Vision-based Approach in Finding Multitype Parking Stall Entrance

Changmu Seo[1], Joongsik Kim[1], Yunhee Lee[2], Whoi-Yul Kim[1]

[1]Department of Electronics and Computer Engineering, Hanyang University, 222 Wangsimri-ro, Seongdong-gu, Seoul, Korea
[2]CAMMSYS Corporation, 26, Venture-ro 100beon-gil, Yeonsu-gu, Incheon, Korea
{cmseo, jskim}@vision.hanyang.ac.kr, leeyunis@cammsys.net, wykim@hanyang.ac.kr

Abstract: This paper proposes a novel method of finding the entrance of the parking stall for PAS to cope with multitype parking stalls while drving in the parking aisle. For this purpose, we propose to categorize parking stalls into two types and three shapes for each type. Utilizing the information along the parking lines surrounding the parking stall, the type and the location information of the parking stall can be obtained to guide the vehicle. Experimental results show that the proposed method can successfully detect various types of parking stalls enough to park a vehicle via autonomous valet parking.

Keywords: Parking assist system; Around view monitor; Parking stall detection; Parking stall entrance; Valet parking

1 Introduction

A large number of vision-based methods using an around view monitor images (AVM) have been proposed to find empty parking stalls in parking assist system (PAS) [1]-[4]. However, previously proposed methods can detect an only basic form of parking stall such as the 'closed rectangular shape' although there are some diffent types and shapes of parking stalls often used in the real world as illustrated in figure 2. Besides, it is difficult to detect the parking stall when the marking of the parking stall is partially occluded or erased. In this paper, we propose a robust parking stall detection method than can handle several types of parking stalls including basic types. With the proposed method we first extract line information of parking lines encompassing the parking stalls. From the line information, cross-point type features or end-point type features which define the parking stall entrance are generated and combined to detect parking stalls regardless of types or shapes of the parking stall. Experimental results show that the proposed method can successfully detect various types of parking stalls by showing a 96.37% recall and a 97.32% precision.

2 Proposed Method

2.1 Parking Layout Terminology

Figure 1 illustrates an imaginary parking lot to describe or define the terminology often used in the parking lot management systems. A vehicle is driving along the *parking aisle*, meaning the portion of the parking lot devoted to providing immediate access to the parking stalls or spaces [5]. In this paper, we define entrance part of a parking stall as the *entrance of parking stall*, and opposite part of entrance of parking stall as the *rear-end of parking stall*. A blue line with a dot and an arrow on

