View Monitor (AVM) system which is gaining popularity as a parking-aid product. The proposed method utilizes an approach which finds parking slots in AVM image sequences using a simple detector and combines sequentially acquired slots rather than using a sophisticated heavy detector in order to achieve the robustness against diverse practical situations. Parking slots are first detected in the current image using the hierarchical tree structure based approach presented in our previous paper [1]. Then, the current positions of previously detected parking slots are predicted using a transformation between consecutive images, and the parking slots detected in the current image and predicted from previous images are combined. The resulting parking slots are clustered according to their types and orientations, and the clusters which contain more than a predetermined number of slots are selected as final parking slots. The proposed method was evaluated using 10 AVM image sequences that include 134 parking slots, and demonstrated a detection rate of 95% with only three false detections.**

I. Introduction

Since the needs and interests in automatic parking systems are increasing rapidly, various parking space designation approaches have been proposed [2]. Among these approaches, the parking slot marking based one has the following advantages: 1) It can be combined with the user-interface based approach [3] in order to reduce repetitive driver operations for parking space localization. 2) Unlike the free-space based approach [4], it is able to more accurately align parking spaces because its performance does not depend on the existence and poses of adjacent vehicles. 3) Its computational costs are substantially lower than the free-space based approach using binocular or motion stereo [5]. The parking slot marking based approach has limitations as it can only be used in cases where parking slot markings are present. However, this approach is still useful since drivers need automatic parking systems in complex urban situations where parking slot markings are usually present.

The parking slot marking based approach can be categorized into semi-automatic and fully-automatic methods. In semi-automatic methods, both a one-touch based method for a single type of parking slot marking [6] and a two-touch based method for various types of parking slot markings have been proposed [7]. Various fully-automatic methods which utilize color information [8], stereo cameras [9], line detectors [10], [11] have also been presented. However, all of these fully-automatic methods can only deal with a single type of parking slot marking. To overcome this drawback, our previous work presented a fully-automatic parking slot marking recognition method based on an hierarchical tree structure [1]. Since most of the previous works, including ours, were targeted for images acquired by rearview cameras, they are able to deal with the small number of parking slots presented in a restricted view.

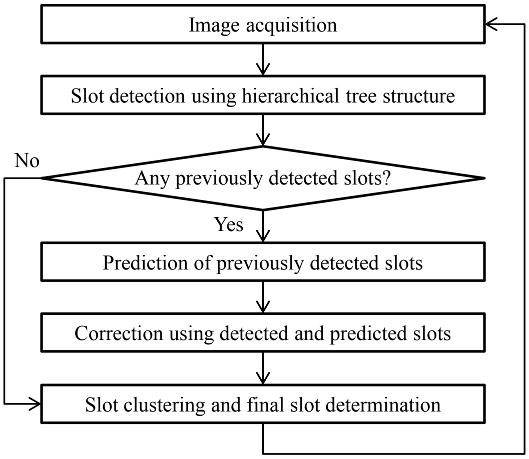


Image acquisition
Slot detection using hierarchical tree structure
No
Any previously detected slots?
Yes
Prediction of previously detected slots
Correction using detected and predicted slots
Slot clustering and final slot determination

Figure 1. Flowchart of the proposed method.

Around View Monitor (AVM) systems which offer 360° surrounding view images have been produced by many automobile companies as parking-aid products [12-15]. Unlike rearview camera images, AVM images include a very wide view around vehicles, and many different types of parking slots may appear and disappear if AVM images are acquired from moving vehicles. Since previous fully-automatic methods find these situations difficult to handle, this paper proposes a novel fully-automatic method which can effectively recognize various parking slot markings in AVM image sequences. The proposed method utilizes an approach which finds parking slots in AVM image sequences using a simple detector and combines sequentially acquired slots in order to achieve robustness against various practical situations rather than using a sophisticated heavy slot detector. Fig. 1 shows the flowchart of the proposed method. This method first detects various parking slots in the current image using a hierarchical tree structure. If slots are detected in the previous images, their current positions are predicted, and slots in the current image are corrected by combining currently detected slots and predicted slots. Lastly, the remaining slots are clustered according to type and orientation, and final slots are determined based on the clustering result. In

*This work was supported by Hyundai Motor Company.

