

Article

A Universal Vacant Parking Slot Recognition System Using Sensors Mounted on Off-the-Shelf Vehicles

Jae Kyu Suhr ¹  and Ho Gi Jung ^{2,*} 

¹ School of Intelligent Mechatronics Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea; jksuhr@sejong.ac.kr

² Department of Electronic Engineering, Korea National University of Transportation, 50 Daehak-ro, Chungju-si, Chungbuk 27469, Korea

* Correspondence: hogijung@ut.ac.kr; Tel.: +82-41-841-5366

Received: 5 March 2018; Accepted: 13 April 2018; Published: 16 April 2018



Abstract: An automatic parking system is an essential part of autonomous driving, and it starts by recognizing vacant parking spaces. This paper proposes a method that can recognize various types of parking slot markings in a variety of lighting conditions including daytime, nighttime, and underground. The proposed method can readily be commercialized since it uses only those sensors already mounted on off-the-shelf vehicles: an around-view monitor (AVM) system, ultrasonic sensors, and in-vehicle motion sensors. This method first detects separating lines by extracting parallel line pairs from AVM images. Parking slot candidates are generated by pairing separating lines based on the geometric constraints of the parking slot. These candidates are confirmed by recognizing their entrance positions using line and corner features and classifying their occupancies using ultrasonic sensors. For more reliable recognition, this method uses the separating lines and parking slots not only found in the current image but also found in previous images by tracking their positions using the in-vehicle motion-sensor-based vehicle odometry. The proposed method was quantitatively evaluated using a dataset obtained during the day, night, and underground, and it outperformed previous methods by showing a 95.24% recall and a 97.64% precision.

Keywords: automatic parking system; parking slot recognition; around-view monitor (AVM) system; sensor fusion

1. Introduction

As the demand for autonomous driving has rapidly increased, automatic parking systems have also been actively researched [1,2]. Automatic parking systems start by recognizing vacant parking spaces, and the methods for this task can be categorized into four approaches: user-interface-based, free-space-based, slot-marking-based, and infrastructure-based [3]. Among them, the free-space-based and slot-marking-based approaches have been more extensively developed in the automobile industry because they can be fully automated and implemented only with vehicle-mounted sensors. The free-space-based approach finds vacant parking spaces by recognizing adjacent vehicles. This approach works well in situations where the positions of the adjacent vehicles are favorable. However, it cannot find free spaces when there is no adjacent vehicle and its accuracy depends on the positions of adjacent vehicles. The slot-marking-based approach can mitigate this drawback. Since this approach finds parking spaces by recognizing markings on road surfaces, it can accurately localize parking spaces regardless of the existence and positions of adjacent vehicles in the presence of parking slot markings.

In order for a slot-marking-based method to be useful and practical, it needs to satisfy the following conditions: it is able to be fully automated, handle various types of parking slot markings,

and operate in a variety of lighting conditions. Since almost all slot-marking-based methods use cameras, it is particularly important that reliable performances can be achieved at night and underground, where severe lighting conditions are present. Figure 1 shows three parking lot images taken by an around-view monitoring (AVM) system in daytime, nighttime, and underground settings. In this figure, it can be seen that lighting conditions are quite unfavorable at night and underground due to dim lighting, amplified noise, and reflections on the road surface. These images are from the test dataset. The ON/OFF indication on the vehicle bonnet indicates whether the headlights are operating.

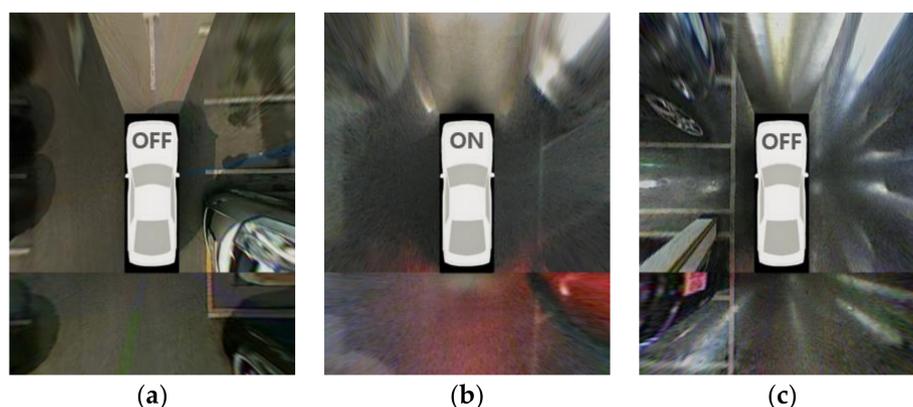


Figure 1. Parking lot images taken by an around-view monitoring (AVM) system. (a) Daytime; (b) Nighttime; (c) Underground.

Regarding the abovementioned conditions of the slot-marking-based approach, the previous methods can briefly be summarized as follows: The first method to recognize various types of parking slot markings was proposed in [3]. This method could not be fully automated because a driver has to manually designate the position of the parking slot's entrance. The method in [4] overcame this limitation by automatically detecting various types of parking slot markings using a hierarchical tree structure. However, since this method is sensitive to lighting conditions, it can only be applied under daytime conditions. To complement this method, the method in [5] was suggested. This method can reliably operate in underground parking lots where lighting conditions are unfavorable, but it can only handle a single type of parking slot marking. A detailed literature review will be presented in Section 2.

Therefore, in order to overcome the limitations of these previous methods, this paper proposes a method that can operate in a fully automatic manner, recognize various types of parking slot markings, and achieve robustness against a variety of lighting conditions. The proposed method works as follows: This method first detects the separating lines of parking slot markings by extracting parallel line pairs from AVM images. The separating lines detected in the current image are then combined with those detected in the previous images. Parking slot candidates are generated by pairing the combined separating lines based on the geometric constraints of the parking slot. These candidates are confirmed by recognizing their entrance positions using both line and corner features and classifying their occupancies using ultrasonic sensors. Finally, the vacant parking slots recognized in the current image are combined with those recognized in the previous images. The current positions of the separating lines and parking slots found in the previous images are tracked based on the in-vehicle motion sensor-based vehicle odometry. The proposed method was quantitatively evaluated with the dataset obtained in various conditions including daytime, nighttime, and underground. This method outperformed the previous methods by showing a 95.24% recall and a 97.64% precision.

The contributions of this paper are as follows:

- (1) To recognize various types of parking slot markings with open and close entrances, this paper proposes an approach that generates parking slot candidates based on separating lines and

- confirms their entrances using both line and corner features. Previous parallel line-based methods cannot handle parking slot markings with open entrances because they only detect line features.
- (2) To achieve robustness in severe lighting conditions at night and underground, this paper proposes an approach that tracks separating lines and parking slots using in-vehicle motion sensor-based odometry, whose accuracy is independent of lighting conditions. Previous slot-marking-based methods have limitations in handling severe lighting conditions because they only use features extracted from the current image.
 - (3) This paper efficiently combines the two aforementioned approaches to develop a universal parking slot recognition method that can handle various types of parking slot markings during the day, at night, and underground. The proposed method was quantitatively evaluated using the dataset obtained under various severe lighting conditions and showed outstanding performance. According to our literature review, there has been no previous method that can handle various types of parking slot markings and a variety of severe lighting conditions, and no previous method has been quantitatively evaluated in these different settings.

The rest of this paper is organized as follows: Section 2 explains the related research. Sections 3 and 4 introduce the sensor configuration and overview of the proposed method, respectively. Section 5 describes the separating line detection and tracking. Sections 6 and 7 explain the parking slot generation, tracking, and combining, respectively. Section 8 presents the experimental results. The paper is concluded in Section 9.

2. Related Research

The parking space detection methods can be categorized into four approaches: user-interface-based, free-space-based, slot-marking-based, and infrastructure-based. The literature review of this paper focuses on the first three approaches that can be implemented with only the vehicle-mounted sensors.

2.1. User-Interface-Based Approach

The user-interface-based approach finds vacant parking spaces based on the driver's manual input. Even though the methods in this approach cannot be automated, they are useful as backup tools for automatic methods. The method in [6] interacts with a driver using an augmented display. It helps parallel parking by allowing a driver to locate a vertical pole at the end of the parked vehicle. The method in [7] utilizes an arrow button-based interface. A driver can locate a rectangle at the desired parking space by clicking several arrow buttons. This method has been applied to the mass-produced Toyota Prius [8]. The method in [9] uses a drag-and-drop interface. A desired parking space can be manually located by dragging the inside and outside of a rectangle displayed in a touch screen.