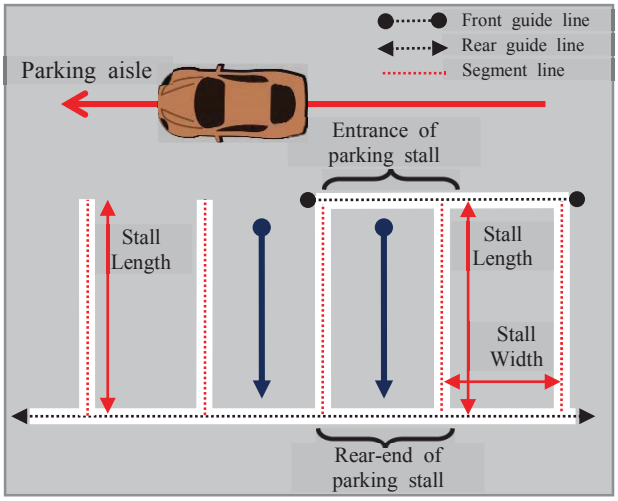

Figure 1 Imaginary parking lot

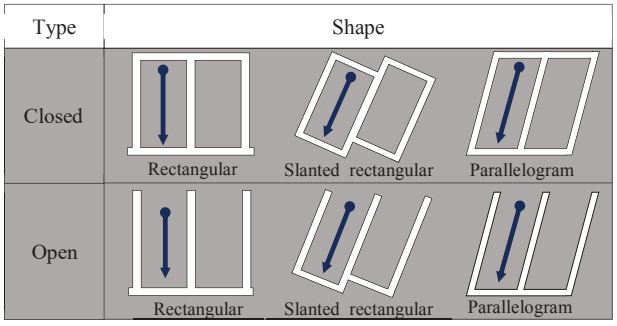

Figure 2 Proposed parking stall grouping based on their types and shapes

ends indicates the travel direction of the vehicle in the parking stall. When parking, the vehicle moves from the dot-head to the arrow-head. Black and red dotted lines in figure 1 indicate markings of the guide and segment lines, respectively, which form the marking of a parking stall. Guide line can be divided into two kinds depending on the distance from the parking aisle. We first define a guide line near the parking aisle as a *front guide line* as shown by the dotted line with dots on both ends in figure 1. Next, *rear guide line* is defined as a guide line away from the parking aisle, as shown by the dotted line with arrows on both ends in figure 1. The dimension of the parking stall is defined by *stall length* and *stall width*.

We then categorize parking stalls into two types depending on the form of the parking stall entrance. The first type of parking stall is a *closed type* where the front guide line exists at the parking stall entrance. The second type, called an *open type*, is a parking stall with no front

978-1-5386-6067-6/18/$31.00 ©2018 IEEE

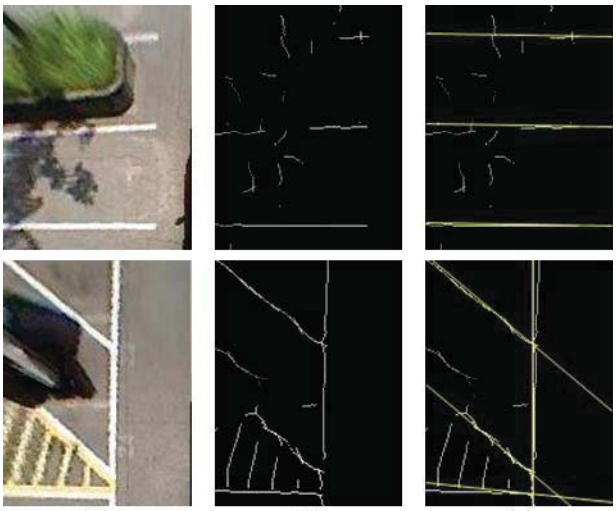

Figure 3 Line candidate extraction **(a)** Input AVM image **(b)** Line feature map **(c)** Line candidate extraction results

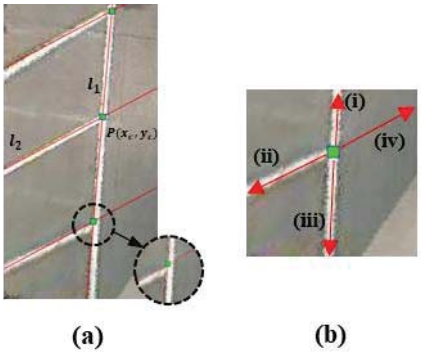

Figure 4 Cross-point type feature extraction **(a)** Line candidate extraction results **(b)** Intersection correction result

guide line. Figure 2 shows the proposed parking stall categorization and the example of the parking stall shapes belonging to their categories. Similarly, a blue line with an arrow in figure 2 indicates the travel direction of vehicle in the parking stall.

2.2 Line Candidate Extraction

Typically, a Sobel [7] or Canny edge detector [8] is commonly used to obtain the line feature map in the AVM images. In this paper, we use a line filter [6] that is robust to shadows and capable of obtaining line information at the center of the parking stall marking. Then pixel values corresponding to the center of the actual parking stall markings in AVM images are used for Radon Transform [1] to detect straight lines reliably. Figure 3 (b) shows the line feature maps obtained from the AVM images of figure 3 (a). Figure 3 (c) shows the line candidate extraction results.

2.3 Cross-point type feature Generation

This section defines a cross-point type feature to find the entrance of closed type parking stall. The cross-point type feature includes the location of the point and the direction along the parking stall marking.