J. K. Suhr is with the Research Institute of Automotive Electronics and Control, Hanyang University, Seoul, Korea.

H. G. Jung is with the Department of Automotive Engineering, Hanyang University, Seoul, Korea (email: hogijung@hanyang.ac.kr).

978-1-4673-3063-3/12/$31.00 ©2012 IEEE

experiments, the proposed method was evaluated using 10 AVM image sequences which include 134 parking slot markings, and showed a detection rate of 95% with only three false detections.

II. Revisiting the Hierarchical Tree Structure Based Parking Slot Marking Recognition Method

This section briefly reviews the hierarchical tree structure based parking slot marking recognition method presented in our previous paper [1]. This method deals with the four most commonly appearing types of parking slot markings (rectangular, slanted rectangular, diamond, and open rectangular types) as shown in Fig. 2(a). These four types of parking slot markings consist of four types of slots (TT-slot, TL-slot, YY-slot, and II-slot) as shown in Fig. 2(b), and each slot is composed of two junctions as shown in Fig. 2(b) with red rectangles. Junctions composing of four types of slots can be categorized into four types (T-junction, L-junction, Y-junction, and I-junction) as shown in Fig. 2(c), and each junction consists of two corners as shown in Fig. 2(c) with red circles. These corners can be divided into four types (60°-corner, 90°-corner, 120°-corner, and 270°-corner) according to the arc angles of the ground plane (black) as shown in Fig. 2(d). This is the hierarchical tree structure of parking slot markings.

Based on this structure, our previous method detects and classifies corners and produces junctions by pairing two corners. Slots are generated by joining two appropriate junctions, and the final type and orientation of parking slot marking is determined by combining all the information from the generated slots. This is a bottom-up process. After that, the top-down process deletes falsely generated slot-junction-corners by checking whether the slots coincide with the type and orientation of the finally determined parking slot markings. This method was evaluated with rearview camera images which include various parking slot markings, and showed a better performance than the recent semi-automatic method in [7], even though this method is fully-automatic.

III. Parking Slot Marking Recognition in AVM Image Sequences

Since the hierarchical tree structure based method was developed for rearview camera images, it works well only in situations where a small number of parking slots are presented in a restricted view. Thus, we should consider the properties of AVM image sequences in order to successfully recognize parking slot markings in images acquired from AVM systems. Fig. 3(a) and (b) show example images captured by rearview camera and AVM systems, respectively. In this figure, the rearview image includes only three parking slots but the AVM image contains about 10 parking slots, and the positions of parking slots in the AVM images are more diverse than those of the rearview images. If AVM images are acquired from a moving vehicle, the number of parking slots and diversity of their positions will be greatly increased. Furthermore, junctions of parking slots can be occluded (Fig. 3(b) with a green arrow) and their shapes can be distorted when they are located near the borderline of images from different cameras (Fig. 3(b) with a red arrow) or quite far from the camera location (Fig. 3(b) with a yellow arrow).

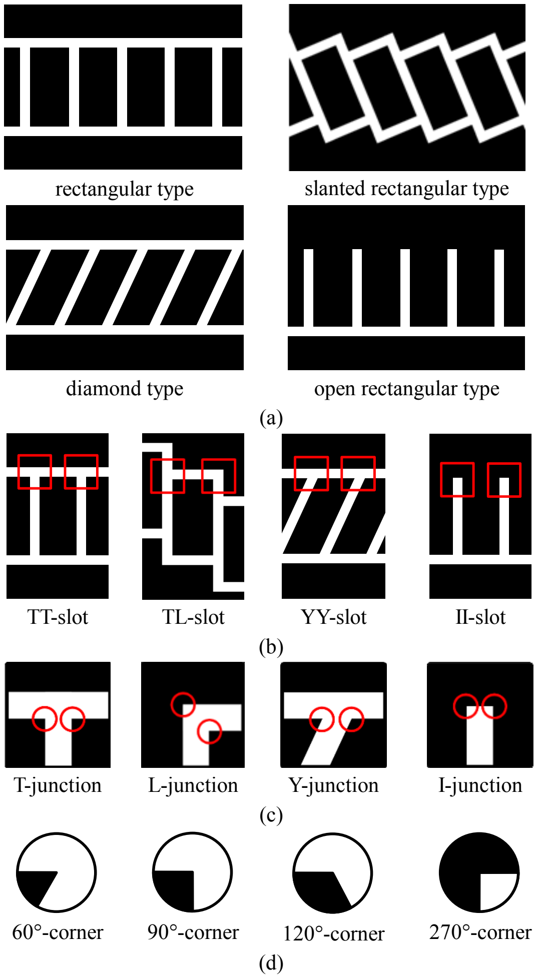


Figure 2. Hierarchical tree structure of parking slot markings. (a) parking slot markings (b) slots (c) junctions (d) corners.