2.2. Free-Space-Based Approach

The free-space-based approach recognizes vacant parking spaces by finding adjacent parked vehicles. This is the most popular approach since various range sensors can be used to find adjacent parked vehicles. However, this approach has a fundamental drawback in that it cannot properly work when there is no adjacent parked vehicle, so its performance is highly dependent on the positions of the adjacent parked vehicles. Among various range sensors, ultrasonic sensors are most popularly used because they are relatively inexpensive and easy to mount on mass-produced vehicles. The ultrasonic-sensor-based method has been adopted by most car manufacturers [10–13]. It recognizes vacant parking spaces based on the range data acquired by registering the outputs of the ultrasonic sensors. Cameras have also been popularly used in this approach. Most of the camera-based methods recognize vacant parking spaces by reconstructing surrounding 3D structures. The 3D reconstruction has been conducted by a variety of techniques including monocular motion stereo [14–16], binocular stereo [17,18], and structured light [19]. Contrary to these methods, the method in [20] recognizes

vacant parking spaces based on the horizontal edges of the adjacent vehicles without 3D reconstruction. The camera-based methods can readily use the cameras already mounted on mass-produced vehicles, but they are sensitive to lighting conditions and usually require a large amount of computation for 3D reconstruction. The method in [21] fuses a camera and ultrasonic sensors in a low-level. It recognizes vacant parking spaces by combining the range data obtained from monocular motion stereo and ultrasonic sensors. Laser scanners have also been used for this approach and produced excellent performances because their range data are highly accurate [22,23]. However, laser scanners have drawbacks in that they are costly and have a durability problem due to rotation mirrors. Besides these sensors, photonic mixer devices [24] and short-range radars [25–27] have also been used in this approach.

2.3. Slot-Marking-Based Approach

The slot-marking-based approach recognizes vacant parking spaces by analyzing markings on road surfaces. Contrary to the free-space-based approach, its performance is not dependent on the existence and positions of adjacent parked vehicles. However, in order for this approach to properly work, parking slot markings should be presented. All methods in this approach use cameras that can easily capture markings on the ground. These methods can be categorized into semi-automatic and fully automatic. The methods in [3,28,29] detect parking slot markings in a semi-automatic manner. The one-touch method in [28] recognizes parking slot markings based on one manually designated point, and an efficient version of this method was presented in [29]. Since this method can deal with only a single type of parking slot marking, the two-touch method in [3] was proposed. It can recognize various types of parking slot markings based on two manually designated points. Unlike these semi-automatic methods, the methods in [4,5,30–41] detect parking slot markings in a fully automatic manner. The method in [30] recognizes parking slots based on neural-network-based color segmentation. The method in [31] detects parallel line pairs using a specialized filter and Hough transform to recognize parking slot markings. The method in [32] uses a similar approach that detects parallel line pairs using Radon transform. The method in [33] detects straight lines using an improved random sample consensus (RANSAC) algorithm and recognizes parking slots based on the line detection result. The method in [34] recognizes parking slots using vertically oriented lines and their occupancies using a difference-of-Gaussians-based histogram. The method in [35] extracts parking slot boundaries by applying a sequential RANSAC line estimator to binarized ridge images. The method in [36] detects parallel line pairs and lines perpendicular to them using a probabilistic Hough transform and recognizes parking slots by combining those lines. The method in [5] reliably recognizes parking slots in indoor and underground settings by detecting a guideline using RANSAC and separating lines using distance transform. The method in [37] detects junctions of parking slots based on adaptive boosting with integral channel features and recognizes parking slots by combining the detected junctions. While all of the above methods have a limitation in that they can only deal with one or two types of parking slot markings, several methods have been proposed to handle more types of parking slot markings. The method in [38] recognizes various types of parking slot markings using a hierarchical tree structure in rearview camera images. This method has been applied to AVM images in [39] and combined with an ultrasonic-sensor-based occupancy classification in [4]. The method in [40] clusters a con-hat filtering result and detects lines using RANSAC to recognize parking slots in rearview camera images, and this method has been applied to AVM images in [41]. It works well in the case of parking slots with closed entrances but cannot recognize those with open entrances. In terms of lighting conditions, there has been no previous method that quantitatively shows its ability to handle various types of parking slot markings under unfavorable lighting conditions such as nighttime and underground.

The proposed method can be categorized as a slot-marking-based approach. Compared with the other previous methods in this approach, this method can operate in a fully automatic manner without

any driver intervention, recognize various types of parking slot markings whether its entrance is close or open, and reliably work even in nighttime and underground settings.

3. Sensor Configuration

The proposed method uses only those sensors already mounted on an off-the-shelf vehicle, the Hyundai Azera [42]. They include an AVM system, ultrasonic sensors, and in-vehicle motion sensors. The AVM system is composed of four fisheye cameras located at the centers of the front and rear bumpers and under the side-view mirrors as shown in Figure 2 with red triangles. Four images taken from four different cameras are transformed into bird's-eye-view images and are stitched to each other in order to produce an AVM image. The resolution of each fisheye camera is 640×480 pixels and that of the AVM image is 360×480 pixels. Two ultrasonic sensors are located at both sides of the front bumper as shown in Figure 2 with green diamonds. Their operating range and resolution are 30~450 cm and 2 cm, respectively. In-vehicle motion sensors including wheel speed and yaw rate sensors are located inside the vehicle, and are used for vehicle odometry estimation. The vehicle odometry is calculated by the parking assistant system embedded in the Hyundai Azera, and the proposed method uses odometry calculation results obtained through the in-vehicle network. It contains the orientation angle and displacements in longitudinal and lateral directions with respect to the center of the two rear wheels. The resolutions of the orientation angle and displacement are 0.1° and 0.1 cm, respectively. All those data are synchronized to have a 15 Hz acquisition frequency.

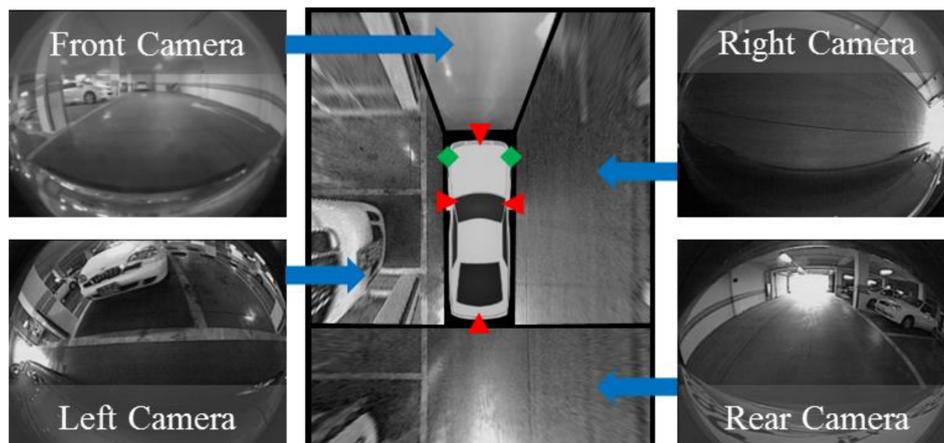


Figure 2. Sensor configuration. Red triangles and green diamonds are four cameras of the AVM system and two ultrasonic sensors, respectively.

4. Overview of the Proposed Method

The proposed method utilizes a line-detection-based approach for recognizing parking slots because this approach has been proven to be more robust against lighting conditions compared with the corner-based approach [5]. This method handles four types of parking slot markings: rectangular, slanted rectangular, diamond, and open rectangular as shown in Figure 3. In this figure, a line marking composed of a pair of red and blue lines is called a separating line because it separates adjacent parking slots. Since the parking slots of various types consist of separating line pairs, the proposed method recognizes parking slots using an approach that detects and combines separating lines. This method also tracks the detected separating lines through the image sequence using the vehicle odometry. This approach can increase the recognition performance because it is difficult to detect the separating lines in every image, especially in severe lighting conditions. In addition, this method uses both line and corner features when recognizing the parking slots' entrances. This approach makes it possible to handle parking slots with both closed and open entrances.

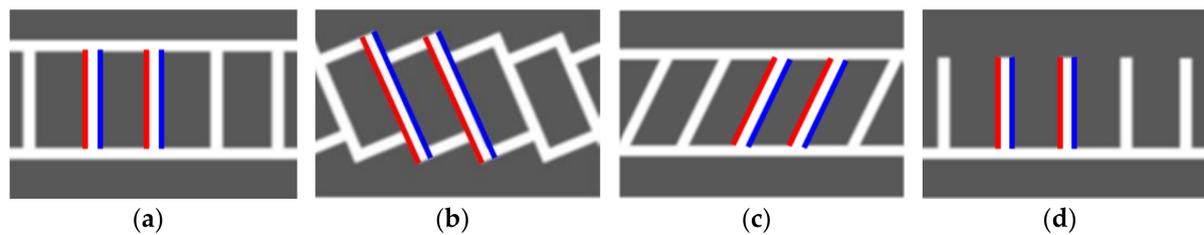


Figure 3. Four types of parking slot markings. (a) Rectangular type; (b) Slanted rectangular type; (c) Diamond type; (d) Open rectangular type.