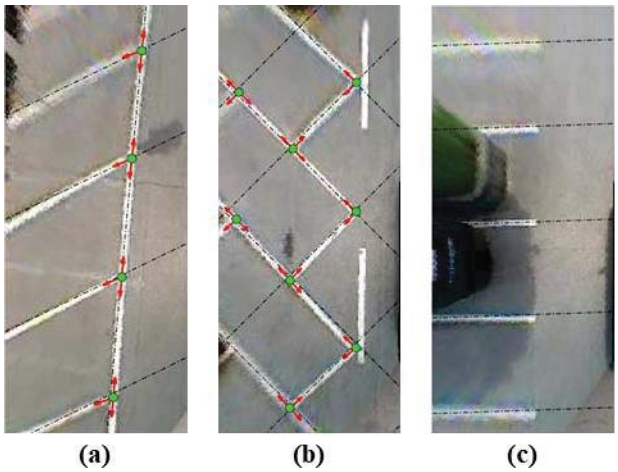

Figure 5 Cross-point type feature extraction results **(a)** Closed parallelogram shape **(b)** Closed slanted rectangular shape **(c)** Open rectangular shape

2.3.1 Location of the cross-point type feature

First, calculate the intersection of the line candidates and examine the direction of the line features around the intersection. Figure 4 (a) shows the line candidate extraction results from the AVM image including closed type parking stalls. The intersection P of the two straight lines l_1 and l_2, represented as $a_1x + b_1y + c_1 = 0$ and $a_2x + b_2y + c_2 = 0$ in figure 4 (a), is calculated by the following equation (1) using the homogenous coordinate.

(In homogenous coordinate)

$$l'_1 = [a_1 \quad b_1 \quad c_1]^T, l'_2 = [a_2 \quad b_2 \quad c_2]^T$$

$$P' = [x'_c \quad y'_c \quad w'_c]^T = l'_1 \times l'_2 = \begin{bmatrix} b_1c_2 - b_2c_1 \\ a_2c_1 - a_1c_2 \\ a_1b_2 - a_2b_1 \end{bmatrix} \quad (1)$$

(In image coordinate)

$$P = [x_c \quad y_c] = \left[\frac{x'_c}{w'_c} \quad \frac{y'_c}{w'_c}\right]^T = \left[\frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1} \quad \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1}\right]^T$$

As shown by a dotted circle in figure 4 (a), however, an error occurs between the calculated intersection and the intersection of the actual parking stall markings due to the AVM calibration error. Therefore, it is necessary to correct the calculated intersections before checking the line directions. To extract lines using the radon transform again, we first set the regions of interest around the intersections. Then extract lines in the regions of interest and correct to the intersection of extracted lines. For computational efficiency, our proposed method takes only the angles near the slopes of the two straight lines that generated the intersection. The intersection information in which two lines are not detected is deleted. Figure 4 (b) shows the intersection correction result.

2.3.2 Direction of the cross-point type feature

As shown in figure 4 (b), since there are up to four directions to search from the intersection, the presence of

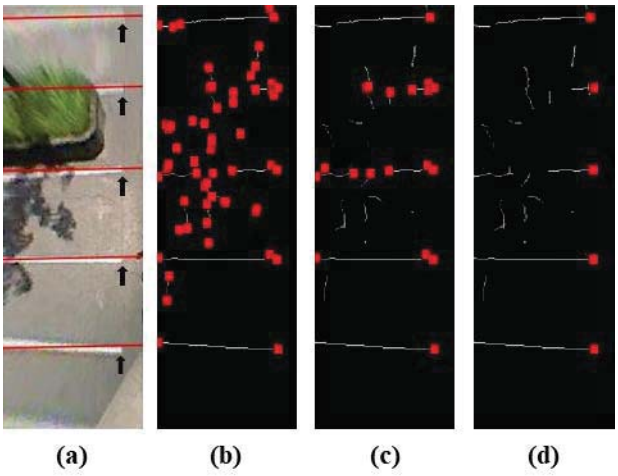

Figure 6 Location extraction of the end-point type feature **(a)** Line candidate extraction results **(b)** Locations of candidates for the end-point type features **(c)** Locations of candidates for the end-point type features along the line candidates **(d)** Final locations of the end-point type features

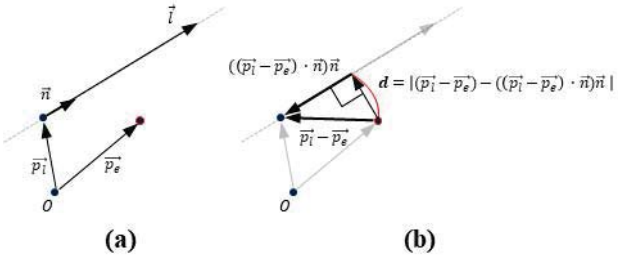

Figure 7 Calculation of distance between end-point type feature and line candidate **(a)** Line candidate and end-point type feature **(b)** Distance from a point to a line

line feature is confirmed and verified in four directions, and the angle information is also calculated. Figure 5 shows the location of the cross-point type feature (marked as a green circle) finally extracted from the AVM image regardless of types of parking stalls. The arrows in red indicate the angle information of the cross-point type features. In figure 5 (c), the cross-point type feature is not generated because the line candidates do not intersect.