To robustly recognize parking slots against these situations, it is possible to use a sophisticated method. However, this will be a burden to the real-time automobile system since a sophisticated method is usually costly. Therefore, we utilize an approach which applies a simple method to AVM image sequences and then combines sequentially detected parking slots to achieve reliable results. The proposed method consists of the following four procedures: 1) a parking slot detection procedure using a hierarchical tree structure; 2) a prediction procedure for estimating current positions of previously detected parking slots; 3) a correction procedure using detected and predicted parking slots; and 4) a clustering procedure to determine the final parking slots.

A. Detecting Parking Slots using an Hierarchical Tree Structure

In order to detect parking slots using an hierarchical tree structure, corners are detected via a Harris corner detector [16] and classified into four types in Fig. 2(d) based on a

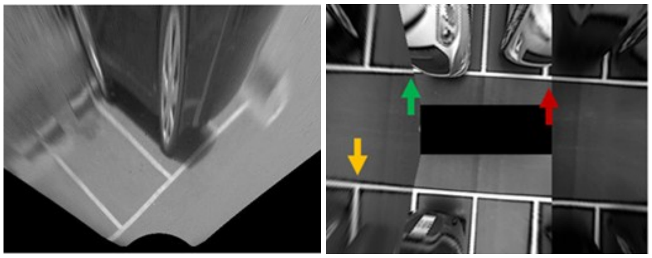


Figure 3. Comparison between (a) a rear-view image and (b) an AVM image.

template matching technique using circular intensity profiles. After that, junctions in Fig. 2(c) are generated by pairing two corners, and the slots in Fig. 2(b) are produced by joining two junctions. Fig. 4(a) shows an example of a parking slot detection result. In this figure, red, cyan, green, and blue lines indicate the TT-slot, TL-slot, YY-slot, and II- slot, respectively. Since this approach excessively generates parking slots so as not to lose the correct ones, its initial results may include some false detections. Fig. 4(a) contains several overlapped parking slots, but it is impossible for multiple parking slots to be overlapped in real situations. Thus, we should select the most appropriate one among the overlapped parking slots. To determine the overlapping of two slots, the Jaccard coefficient [17] in (1) is utilized.

$$J(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{1}$$

where $J(S_i, S_j)$ indicates the Jaccard coefficient between two rectangles formed by the i-th slot (S_i) and the j-th slot (S_j). If this value is larger than a predetermined threshold (T_1), two slots are considered as overlapped. When there are overlapped slots, the normalized average intensity values (NAIV) of two slots are calculated as (2) and the slot with a NAIV larger than that of the other is selected.

$$NAIV_i = \frac{1}{MAX(I)} \left\{ \frac{1}{N} \sum_{(x,y)\subset S_i} I(x,y) \right\} \tag{2}$$

where $NAIV_i$ and $MAX(I)$ are the NAIV of S_i and the maximum intensity value of image I, respectively. N and (x, y) are the number of pixels in the skeleton of the parking slot entrance and their locations in the x and y axes, respectively. Since parking slot markings are much brighter than the ground plane and pixels near the center of the parking slot marking lines tend to have larger intensity values than the boundary pixels [6], the slot located at the most appropriate position has the largest NAIV compared with the other slots. Fig. 4(b) shows the results after removing the overlapped slots from Fig. 4(a). It can be seen that all the correctly detected slots are retained except one false detection (TL-slot with a cyan line).

B. Predicting current positions of previously detected parking slots

To predict the current positions of previously detected parking slots, this paper estimates the transformation between ground plane images captured by the AVM system at different vehicle locations, and transforms the parking slot positions in previous images into the current image. Generally, if two cameras at different locations take the same ground plane, two ground plane images are related with a homography (projective transformation) [18] as (3).