Figure 4 shows the flowchart of the proposed method. This method first detects separating lines from an AVM image. The detected separating lines are then combined with the previously detected separating lines whose positions are tracked by the vehicle odometry. Once the separating lines are combined, parking slot candidates are generated by pairing the separating lines based on the geometric constraints of the parking slot. The parking slot candidates are then confirmed by the ultrasonic-sensor-based occupancy classification followed by the line- and corner-feature-based entrance determination. Finally, the recognized vacant parking slots are combined with the previously recognized vacant parking slots whose positions are tracked by the vehicle odometry in order to achieve a more reliable recognition result.

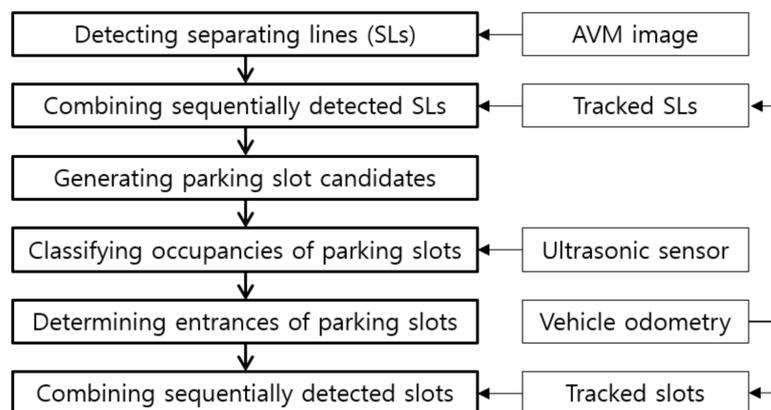


Figure 4. Flowchart of the proposed method.

5. Separating Line Detection and Tracking

5.1. Separating Line Detection

The separating line has two important properties: one is that it consists of two parallel lines and the other is that gradient directions of two parallel lines are opposite to each other. Red and blue line pairs in Figure 3 indicate the line pairs constituting the separating lines. It can be noticed that the red and blue lines are parallel and their gradient directions are opposite. The proposed method detects the separating lines based on these two properties. One of the simplest ways to find parallel line pairs is to detect multiple lines and then pair them together. However, this approach has limitations in that it needs a sophisticated line pairing condition and reduces robustness by separately handling two correlated lines. Thus, the proposed method uses an approach that simultaneously finds a parallel line pair whose gradient orientations are opposite. The methods in [31] and [32] also find parallel lines using Hough and Radon transforms, respectively. However, they ignore one of the important properties that gradient detections of two lines are opposite and suffer from the problem of parameter resolution, a fundamental limitation of the voting-based approach. Therefore, the proposed method utilizes RANSAC in order to accurately and robustly detect parallel line pairs.

The Sobel operator [43] is first applied to the AVM image for extracting edge pixels and their gradient orientations, and RANSAC is then applied to those edge pixels. To properly apply RANSAC, a least squares estimator should be formulated in advance. A pair of parallel lines can be described as

$$\begin{aligned} \alpha u + \beta &= v \\ \alpha u' + \gamma &= v' \end{aligned} \quad (1)$$

where α , β , and γ are parameters of a pair of parallel lines, and (u, v) and (u', v') are locations of the edge pixels whose gradient orientations are opposite. If two parallel lines are composed of N and M edge pixels, Equation (1) can be rephrased as

$$\underbrace{\begin{bmatrix} u_1 & 1 & 0 \\ \vdots & \vdots & \vdots \\ u_N & 1 & 0 \\ u'_1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ u'_M & 0 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}}_x = \underbrace{\begin{bmatrix} v_1 \\ \vdots \\ v_N \\ v'_1 \\ \vdots \\ v'_M \end{bmatrix}}_b. \quad (2)$$

x , which minimizes the least squares error, can be estimated by the pseudo-inverse as

$$x = (A^T A)^{-1} A^T b. \quad (3)$$

The proposed method applies RANSAC to the above least squares estimator as follows:

1. Randomly select one edge pixel.
2. Randomly select two edge pixels whose orientations are opposite to that of the previously selected one.
3. Estimate parallel line pair parameters using the selected three edge pixels via the least squares estimator.
4. Count consensus sets (edge pixels) based on perpendicular distance. Orientations of the consensus sets should coincide with those of the randomly selected edge pixels.
5. Repeat the above procedures and select the parameters that give the maximum number of consensus set.
6. Re-estimate parallel line pair parameters using all edge pixels in the selected consensus set based on the least squares estimator.

RANSAC needs a threshold value that determines whether an edge pixel is an inlier or not. This value is experimentally set to 2 pixels. It means that the separating line detection method allows edge pixels deviating by ± 2 pixels from the ideal line. To detect multiple parallel line pairs, the proposed method repeats the above procedure while removing the edge pixels that support the previously estimated line pairs. This method discards the separating line if its width is dissimilar to the prior knowledge of the marking line width (15~25 cm) or it is not supported by enough edge pixels. In most cases, the separating lines are not parallel to the travel direction of the vehicle when the vehicle passes beside the parking slots. Thus, the separating lines whose directions are similar to the travel direction of the vehicle are discarded. Figure 5a–c show the separating lines detected by the proposed method in daytime, nighttime, and underground settings, respectively. In this figure, the area of the AVM image with no parking slot is omitted for conciseness. Note that, in Figure 5, the separating line is drawn from the image boundary to the position near the side of the vehicle because the separating line detection method does not explicitly recognize the start and end positions. It can be found that the proposed method can reliably find multiple separating lines regardless of

lighting conditions. Since the separating lines whose directions are similar to the travel direction of the vehicle are removed, most of the entrance lines are not detected during the separating line detection procedure. Missed separating lines will be substituted while sequential detection results are combined, and falsely detected separating lines will be removed while parking slot candidates are generated. This will be explained later in detail. This paper assumes that the AVM system is properly calibrated at its manufacturing stage. If the AVM system is incorrectly calibrated, the performance of the parallel line detection may be degraded.

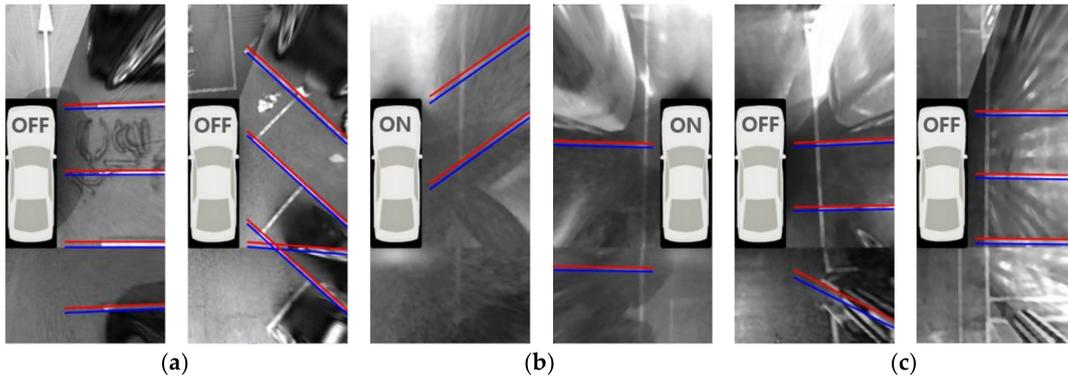


Figure 5. Separating line detection results. (a) Daytime; (b) Nighttime; (c) Underground.

5.2. Separating Line Tracking

Under severe lighting conditions, the separating line detection performance is normally degraded. For instance, at night, a separating line near the camera is relatively well detected, but a separating line far from the camera is hardly detected because the image quality of that area is quite inferior. Figure 6a–c show AVM images taken while the vehicle is moving next to the parking slots at night. In these figures, A, B, and C are indexes of three separating lines. In Figure 6a, A is detected because it is located near the camera, but B is not detected due to the inferior image quality. In Figure 6b, as the vehicle moves, B becomes close to the camera and is detected, but A becomes further from the camera and is not detected. In Figure 6c, as the vehicle moves again, C is detected but A and B are not detected. In such a situation, parking slots cannot be reliably recognized if only the separating lines detected in the current image are used. Thus, the proposed method utilizes an approach that tracks positions of the separating lines detected in previous images and recognizes parking slots using the separating lines detected in both the current and previous images. The separating lines are tracked based on the vehicle odometry calculated by in-vehicle motion sensors: wheel speed and yaw rate sensors. These sensors have been already mounted on mass-produced vehicles. When the orientation angle and displacement vector of the vehicle are θ and $[t_x \ t_y]^T$, respectively, and the parameters of the previously detected line is $\mathbf{l}_p = [\alpha \ -1 \ \beta]^T$, where α and β are the line parameters shown in Equation (1), the tracked parameters of the corresponding line in the current image, \mathbf{l}_c can be calculated as

$$\mathbf{l}_c = H^{-T} \mathbf{l}_p, \quad \text{where } H = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

If the currently detected separating line overlaps the previously detected and tracked separating line, the one supported by a larger number of edge pixels is selected. When comparing those numbers, the number for the tracked separating line is weighted by r (<1.0) because its position includes a cumulative error induced by the vehicle odometry. r is experimentally set to 0.7. Whether or not the two lines overlap is determined based on the similarity of the slope and y-intercept. Figure 6d shows the separating lines obtained by combining those detected in both the current and previous images.