2.4 End-point type feature Generation

This section defines an end-point type feature to find entrance of open type parking stall. The end-point type feature includes the location of the point and the direction along the parking stall marking.

2.4.1 Location of the end-point type feature

Figure 6 (a) shows the line candidate extraction results from the left side of AVM image including the open type parking stalls. As shown by arrows in black in figure 6 (a), the location of the end-point type feature can be obtained by detecting the point where the line feature ends along the line candidate in the line feature map.

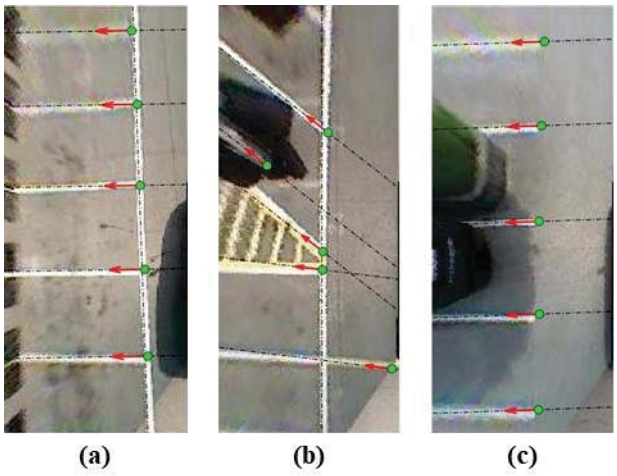

Figure 8 End-point type feature extraction results **(a)** Closed rectangular shape **(b)** Closed rectangular, and parallelogram shape **(c)** Open rectangular shape

Figure 6 shows the several steps of finding the location of the end-point type feature. First, the intensity value of the feature map is binarized, and the skeleton of the feature map is obtained using the morphological operation [9]. The end points of the skeleton become the locations of candidates for the end-point type feature. The locations of candidates for the end-point type feature obtained from the AVM image of figure 6 (a) are shown in figure 6 (b) marked with a red square over the feature map.

Assuming that ego vehicle is moving along the parking aisle, the entrance of parking stall can be found either at the left or right to the vehicle. The actual location of the end-point type feature can be found as an end point near the vehicle. Therefore, in the left/right side of AVM images, the end-point type features correspond to the right/left most pixels along the line candidates. The locations of candidates for the end-point type features and the final extracted locations of the end-point type features are shown in figure 6 (c) and (d), respectively.

The distance between the line candidate and the location of a candidate for the end-point type feature can be easily calculated as shown in figure 7. As shown in figure 7 (a), if a point on the line candidate is p_l and the unit vector in the line candidate direction is $\vec{n}$, the line candidate vector $\vec{l}$ can be represented as $\vec{p_l} + t\vec{n}$. When t is a nonzero scalar value, the distance **d** between the line candidate vector $\vec{l}$ and the location of a candidate p_e can be calculated as shown in figure 7 (b).

2.4.2 Direction of the end-point type feature

Since this paper assumes that the vehicle runs along the parking aisle, the angle of the end-point type feature is obtained from the angle of line candidate not facing the vehicle. Figure 8 shows the location of the end-point type feature (marked as a green circle) finally extracted from the AVM image regardless of types of parking stalls. The arrows in red indicate the angle information of the end-point type features.