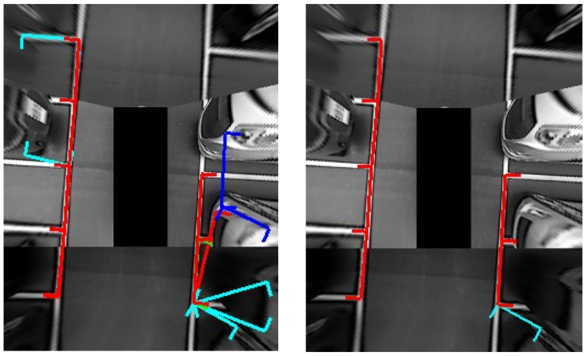


Figure 4. An example result of parking slot detection using the hierarchical tree structure. (a) initial detection result (b) result after removing overlapped slots.

$$\mathbf{x}'_h = H\mathbf{x}_h \tag{3}$$

where $\mathbf{x}_h$ and $\mathbf{x}'_h$ are the images of the same 3-D point on the ground plane in the previous and current images, respectively. They are in homogeneous coordinates. H is a 3x3 matrix which represents an homography. However, the homography between ground plane images can frequently be misestimated due to its high degree of freedom because ground plane images have less texture while images of the front and rear of automobiles include an abundance of texture in AVM environments. Therefore, this paper utilizes a 3-parameter similarity transformation whose degree of freedom is much less than a homography. There is only a scale difference between the ground plane in the AVM images and the ground plane in the real world if four cameras are correctly calibrated and the ground plane is approximately flat. Thus, the ground planes in two consecutive AVM images are related with two-dimensional translation and rotation as (4).

$$\mathbf{x}'_{in} = R\mathbf{x}_{in} + \mathbf{t} \tag{4}$$

where $\mathbf{x}_{in}$ and $\mathbf{x}'_{in}$ are the images of the same 3-D points on the ground plane in previous and current images, respectively. They are in inhomogeneous coordinates. R and $\mathbf{t}$ are a 2x2 rotation matrix and 2x1 translation vector, respectively. There are only three parameters in this transformation: translations in x and y axes (t_x and t_y), and rotation angle (θ). These three parameters can be estimated via Procrustes analysis [19] in (5)-(7).

$$t_x = \frac{1}{K} \left\{ \sum_{i=1}^{K} (x'_i - x_i) \right\} \tag{5}$$

$$t_y = \frac{1}{K} \left\{ \sum_{i=1}^{K} (y'_i - y_i) \right\} \tag{6}$$

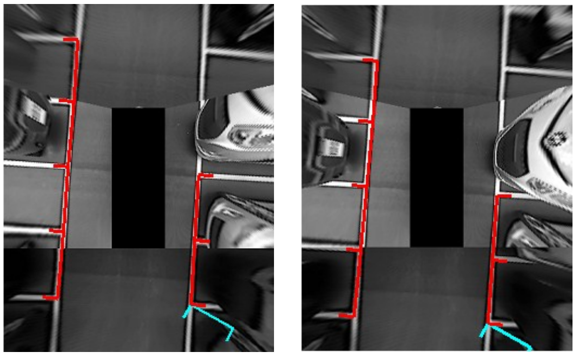


Figure 5. Detected and predicted parking slot positions. (a) parking slots detected in the previous image (b) predicted positions of parking slots in the current image.

$$\theta = \tan^{-1}\left\{ \sum_{i=1}^{K}\left(\bar{x}_i y_i' - \bar{y}_i x_i'\right) \Big/ \sum_{i=1}^{K}\left(\bar{x}_i x_i' + \bar{y}_i y_i'\right) \right\} \quad \text{where } \bar{x}_i = x_i + t_x \text{ and } \bar{y}_i = y_i + t_y \tag{7}$$

where K indicates the number of corresponding feature points, and (x_i, y_i) and (x'_j, y'_j) are positions of corresponding feature points in previous and current images, respectively. This paper utilizes the interest points produced by a Harris corner detector during the parking slot detection procedure as feature points and matches them via normalized cross correlation (NCC) in (8). The reuse of corners can save on computational costs.