In this figure, C is detected in the current image, while A and B are detected in the previous images and tracked based on the vehicle odometry. If this approach is applied, the parking slot recognition performance is expected to increase because it is possible to use the separating lines located at both the superior and inferior quality regions of the current image.

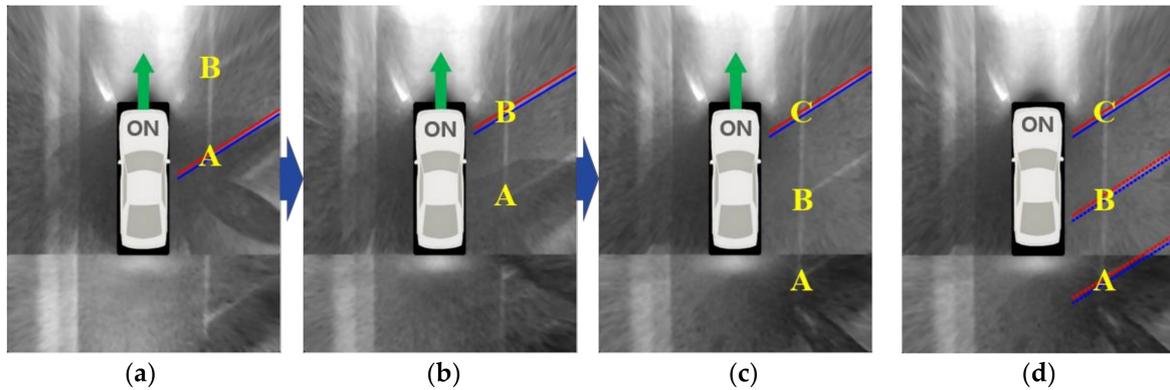


Figure 6. Separating line detection and tracking at night. (a–c) The consecutive detection results using only the current image; (d) The detection result obtained by combining the consecutive detection results via the vehicle-odometry-based tracking.

6. Parking Slot Generation

6.1. Parking Slot Candidate Generation

Parking slot candidates are generated by pairing the separating lines based on the geometric constraints of the parking slot. The proposed method uses two simple geometric constraints: two separating lines are parallel, and the width between two separating lines is in a certain range (190~350 cm). Although the separating line tracking can increase the detection rate, it might increase the number of false detections. Thus, in order to prevent false parking slot candidates, the proposed method uses an additional constraint that at least one separating line should be detected in the current image. Figure 7a,b depict the parking slot candidate generation procedure. In these figures, the first row shows the rectangular type parking slots that have closed entrances, and the second row shows the open rectangular type parking slots that have open entrances. Figure 7a depicts the separating line detection results, and Figure 7b depicts the parking slot candidates generated by the separating lines in Figure 7a. It can be seen that the parking slot candidates are correctly generated, while some falsely detected separating lines that do not follow the aforementioned geometric constraints are discarded. Note that the entrances of the parking slots are not yet recognized, so they are arbitrarily drawn in this figure. The procedure for the entrance recognition will be explained in Section 6.3.

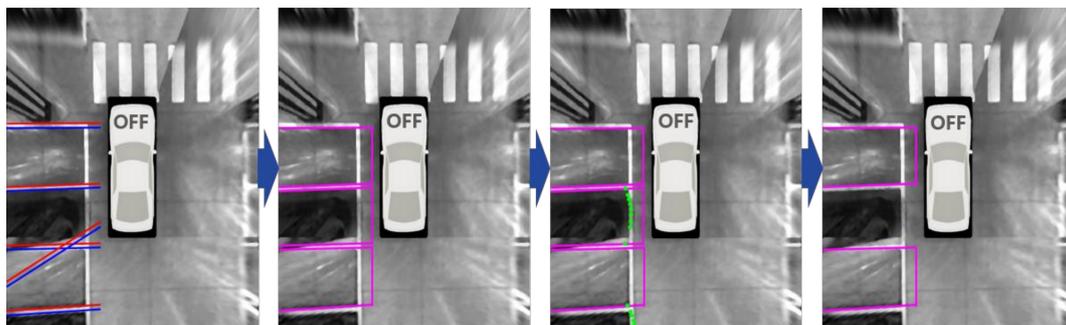


Figure 7. Cont.

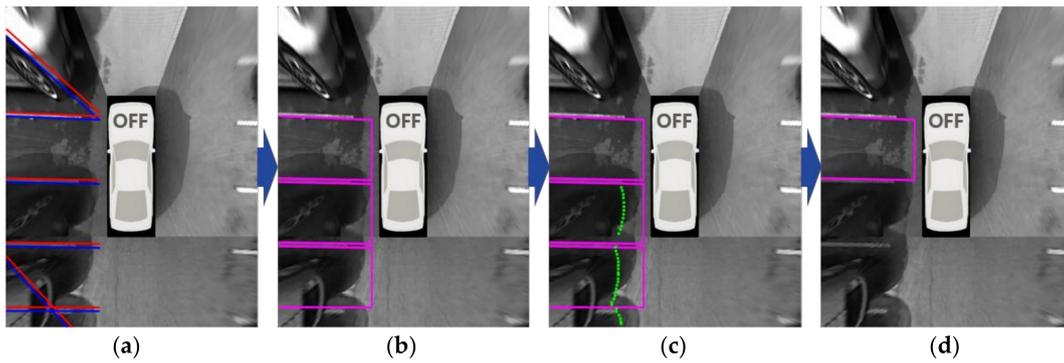


Figure 7. Parking slot candidate generation and occupancy classification. (a) Separating line detection results; (b) Parking slot candidate generation results; (c) Range data obtained by an ultrasonic sensor; (d) Parking slot candidates classified as vacant.

6.2. Parking Slot Occupancy Classification

After generating the parking slot candidates, occupancies of those candidates are classified. If a slot candidate is determined as occupied, this slot is deleted because the vehicle cannot park in it. The occupancy classification is conducted based on the range data acquired by ultrasonic sensors. Two ultrasonic sensors are located at both sides of the front bumper as shown in Figure 2 with green diamonds. Green dots in Figure 7c indicate the range data obtained by registering the outputs of the ultrasonic sensor while the vehicle is moving. This paper does not propose a parking slot occupancy classification method but adopts the method proposed in [4]. This method treats a parking slot as a single cell of the occupancy grid and calculates the posterior probability to be occupied. In this case, the occupancy of the parking slot can be modeled as a binary estimation problem with static state [44]. The posterior probability of parking slot occupancy can be sequentially calculated in log odds representation as

$$\begin{aligned}
 l_{1:t}(O) &= \log \frac{p(O|z_{1:t})}{1-p(O|z_{1:t})} = \log \left\{ \frac{p(O|z_t)}{1-p(O|z_t)} \cdot \frac{p(O|z_{1:t-1})}{1-p(O|z_{1:t-1})} \cdot \frac{1-p(O)}{p(O)} \right\} \\
 &= \log \frac{p(O|z_t)}{1-p(O|z_t)} + \log \frac{p(O|z_{1:t-1})}{1-p(O|z_{1:t-1})} - \log \frac{p(O)}{1-p(O)} \\
 &= l_t(O) + l_{1:t-1}(O) - l_o(O)
 \end{aligned} \tag{5}$$

where O means the parking slot is occupied, and V has the opposite meaning. z_t represents whether the ultrasonic sensor output at time index t is located inside the parking slot candidate, and it is either positive (P) or negative (N). $p(O|z_{1:t})$ is a posterior probability to be occupied given the ultrasonic sensor outputs from time index 1 to t . $p(O)$ is a prior probability to be occupied and is set to 0.5. To utilize Equation (5), $p(O|P)$ and $p(O|N)$, two cases of $p(O|z_t)$ were pre-calculated by Bayes' theorem using $p(P|O)$ and $p(P|V)$, which were estimated based on the training dataset. The training dataset includes 456 occupied and 357 vacant parking slots of various types with the corresponding ultrasonic sensor outputs. This training dataset is totally different from the test dataset. The log odds ratio in Equation (5) can be converted to the posterior probability as

$$p(O|z_{1:t}) = 1 - \frac{1}{1 + \exp\{l_{1:t}(O)\}}. \tag{6}$$

Occupancies of the parking slot candidates are classified based on the posterior probabilities, $p(O|z_{1:t})$, and those classified as occupied are removed. Figure 7d shows the parking slot candidates classified as vacant using the range data shown in Figure 7c.