Figure 9 AVM camera configuration

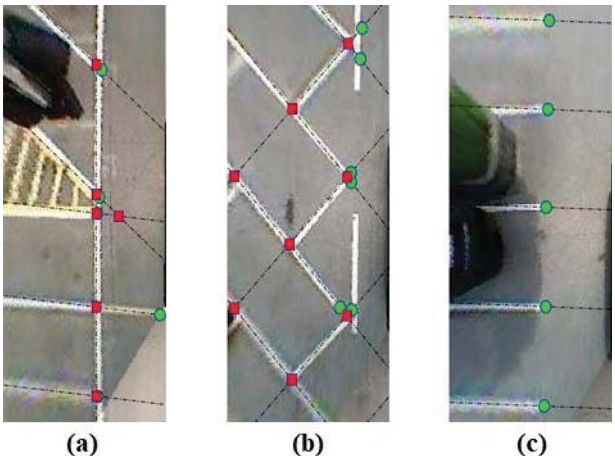

Figure 10 Both cross-point type feature and end-point type feature extraction results **(a)** Closed rectangular as well as parallelogram shape **(b)** Closed slanted rectangular shape **(c)** Open rectangular shape

3 Experimental Results

3.1 AVM Configuration

The video database on the parking lot used in these experiments was collected by an AVM system on a Hyundai Tucson. An AVM system consists of four fisheye cameras located at the front and rear sides and below the side-view mirrors as illustrated in figure 9. The AVM image resolution is 480 × 300, but in this paper, the parking stall detection was performed only in the left or right side of AVM images of 330 × 110 resolution. The actual displacement per pixel in the AVM image is 3.75cm/pixel.

3.2 Parking feature Extraction

Figure 10 shows the location of the cross-point type feature as a red rectangle and that of the end-point type feature as a green circle, extracted from AVM images including various types of parking stalls. However, as shown in figure 10 (a) and (b), the end-point type feature can also be detected at the entrance of closed type parking stall because it is extracted from the end points of all line candidates. Therefore, the end-point type features that are within a certain distance from the cross-point type features should be removed. After this process, the cross-point type and end-point type features are integrated into the parking features.

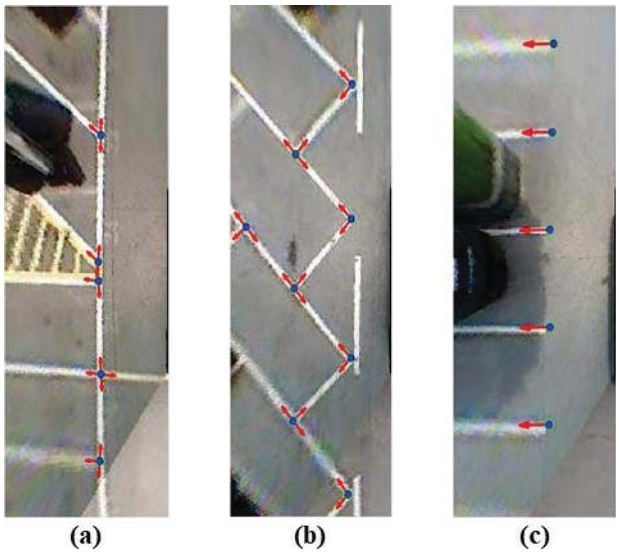

Figure 11 Parking feature extraction results **(a)** Closed rectangular, and parallelogram shape **(b)** Closed slanted rectangular shape **(c)** Open rectangular shape

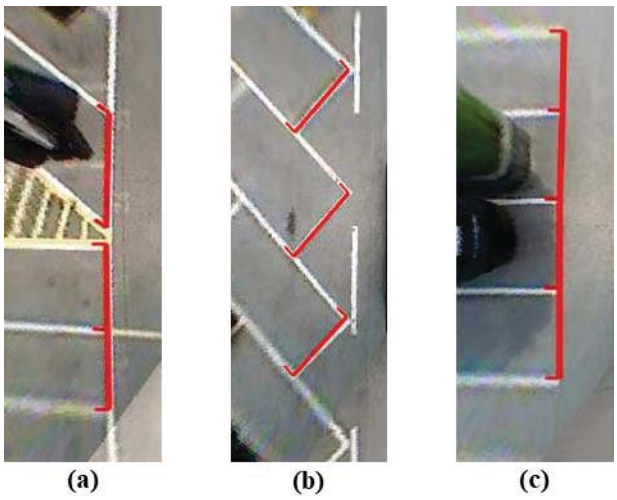

Figure 12 Parking stall detection results **(a)** Closed rectangular as well as parallelogram shape **(b)** Closed slanted rectangular shape **(c)** Open rectangular shape

Figure 11 shows the location of the parking features (marked as a blue circle) finally extracted from the AVM images including the various types of parking stalls, and the arrows in red indicate the angle information of the parking features. The end-point type features which exist at a Euclidean distance of fewer than 30 pixels from the cross-point type features have been removed. As shown in figure 11, the parking features were detected correctly at the parking stall entrance.