$$NCC(i, j) = \frac{1}{N-1} \cdot \frac{\sum_{x,y}\left\{I(x_i, y_i) - m_i^I\right\}\left\{I'\left(x_j', y_j'\right) - m_j^{I'}\right\}}{\sigma_i^I \sigma_j^{I'}} \tag{8}$$

where $NCC(i, j)$ is the NCC value between the i-th corner in I and the j-th corner in I', and m^I_i, $m^{I'}_j$, σ^I_i, and $\sigma^{I'}_j$ are means and standard deviations calculated from neighborhood pixels around the i-th corner in I and the j-th corner in I', respectively. Since it is inevitable for the feature point matching result to be contaminated by outliers, the final parameters are estimated by RANSAC [20] which is one of the most widely used robust estimators. Fig. 5(a) shows the parking slots detected in the previous image and Fig. 5(b) shows their predicted positions in the current image. It can be seen that the positions of parking slots are successfully predicted.

C. Correcting parking slots by combining detection and prediction results

Once positions of parking slots are detected and predicted, the correction procedure is conducted by combining all the detected and predicted parking slots. For this task, we first check whether the detected and predicted slots are overlapped or not using the Jaccard coefficient in (1). Overlapping situations are classified into the following three cases according to the Jaccard coefficient and two predetermined thresholds, T_1 and T_2 ($T_1 < T_2$): 1) No overlapping, 2) Overlapping between the same parking slot, 3) Overlapping between different parking slots. Detailed procedures for these three cases are as follows:

Algorithm I

```
for all slots $S_i$ do
    if $S_i$ is detected in the $t$-th image
        $l_{1:t}(S_i) = l_t(S_i) + l_{1:t-1}(S_i) - l_o(S_i)$
    else
        $l_{1:t}(S_i) = l_{1:t-1}(S_i)$
    endif
endfor
```

[Case I] $J(S_i, S_j) < T_1$: If the Jaccard coefficient between the i-th slot detected in the current image (S_i) and the j-th slot predicted from previous images (S_j) is less than T_1, two slots are considered not to be overlapped. If S_i is not overlapped with any previously detected slots, S_i is identified as a newly detected slot. In this case, $p(S_i|z_{1:t})$ is set to $p(S_i|z_t)$, where $p(S_i|z_{1:t})$ is a posterior probability that S_i is a true parking slot after observing the detection results up to the t-th image and $p(S_i|z_t)$ is an experimentally calculated posterior probability.

[Case II] $J(S_i, S_j) \geq T_2$: If the Jaccard coefficient is greater than or equal to T_2, it can be considered that the same slot is repetitively detected in sequential images. In this case, we correct the position of this parking slot by selecting one slot position which gives a larger NAIV in (2) than the other, and then update the posterior probability. Since the posterior probability in parking slot detection tasks can be modeled by a binary estimation problem with static state, it can be updated as (9) in log odds representation [21].

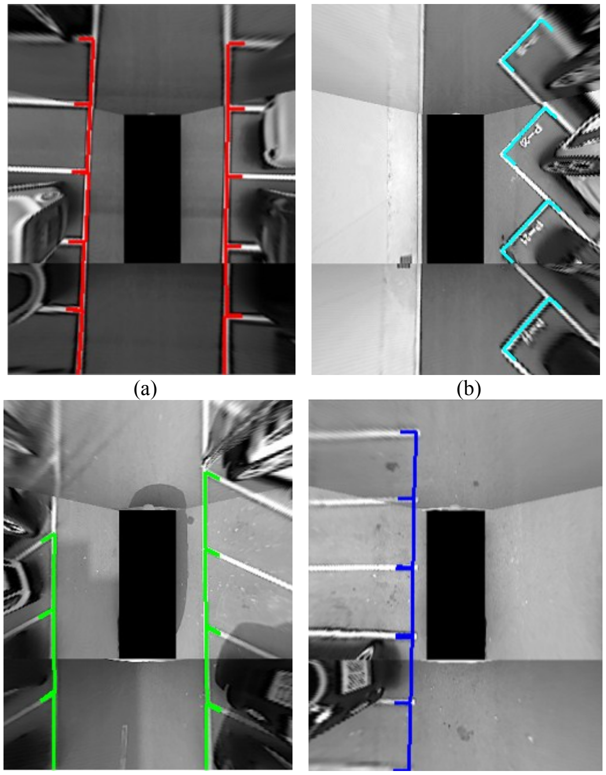
$$\begin{aligned} l_{1:t}(S_i) &= \log\frac{p(S_i \mid z_{1:t})}{1-p(S_i \mid z_{1:t})} \\ &= \log\left\{\frac{p(S_i \mid z_t)}{1-p(S_i \mid z_t)} \cdot \frac{p(S_i \mid z_{1:t-1})}{1-p(S_i \mid z_{1:t-1})} \cdot \frac{1-p(S_i)}{p(S_i)}\right\} \\ &= \log\frac{p(S_i \mid z_t)}{1-p(S_i \mid z_t)} + \log\frac{p(S_i \mid z_{1:t-1})}{1-p(S_i \mid z_{1:t-1})} - \log\frac{p(S_i)}{1-p(S_i)} \\ &= l_t(S_i) + l_{1:t-1}(S_i) - l_o(S_i) \end{aligned} \tag{9}$$