6.3. Parking Slot Entrance Detection

Once the vacant parking slot candidates are selected, their entrances are found. To this end, the proposed method uses both line- and corner-based approaches. Figure 8 shows the shapes of the entrances for four types of parking slot markings. Since the entrances of the rectangular, slanted rectangular, and diamond types are composed of lines, the proposed method utilizes the line-based approach. However, the entrance of the open rectangular type cannot be found by the same approach because its entrance does not consist of lines. Thus, the corner-based approach is utilized in this case. The proposed method first applies the line-based approach to the vacant parking slot candidate. If this approach cannot find the entrance, the corner-based approach is then applied.

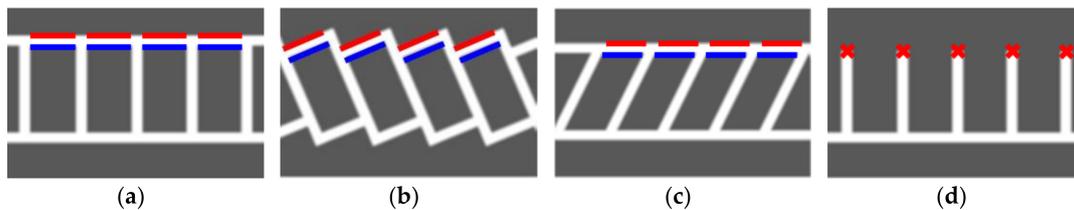


Figure 8. Entrances of four types of parking slot markings. (a) Rectangular type; (b) Slanted rectangular type; (c) Diamond type; (d) Open rectangular type.

The line-based entrance detection is almost the same as the parallel line pair detection used for finding the separating lines in Section 5.1. In this case, only the edge pixels located between two separating lines constituting the parking slot candidate is used. The edge pixels already found during the separating line detection are reused. Figure 9 depicts the line-based entrance detection procedure. Figure 9a shows a separating line pair constituting a parking slot candidate and Figure 9b shows the edge pixels located between two separating lines. A pair of red and blue lines in Figure 9c indicates the parallel line pair detection result produced by RANSAC, and Figure 9d presents the completely recognized parking slot after the line-based entrance detection.

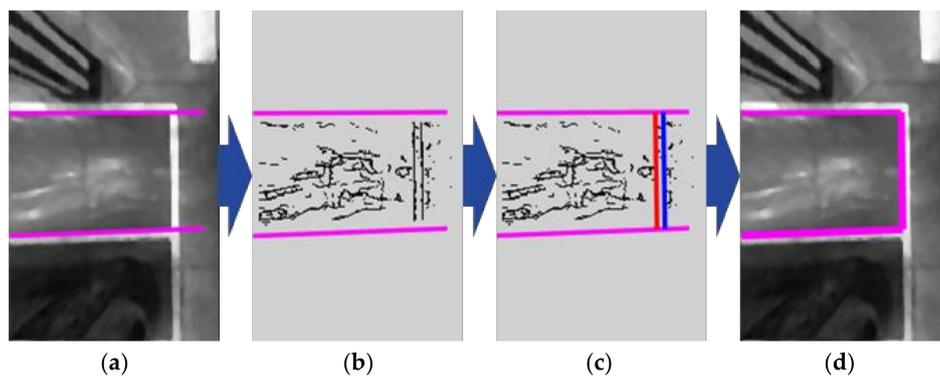


Figure 9. Line-based entrance detection procedure. (a) Separating line pair constituting a parking slot candidate; (b) Edge pixels between two separating lines; (c) RANSAC-based parallel line pair detection result; (d) Completely recognized parking slot.

If the number of edge pixels classified as inliers by the line-based entrance detection using RANSAC is smaller than a predetermined value (25% of the slot width), the parking slot candidate is considered as an open rectangular type, and the corner-based entrance detection is then applied. The corner-based entrance detection is based on cornerness values on two separating lines constituting a parking slot candidate. The cornerness value indicates how much an image location is similar to

the corner and is calculated based on the second moment matrix used for Harris corner detector [45]. The second moment matrix can be expressed as

$$M = \sum_{(u,v) \in W} \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} \quad (7)$$

where (u, v) indicates the image location, and W is a small spatial neighborhood around (u, v) . I_u and I_v are the image gradients in the horizontal and vertical directions, respectively. Since the matrix M contains information on the brightness variation in W , the image location (u, v) is considered as a corner if both eigenvalues (λ_1, λ_2) of M are large. Thus, the smaller eigenvalue (λ_{min}) of λ_1 and λ_2 is used for the cornerness value. λ_{min} can be efficiently calculated by the following approximation [45]:

$$\lambda_{min} \approx \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)} = \frac{\det(M)}{\text{trace}(M)}. \quad (8)$$

Figure 10 depicts the corner-based entrance detection procedure. Figure 10a shows a separating line pair constituting a parking slot candidate of open rectangular type, and Figure 10b shows the cornerness values on the separating line pair. It can be noticed that the entrance positions, which are the ends of the separating lines have high cornerness values. Once the cornerness values are calculated, a cornerness value profile is extracted from each separating line. Figure 10c shows two cornerness value profiles (C_1 and C_2) extracted from two separating lines. C_1 and C_2 are combined as

$$C = (C_1 + C_2) - |C_1 - C_2|. \quad (9)$$

The sum of two profiles is subtracted by the difference of the two profiles because it is desired that the location where two profiles both have large cornerness values is found. Discarding the location where the cornerness value is large on only one profile can reduce false entrance detection. If the maximum value of C is smaller than a predetermined value (50), the parking slot candidate is not confirmed as a parking slot and rejected. Figure 10d shows the combined cornerness value profile, C , calculated by C_1 and C_2 in Figure 10c. In this figure, the horizontal axis (index) refers to the coordinates moved by 1 pixel along the separating line. The location where C is maximized is detected as the entrance location. Figure 10e presents the completely recognized parking slot after the corner-based entrance detection. In the case of the open rectangular type, the entrance is depicted with a dotted line.

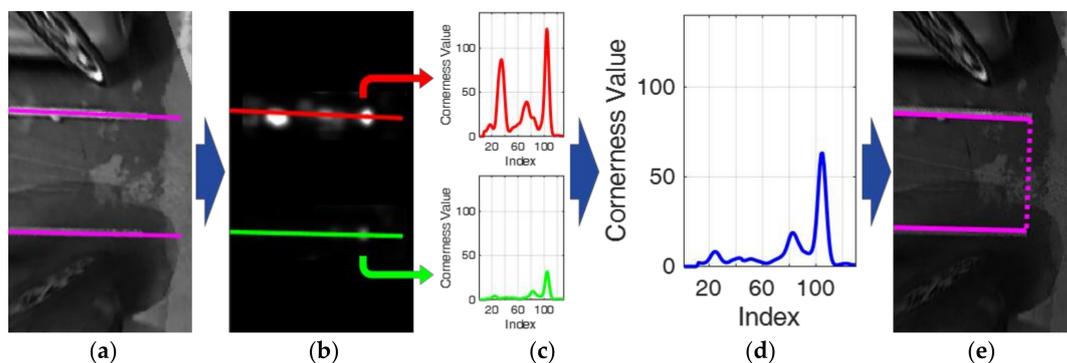


Figure 10. Corner-based entrance detection procedure. (a) Separating line pair constituting a parking slot candidate; (b) Cornerness values; (c) Two cornerness value profiles extracted from two separating lines; (d) Combined cornerness value profile; (e) Completely recognized parking slot.

7. Parking Slot Tracking and Combining

The proposed method uses vacant parking slots recognized not only from the current image, but also from the previous images. Positions of the previously recognized parking slots are tracked based on the vehicle odometry. Since a parking slot consists of two separating lines and two entrance points, the current position of the previously recognized parking slot can be tracked by the matrix H in Equation (4). The separating line can be tracked using Equation (4), and the entrance point can be tracked as

$$\mathbf{p}_c = H\mathbf{p}_p \quad (10)$$

where \mathbf{p}_p is the entrance point of the previous recognized parking slot and \mathbf{p}_c is its tracked position.