3.3 Parking stall Detection

To find an entrance of parking stall, our proposed method combines the parking features as follows. If two features are separated by a width of W, which is the pixel distance of the stall width shown in AVM, and at the same time

Table I Parking stall detection performance of the proposed method

Shape	GT	TP	FP	Recall	Precision
CR	1161	1130	0	97.33%	100%
CS	469	441	17	94.03%	96.29%
CP	686	671	0	97.81%	100%
OR	1793	1718	92	95.82%	94.92%
Total	**4109**	**3960**	**109**	**96.37%**	**97.32%**

share the angle information, not an angle between the two features, then they are recognized as an entrance of parking stall. Figure 12 shows the parking stall detection results. Since the actual stall width is approximately 230cm on average, W was set to 60 in the experiment.

3.4 Performance Evaluation

The AVM images used in the experiment include Closed Rectangular shape(CR), Closed Slanted rectangular shape(CR), Closed Parallelogram shape(CP), and Open Rectangular shape(OP) parking stalls. Recall and precision used as measures of performance evaluation are calculated by the following equation (2).

$$\text{Recall} = \frac{\#\ of\ correctly\ detected\ parking\ stalls}{\#\ of\ existing\ parking\ stalls}$$
$$\text{Precision} = \frac{\#\ of\ correctly\ detected\ parking\ stalls}{\#\ of\ detected\ parking\ stalls} \quad (2)$$

Table I shows the performance evaluation results of the proposed parking stall detection method. In the table, GT, TP, and FP indicate the number of existing parking stalls, the number of correctly detected parking stalls, and the number of falsely detected parking stalls in the AVM images, respectively. As shown in the Table I, the proposed method correctly detected 3,960 parking stalls among 4,109 various types of parking stalls. On average, its recall and precision are 96.37% and 97.32%, respectively.

3.5 Analysis of parking stall detection results

Figure 13 shows the final detection results of the parking stalls in various environments on the whole AVM image. As shown in figure 13, the proposed method detected the parking stalls robustly despite parked vehicles or shadows. However, since it is not easy to determine reliably whether the detected parking stall is vacant using the AVM image alone as shown in figure 13, it is necessary to study the empty parking stall classifier later for automatic parking system.

Figure 14 show the parking stall detection results while the vehicle is moving. In figure 14, L1 to L4 and R1 to R3 refer to the parking stalls to the left side and to the right side, respectively. While the vehicle was moving from figure 14 (a) to (b), R1 parking stall was not detected because one of the parking features at the entrance of parking stall disappeared from the AVM image. In the case of the vehicle moving from figure 14 (b) to (c) and from (c) to (d), L1 and L2 parking stalls were not detected because the parking stall marking was occluded by other vehicles. To cope with the detection failure cases shown in figure 14, a parking stall tracking algorithm is developed.

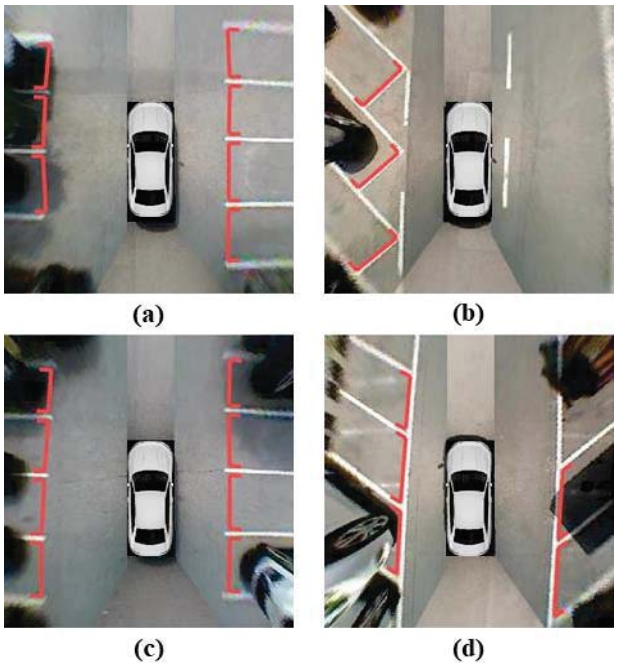

Figure 13 Parking stall detection results on the whole AVM images **(a)** Open rectangular shape **(b)** Closed slanted rectangular shape **(c)** Open rectangular shape **(d)** Closed parallelogram shape