where $p(S_i)$ is a prior probability that S_i is a parking slot. The posterior probability is easily recovered from the log odds ratio via (10).

$$p(S_i \mid z_{1:t}) = 1 - \frac{1}{1+\exp\left\{l_{1:t}(S_i)\right\}} \tag{10}$$

Based on the updating rule in (9), the log odd ratio, $l_{1:t}(S_i)$ is recursively calculated using Algorithm I.

[Case III] $T_1 \leq J(S_i, S_j) < T_2$: If the Jaccard coefficient is between T_1 and T_2, two slots are considered to be overlapped but they are not a repetitive detection of the same slot. Since two different parking slots are unable to be overlapped in real situations, only one of the two slots is selected by considering their “slotness”. As described previously, we have already had two “slotness” scores: NAIV in (2) and $p(S_i|z_{1:t})$ in (10).



(a) (b)

(c) (d)

Figure 6. Example results of the proposed method in various parking slot marking environments. (a) rectangular type (b) slanted rectangular type (c) diamond type (d) open rectangular type.

Because these two scores are normalized between 0 to 1, we fuse them using a simple sum rule [22] which is one of the popular score fusion rules. Then, one slot which has a larger summed score is selected and the other is eliminated.

D. Determining final parking slots via slot clustering

To determine the final parking slots, we cluster the remaining non-overlapped parking slots according to their types and orientations. Assuming that there is only one type of parking slot marking in a single AVM image sequence, the parking slot type of the cluster which includes the largest number of slots is considered as the final parking slot marking type. Then, the clusters which are the same type as the final parking slot marking and include more than two slots are selected as final parking slots. This procedure can eliminate false detections such as the TL-slot in Fig. 5 with a cyan line since this type of slot is different from the type of final parking slot marking and is the only slot which forms a cluster. If there is a single slot which coincides with the type of the final parking slot marking but is the only slot that forms a cluster, this parking slot is not displayed to drivers but internally stored because it may be a newly appearing slot.

IV. Experiments

The proposed method was evaluated using 10 AVM image sequences acquired by a development version of Hyundai Motor Company's AVM system [23]. AVM images were cropped to have a view up to 3.5 meters from the front, rear, left, and right ends of the vehicle. Fig. 6(a)-(d) show examples of parking slot detection results in AVM image sequences which include rectangular, slated rectangular, diamond, and open rectangular types, respectively. Table I shows the detailed description of the used AVM image database and detection results of the proposed method. In this table, a slot is considered to be detected if its location and orientation are correctly estimated until it disappers from an AVM image. Any slots that do not follow this definition are treated as false detections. Detection rate refers to a portion between the number of true detections and the number of slots. As a summary, the proposed method detects 127 slots among 134 slots presented in 2545 AVM images while producing only three false detections. Thus, the detection rate of the proposed method is 95%. Based on Fig. 6 and Table I, it can be said that the proposed method can effectively recognize various parking slot markings in AVM image sequences. Fig. 7 shows the slots detected in the current images (first column) and the slots produced by combining currently and previously detected slots (second column). In this figure, (a)-(d) are sequential AVM images. The detection results using only current images in the first column of Fig. 7 include several missing slots (yellow arrows in (a) and (c)) and a falsely detected slot (green arrow in (b)). But the detection results produced by combining currently and previously detected slots in the second column of Fig. 7 succeeds to recognize all visible parking slots.