Once the vacant parking slots recognized from the current and previous images are obtained, this method finally combines them to obtain more reliable results. The parking slot combination procedure starts by checking overlapping conditions of the obtained parking slots using the intersection of union (IOU). IOU between rectangular areas of two parking slots (A_i, A_j) is calculated as

$$IOU(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \quad (11)$$

The overlapping conditions can be categorized in two cases: one is the overlap between parking slots recognized from the current image, and the other is the overlap between parking slots recognized from the current and previous images. In the first case, only one of the two parking slots should be selected since two different parking slots are unable to overlap in real situations. Two parking slots are considered to overlap if IOU is larger than a predetermined value T_c . In order to select a more appropriate parking slot, the proposed method uses two scores. The first score s_l is based on the reliability of two separating lines constituting the parking slot and is calculated using the number of inlier edge pixels that support two separating lines as

$$s_l = \min\left(1, \frac{N_1}{2L_1}\right) + \min\left(1, \frac{N_2}{2L_2}\right) \quad (12)$$

where N_1 and N_2 are the numbers of inlier edge pixels that support two separating lines, and L_1 and L_2 are the lengths of two separating lines in pixels. The reason for putting 2 in the denominator is that each separating line consists of two parallel lines. The minimum operator (\min) is used to limit the maximum value to 1. The second score s_e is calculated based on the reliability of the parking slot entrance. This score is calculated by the cornerness value in case of the open rectangular type and by the number of inlier edge pixels that supports the entrance in case of the other types as

$$s_e = \begin{cases} \min\left(1, \frac{K}{K_{max}}\right), & \text{if open rectangular} \\ \min\left(1, \frac{N_3}{2L_3}\right), & \text{otherwise} \end{cases} \quad (13)$$

where K is the cornerness value of the parking slot entrance, which is calculated by finding the maximum value from the combined cornerness value profile in Figure 10d, and K_{max} is the maximum value of K , which is experimentally predetermined. N_3 is the numbers of inlier edge pixels obtained by the line-based entrance detection procedure, and L_3 is the length of the parking slot entrance in pixels. The denominator includes 2 because the parking slot entrance consists of two parallel lines. If two parking slots recognized from the current image are determined to overlap based on IOU in Equation (11), the one that has a larger s_c in Equation (14) is selected.

$$s_c = s_l + s_e \quad (14)$$

The second overlapping situation is the overlap between parking slots recognized from the current and previous images. This overlapping situation can be further categorized into two cases: one is that two parking slots almost completely overlap, and the other is that two parking slots slightly overlap. The first case is considered that the same parking slot in the real world is consecutively recognized in an image sequence. It is determined that this case occurs when IOU is greater than T_s ($\gg T_c$). In this case, the number of repeated detections (D) for this parking slot is increased by 1, and the one that has the largest s_c in Equation (14) is selected. When comparing these scores, s_c of the previously recognized parking slot is weighted by r (<1.0) because its position includes a cumulative error induced by the vehicle odometry. r is experimentally set to 0.7. The second case is that two parking slots recognized from the current and previous images slightly overlap. It is determined that this case occurs when IOU is between T_c and T_s . Since two different parking slots are unable to overlap in real situations, the one that has a larger s_t in Equation (15) is selected.

$$s_t = s_l + s_e + s_d, \quad \text{where } s_d = \min\left(1, \frac{D}{D_{\max}}\right) \quad (15)$$

where s_d is a score calculated by the number of repeated detections (D), and D_{\max} indicates the maximum value of D , which is experimentally predetermined. As in the first case, when comparing these values, s_c of the previously recognized parking slot is weighted by r (<1.0). Figure 11 depicts the parking slot tracking and combining procedure at night. In this figure, A, B, C, and D indicate the indexes of the parking slots. Figure 11a–c are the parking slot recognition results of three images captured while the vehicle is moving. As aforementioned, this method uses the constraint that one of two separating lines must be detected in the current image to reduce false detections. Thus, if only a single image is used, it is difficult to recognize the parking slots existing in the inferior quality region of the current image. This is the reason that B in Figure 11a, A in Figure 11b, and A and B in Figure 11c are unrecognized. However, if those parking slots are tracked and combined using the method described in this section, it is possible to reliably recognize all parking slots as shown in Figure 11d.

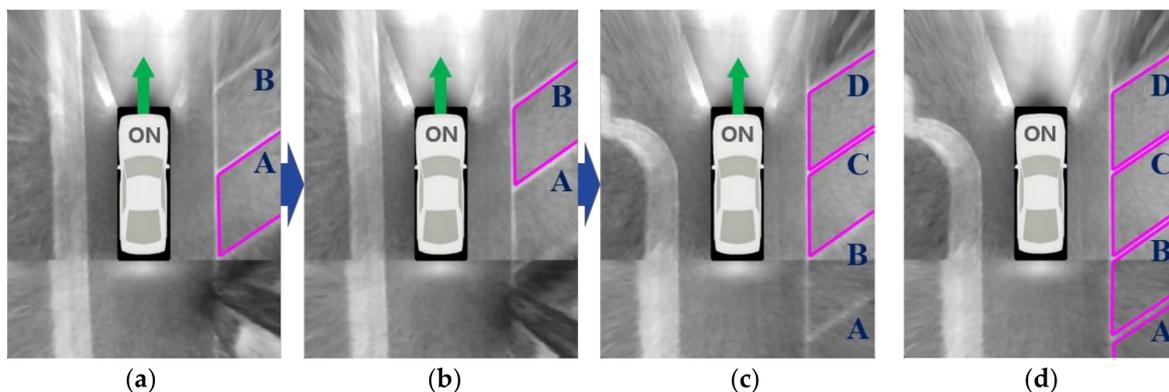


Figure 11. Parking slot tracking and combining procedure. (a–c) show the parking slot recognition results of three individual images; (d) shows the combined result of (a–c) using the suggested approach.

8. Experiments

The test dataset was acquired by an AVM system, ultrasonic sensors, and in-vehicle motion sensors on an off-the-shelf vehicle, the Hyundai Azera [42]. This dataset includes 609 vacant parking slots obtained in 221 different situations. Table 1 shows a detailed description of the test dataset. In terms of lighting condition, the dataset consists of 235, 214, and 160 vacant parking slots acquired during the day, at night, and underground, respectively. In terms of parking slot type, it consists of 287, 98, 120, and 104 vacant parking slots of rectangular, slanted rectangular, diamond, and open

rectangular types, respectively. The outdoor conditions include all four types of parking slots, but the underground conditions include only rectangular parking slots. This is because almost all parking slots in underground parking lots are of a rectangular type in Korea owing to its visibility and space efficiency. Figure 12a–c show AVM images acquired during the day, at night, and underground, respectively. These images were selected from the test dataset. It can be seen that shadows and stains can be seen during the day, and dim lighting and reflections can be seen at night and underground.

Table 1. Description of the test dataset.

Category	Type	Number of Vacant Parking Slots
Lighting condition	Daytime	235
	Nighttime	214
	Underground	160
Parking slot type	Rectangular	287
	Slanted rectangular	98
	Diamond	120
	Open rectangular	104
Total		609

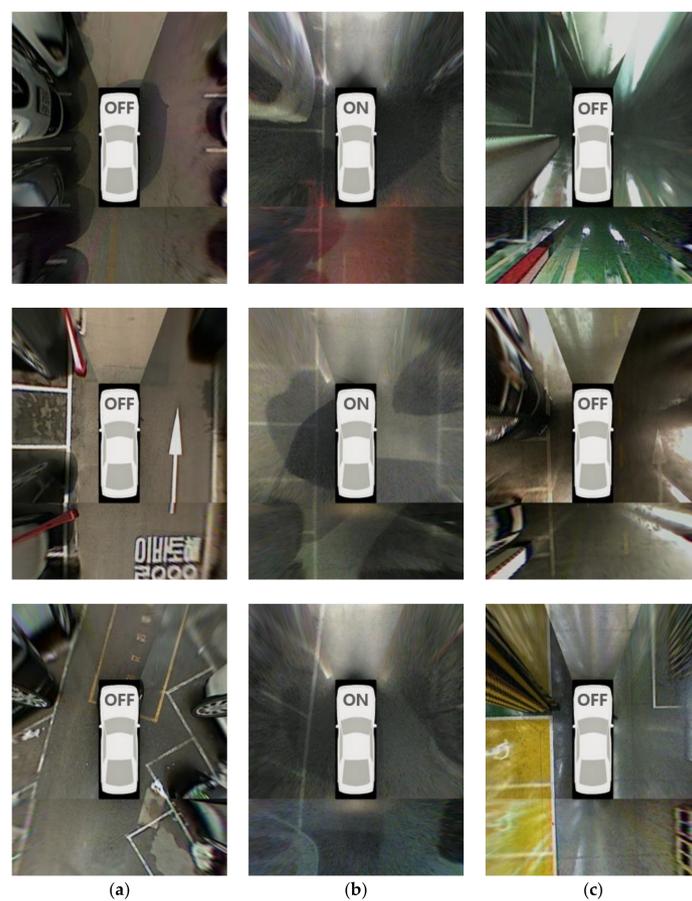


Figure 12. AVM images included in the test dataset. (a) Daytime; (b) Nighttime; (c) Underground.