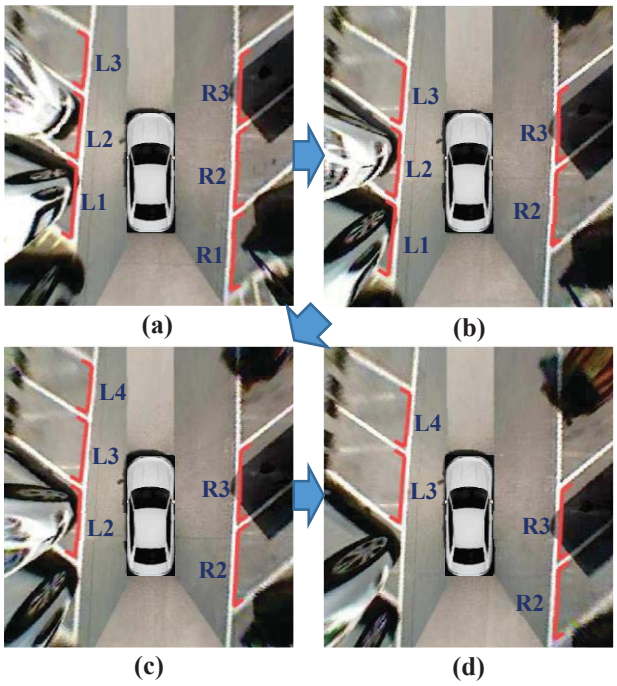

Figure 14 Parking stall detection results while the vehicle is moving **(a)** First sequence **(b)** Second sequence **(c)** Third sequence **(d)** Final sequence

4 Conclusions

This paper proposes a novel method to detect the parking stalls by analyzing the types of the parking stall entrance. For this purpose, cross-point type feature for the closed type parking stall and end-point type feature for the open type parking stall are obtained and analyzed. Experimental results show that the proposed method can detect various types of parking stalls up to 96.37% and 97.32% in terms of recall and precision, respectively. To improve the parking stall detection performance, research on the empty parking stall classifier and the parking stall tracking algorithm is in progress.

Acknowledgments

This work was supported by the World Class 300 R&D program of MSS. [S2367759, Development of driver assistance system using integrated heterogeneous camera system]

References

[1] C. Wang, H. Zhang, M. Yang, X. Wang. Automatic Parking Based on a Bird's Eye View Vision System. Advances in Mechanical Engineering, 2015,6(1):1-13.

[2] Jae-Seol Lee, Seok-Cheol Kee. Empty Parking Space Detection Method using Around View Monitoring System. Journal of Institute of Control, Robotics and Systems, 2017,23(6):455-461.

[3] Ho-Gi Jung, Dong-Suk Kim, Pal-Joo Yoon, Jahie Kim. Parking Slot Markings Recognition for Automatic Parking Assist System. In Proceedings of the 2006 IEEE Intelligent Vehicles Symposium, 2006:106-113.

[4] Jae-Kyu Suhr, Ho-Gi Jung. Automatic Parking Space Detection and Tracking for Underground and Indoor Environments. IEEE Transactions on Industrial Electronics, 2016, 63(9):5687-5698.

[5] ITE(Institute of Transportation Engineers), Brian Wolshon, Anurag Pande. Traffic Engineering Handbook (7th ed.). New Jersey: Jon Wiley & Sons, 2016, p. 437-499.

[6] M. Fan, Z. Hu, K. Hamada, and H. Chen. Line Filter-Based Parking Slot Detection for Intelligent Parking Assistance System. Proceedings of SAE-China Congress, 2014:175-181.

[7] Rafael C. Gonalez, Richard E. Woods. Digital Image Processing (3rd ed.). New Jersey: Pearson Prentice Hall, 2008.

[8] John Canny. A Computational Approach to Edge Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 1986,8(6):679–698.

[9] T.Yung Kong and Azriel Rosenfeld. Topological Algorithms for Digital Image Processing. Amsterdam: North-Holland, 1996, p. 101-106.