TABLE I. Description of the Database and Performance of the Proposed Method

DB No.	**Parking slot marking type**	**# of images**	**# of slots**	**# of true detections**	**# of false detections**	**Detection rate**
1,2	Rectangular	958	49	49	0	100%
3,4,5	Slanted rectangular	626	28	27	3	96%
6,7,8	Diamond	498	29	25	0	86%
9,10	Open rectangular	463	28	26	0	93%
Total		2545	134	127	3	95%

V. Conclusion and Future Work

This paper proposes a novel method for recognizing various parking slot markings in images acquired by AVM systems that are becoming increasingly popular as parking-aid systems. The proposed method is suitable for real-time automobile systems since it applies a simple parking slot detector to image sequences and then combines all detection results rather than utilizing a sophisticated heavy slot detector in order to recognize occluded slots and slots with low resolution. Experimental results show that this method successfully detects 95% of various types of parking slots. In the future, we will develop a method that can recognize the

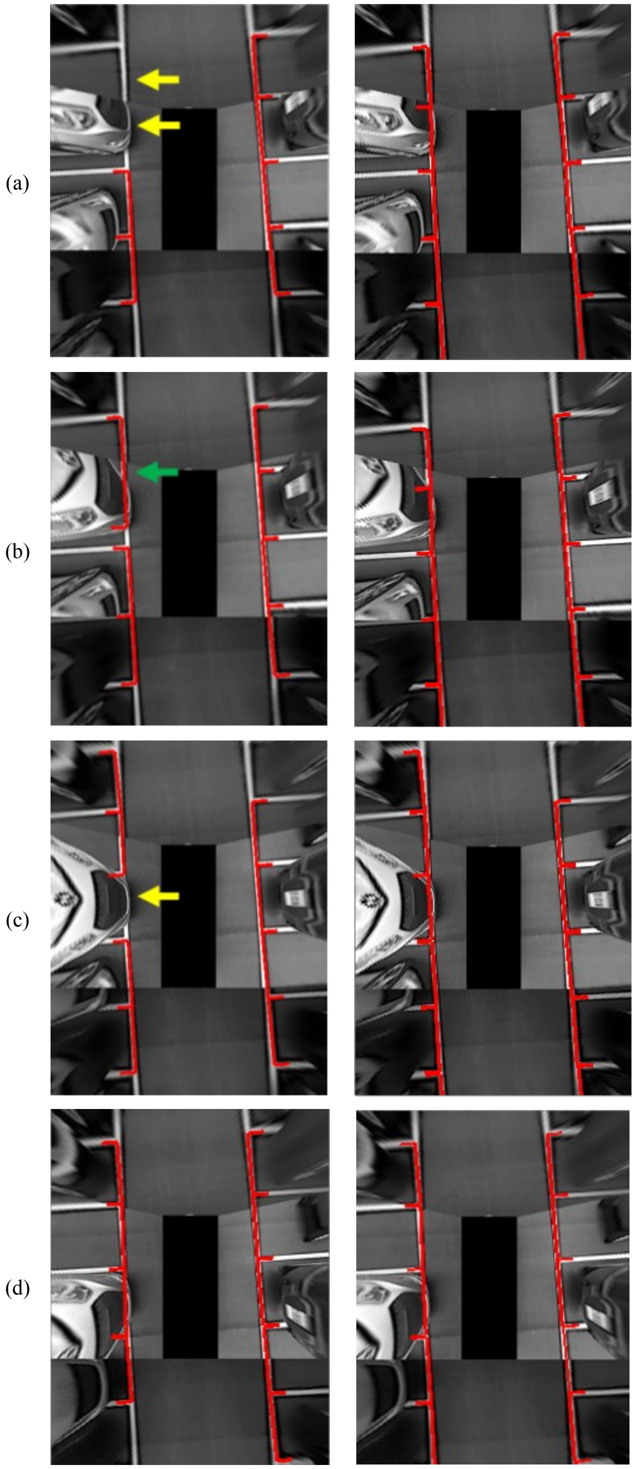


Figure 7. Parking slot detection results. (a) to (d) are sequential AVM images. The first and second columns show the parking slots detected in current images and parking slots produced by combining currently and previously detected slots, respectively.

occupancies of the detected parking slots and deal with situations where various types of parking slots are presented in the same AVM image sequence.