Recall and precision in Equation (16) were used for evaluating the proposed method. These measures have also been used previously [4,5]

$$\begin{aligned} \text{recall} &= \frac{\text{number of correctly recognized slots}}{\text{number of existing slots}} \\ \text{precision} &= \frac{\text{number of correctly recognized slots}}{\text{number of recognized slots}} \end{aligned} \quad (16)$$

As in previous papers [4,5], a parking slot is considered to be correctly recognized if it is recognized before the rear of the vehicle passes by it and the recognition holds until it leaves an AVM image or the vehicle stops. A parking slot is considered to be falsely recognized if it is falsely recognized more than or equal to once through an AVM image sequence.

Table 2 shows the parking slot recognition performance of the proposed method. In this table, #slot, #TP, and #FP indicate the numbers of existing slots, correctly recognized slots (true positives), and falsely recognized slots (false positives), respectively. The proposed method correctly recognizes 580 parking slots out of 609 and produces 14 false positives. That is, its recall and precision are 95.24% and 97.64%, respectively. In terms of recall, this method gives the highest value (97.87%) during the day. This is because parking slot markings are clearly visible during the day because of the favorable lighting conditions. The recall of underground settings (92.50%) is lower than that of the nighttime situation (94.39%). In underground settings, both dim lighting and severe reflections on the ground hinder parking slot recognition as shown in Figure 12c, while only dim lighting hinders it in the nighttime situation. In terms of precision, the proposed method gives the highest value (99.51%) in the nighttime situation. Compared to the daytime and underground conditions, the nighttime situation includes lighting that is more severely dim, as shown in Figure 12b, and this lighting condition suppresses the edges not only from parking slot markings but also from other objects. Therefore, the proposed method rarely produces false positives in nighttime conditions. The small number of false positives means that the precision is high. In underground settings, a precision of 94.27% (the lowest precision of all settings) was achieved. This is because the edges from other objects and reflections on the ground are clearly visible, as shown in Figure 12c, so more false positives are produced in this type of setting. Figure 13 shows the final parking slot recognition results of the proposed method in various parking lot conditions. In this Figure 13a–c are the results of daytime, nighttime, and underground conditions, respectively. The entrance of the open rectangular type is depicted with a dotted line. The AVM image area with no parking slot is omitted for conciseness. The proposed method works successfully, despite the strong shadows and stains on the ground during the day (Figure 13a), the dim lighting and unclear parking slot markings at night (Figure 13b), and the strong reflections and myriad pillars underground (Figure 13c). Figure 14 shows the final parking slot recognition results of the proposed method in image sequences. These results show that the proposed method can reliably recognize various types of parking slots in a variety of lighting conditions. This is mainly because it generates parking slot candidates using separating lines, detects their entrances using line and corner features, and combines sequentially found separating lines and parking slots by tracking them using the vehicle odometry. The proposed method was implemented in C language, and its execution time was measured on a 3.40 GHz Intel Core i7-2600 CPU using a single core. It requires 54 ms/frame on average.

Table 2. Parking slot recognition performance of the proposed method.

Environment	#slot	#TP	#FP	Recall	Precision
Daytime	235	230	4	97.87%	98.29%
Nighttime	214	202	1	94.39%	99.51%
Underground	160	148	9	92.50%	94.27%
Total	609	580	14	95.24%	97.64%



Figure 13. Parking slot recognition results of the proposed method. (a) Daytime; (b) Nighttime; (c) Underground.

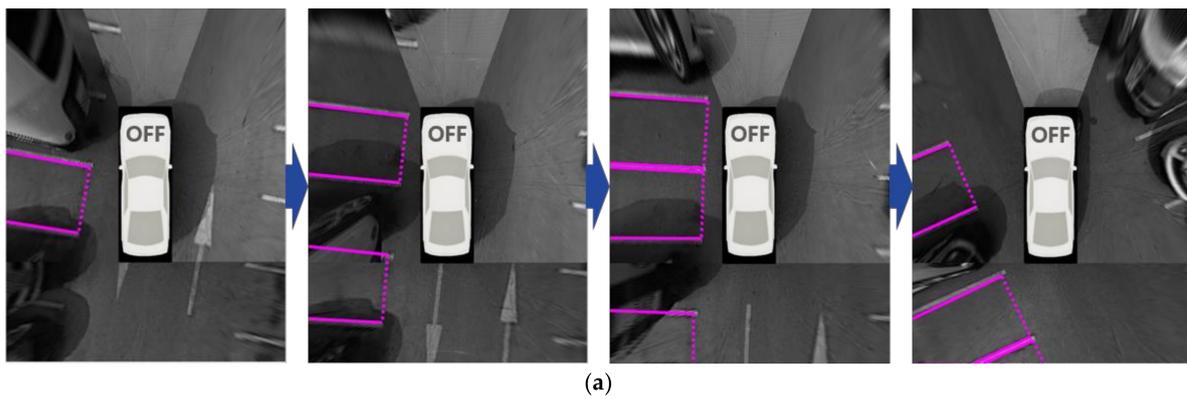


Figure 14. *Cont.*

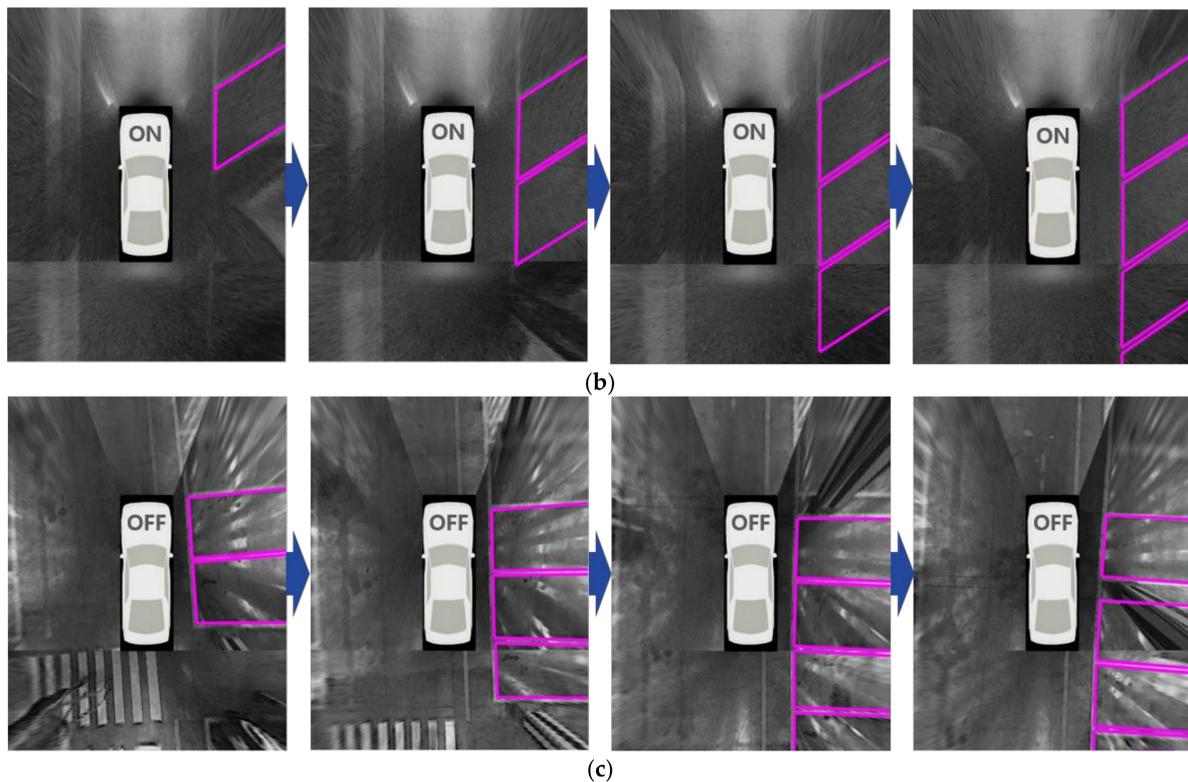


Figure 14. Parking slot recognition results of the proposed method in image sequences. (a) Daytime; (b) Nighttime; (c) Underground.