References

[1] J. K. Suhr and H. G. Jung, "Full-Automatic Recognition of Various Parking Slot Markings Using a Hierarchical Tree Structure," *IEEE Transactions on Intelligent Transportation Systems*, Submitted for publication, Jan. 2012.

[2] H. G. Jung, "Target Position Designation Method for Parking Assistant System," Ph.D. dissertation, School of Electrical and Electronic Engineering, Yonsei University, Seoul, 2009.

[3] S. Hiramatsu, A. Hibi, Y. Tanaka, T. Kakinami, Y. Iwata, and M. Nakamura, "Rearview Camera Based Parking Assist System with Voice Guidance," SAE Paper 2002-01-0759.

[4] H. Satonaka, M. Okuda, S. Hayasaka, T. Endo, Y. Tanaka, and T. Yoshida, "Development of Parking Space Detection Using an Ultrasonic Sensor," *In Proceedings of the 13th World Congress on Intelligent Transportation Systems and Services*, Oct. 2006.

[5] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Machine Vision and Applications*, vol. 21, no. 2, pp. 163-176, Feb. 2010.

[6] H. G. Jung, D. S. Kim, P. J. Yoon, and J. Kim, "Structure Analysis Based Parking Slot Marking Recognition for Semi-Automatic Parking System," *Lecture Notes in Computer Science*, vol. 4109, pp. 384–393, Aug. 2006.

[7] H. G. Jung, Y. H. Lee, and J. Kim, "Uniform User Interface for Semiautomatic Parking Slot Marking Recognition," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 616-626, Feb. 2010.

[8] J. Xu, G. Chen, and M. Xie, "Vision-guided automatic parking for smart car," *In Proceedings of the IEEE Intelligent Vehicles Symposium*, Oct. 2000, pp. 725–730.

[9] H. G. Jung, D. S. Kim, P. J. Yoon, and J. H. Kim, "3D vision system for the recognition of free parking site location," *International Journal of Automotive Technology*, vol. 7, no. 3, pp. 361–367, May 2006.

[10] Y. Tanaka, M. Saiki, M. Katoh, and T. Endo, "Development of image recognition for a parking assist system," *In Proceedings of the 13th World Congress on Intelligent Transportation Systems and Services*, Oct. 2006.

[11] H. G. Jung, D. S. Kim, P. J. Yoon, and J. Kim, "Parking slot markings recognition for automatic parking assist system," *In Proceedings of the IEEE Intelligent Vehicles Symposium*, Jun. 2006, pp. 106–113.

[12] Nissan - Around View Monitor System, http://www.nissan-global.com /EN/TECHNOLOGY/OVERVIEW/avm.html, Mar. 2012.

[13] Honda - Multi-View Camera System, http://world.honda.com/news /2008/4080918Multi-View-Camera-System/, Mar. 2012.

[14] BMW - Surround View, http://www.bmw.com/com/en/insights /technology/connecteddrive/2010/convenience/parking/surround_view_information.html, Mar. 2012.

[15] Hyundai - Around View Monitor System, http://www.hyundai.com/kr /ShowRoom/ShowRoom.aspx?code=Grandeur&sNum=3&types=01, Mar. 2012.

[16] C. Harris and M. Stephens, "A combined corner and edge detector," *In Proceedings of the 4th Alvey Vision Conference*, Sep. 1988, pp. 147–151.

[17] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey", *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294-307, Mar. 2005.

[18] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision - 2nd Edition*, Cambridge University Press, Cambridge, 2003.

[19] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, Oxford University Press, New York, 2004.

[20] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, MIT Press, Cambridge, 2005.

[22] A. K. Jain, K. Nandakumar, and A. Ross, "Score Normalization in Multimodal Biometric Systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270- 2285, Dec. 2005.

[23] Hyundai Motor Company, http://worldwide.hyundai.com/WW/, Mar. 2012.