Since no other method can handle both various types of parking slot markings and a variety of severe lighting conditions, we carefully selected two previous methods that have different strengths and used them for performance comparison. One is the method in [4] that, among the existing methods, can recognize the largest number of parking slot marking types. The other is the method in [5] that, among the existing method, can handle the most severe lighting conditions. The method in [4] uses a hierarchical tree structure consisting of corner, junction, and slot. Table 3 shows the performance of this method in the test dataset. This method correctly recognizes 385 parking slots out of 609 and produces 239 false positives, and its recall and precision are 63.22% and 61.70%, respectively. While this method can recognize various types of parking slot markings, it is sensitive to severe lighting conditions because it uses a local feature called a corner. Since the corner feature is sensitive to amplified noise that is usually present in images captured under dim lighting, this method shows poor performance in both the nighttime and underground conditions. However, it produces a relatively good performance during the day where lighting conditions are favorable. Compared with the proposed method, this method yields a 32.02% lower recall and a 35.94% lower precision. The method in [5] uses an approach that first detects a guide line and then finds separating lines with the help of that guide line. Table 4 shows the performance of this method in the test dataset. This method correctly recognizes 251 parking slots out of 609 and produces 9 false positives, and its recall and precision are 41.22% and 96.54%, respectively. While this method is robust against unfavorable lighting conditions, it can only handle the rectangular parking slot markings. Thus, this method shows a good performance in underground settings where only rectangular parking slot markings exist. However, in terms of recall, it gives poor performances in daytime and nighttime conditions, as such settings include other types of parking slot markings. The precision of this method is quite high because it uses strong geometric constraints, which result in a small number of false positives. Compared with the proposed method, this method gives a 54.02% lower recall and a 1.10% lower precision. Table 5 summarizes

Tables 2–4 by showing only the total performances of the three methods. In this table, it can clearly be confirmed that the proposed method outperforms these two other methods.

Table 3. Performance of the method in [4].

Environment	#slot	#TP	#FP	Recall	Precision
Daytime	235	219	25	93.19%	89.75%
Nighttime	214	65	161	30.37%	28.76%
Underground	160	101	53	63.13%	65.58%
Total	609	385	239	63.22%	61.70%

Table 4. Performance of the method in [5].

Environment	#slot	#TP	#FP	Recall	Precision
Daytime	235	65	1	27.66%	98.48%
Nighttime	214	36	0	16.82%	100.00%
Underground	160	150	8	93.75%	94.94%
Total	609	251	9	41.22%	96.54%

Table 5. Performance of all three methods.

Method	#slot	#TP	#FP	Recall	Precision
Proposed method	609	580	14	95.24%	97.64%
Method in [4]	609	385	239	63.22%	61.70%
Method in [5]	609	251	9	41.22%	96.54%

9. Conclusions

This paper proposes a parking slot recognition method, which is an essential component of autonomous driving. Compared with previous methods, the proposed method is more useful and practical because it can be fully automated, recognize various types of parking slot markings, and operate under unfavorable lighting conditions. To achieve these advantages, this method generates parking slot candidates using separating lines, detects their entrances using both line and corner features, and combines sequentially found separating lines and parking slots by tracking them with the vehicle odometry. Furthermore, the proposed method uses only sensors already mounted on off-the-shelf vehicles so that it can be readily commercialized. Experimental results show that the proposed method recognizes various types of parking slot markings under a variety of lighting conditions, and outperforms the previous methods.

Acknowledgments: This research was supported in part by the Hyundai Motor Company and the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2013-0-00680) supervised by the IITP (Institute for Information & communications Technology Promotion).

Author Contributions: J.K.S. developed the algorithm and performed the experiments. H.G.J. developed the system architecture and analyzed the experimental results. Both authors wrote the paper together.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Heimberger, M.; Horgan, J.; Hughes, C.; McDonald, J.; Yogamani, S. Computer Vision in Automated Parking Systems: Design, Implementation and Challenges. *Image Vis. Comput.* **2017**, *68*, 88–101. [[CrossRef](#)]
2. Correa, A.; Boquet, G.; Morell, A.; Vicario, J.L. Autonomous Car Parking System through a Cooperative Vehicular Positioning Network. *Sensors* **2017**, *17*, 848. [[CrossRef](#)] [[PubMed](#)]

3. Jung, H.G.; Lee, Y.H.; Kim, J. Uniform user interface for semiautomatic parking slot marking recognition. *IEEE Trans. Veh. Technol.* **2010**, *59*, 616–626. [[CrossRef](#)]
4. Suhr, J.K.; Jung, H.G. Sensor Fusion-Based Vacant Parking Slot Detection and Tracking. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 21–36. [[CrossRef](#)]
5. Suhr, J.K.; Jung, H.G. Automatic Parking Space Detection and Tracking for Underground and Indoor Environments. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5687–5698. [[CrossRef](#)]
6. Hiramatsu, S.; Hibi, A.; Tanaka, Y.; Kakinami, T.; Iwata, Y.; Nakamura, M. Rearview camera based parking assist system with voice guidance. In Proceedings of the SAE World Congress and Exhibition, Detroit, MI, USA, 11–15 March 2002.
7. Furutani, M. Obstacle detection systems for vehicle safety. In Proceedings of the Convergence International Congress & Exposition on Transportation Electronics, Detroit, MI, USA, 13 October 2004.
8. Kageyama, Y. Look, No Hand! New Toyota Parks Itself. Available online: <http://www.cnn.com/> (accessed on 14 January 2004).
9. Jung, H.G.; Choi, C.G.; Yoon, P.J.; Kim, J. Novel user interface for semi-automatic parking assistance system. In Proceedings of the 31st FISITA World Automotive Congress, Yokohama, Japan, 22–27 October 2006.
10. Ford FUSION. Available online: https://www.ford.com/cars/fusion/features/smart/#enhanced_active_park_assist (accessed on 21 February 2018).
11. Mercedes-Benz E-Class. Available online: <https://www.mercedes-benz.com/en/mercedes-benz/innovation/the-remote-parking-pilot-in-the-new-e-class-video> (accessed on 21 February 2018).
12. Toyota PRIUS. Available online: <https://www.toyota.com/prius/prius-features> (accessed on 21 February 2018).
13. Hyundai Genesis. Available online: <http://www.hyundai.com/eu/en/Showroom/Cars/Genesis/PIP/index.html> (accessed on 21 February 2018).
14. Vestri, C.; Bougnoux, S.; Bendahan, R.; Fintzel, K.; Wybo, S.; Abad, F.; Kakinami, T. Evaluation of a vision-based parking assistance system. In Proceedings of the 8th IEEE International Conference on Intelligent Transportation Systems, Vienna, Austria, 3–6 September 2005.
15. Suhr, J.K.; Jung, H.G.; Bae, K.; Kim, J. Automatic free parking space detection by using motion stereo-based 3D reconstruction. *Mach. Vis. Appl.* **2010**, *21*, 163–176. [[CrossRef](#)]
16. Unger, C.; Wahl, E.; Ilic, S. Parking assistance using dense motion stereo. *Mach. Vis. Appl.* **2014**, *25*, 561–581. [[CrossRef](#)]
17. Kaempchen, N.; Franke, U.; Ott, R. Stereo vision based pose estimation of parking lots using 3-D vehicle models. In Proceedings of the IEEE Intelligent Vehicles Symposium, Versailles, France, 17–21 June 2002.
18. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J.H. 3D vision system for the recognition of free parking site location. *Int. J. Autom. Technol.* **2006**, *7*, 361–367.
19. Jung, H.G.; Kim, D.S.; Kim, J. Light stripe projection-based target position designation for intelligent parking-assist system. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 942–953. [[CrossRef](#)]
20. Hashizume, A.; Ozawa, S.; Yanagawa, H. An approach to detect vacant parking space in a parallel parking area. In Proceedings of the 5th European Congress on Exhibition Intelligent Transport Systems and Services, Hannover, Germany, 4–6 June 2005.
21. Abad, F.; Bendahan, R.; Wybo, S.; Bougnoux, S.; Vestri, C.; Kakinami, T. Parking space detection. In Proceedings of the 14th World Congress on Intelligent Transport Systems, Beijing, China, 12–15 October 2007.
22. Jung, H.G.; Cho, Y.H.; Yoon, P.J.; Kim, J. Scanning laser radar-based target position designation for parking aid system. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 406–424. [[CrossRef](#)]
23. Zhou, J.; Navarro-Serment, L.E.; Hebert, M. Detection of parking spots using 2D range data. In Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012.
24. Scheunert, U.; Fardi, B.; Mattern, N.; Wanielik, G.; Keppeler, N. Free space determination for parking slots using a 3D PMD sensor. In Proceedings of the IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 12–15 June 2007.
25. Görner, S.; Rohling, H. Parking lot detection with 24 GHz radar sensor. In Proceedings of the 3rd International Workshop on Intelligent Transportation, Hamburg, Germany, 14–15 March 2006.

26. Schmid, M.R.; Ates, S.; Dickmann, J.; Hundelshausen, F.; Wuensche, H.J. Parking space detection with hierarchical dynamic occupancy grids. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011.
27. Dube, R.; Hahn, M.; Schutz, M.; Dickmann, J.; Gingras, D. Detection of Parked Vehicles from a Radar Based Occupancy Grid. In Proceedings of the IEEE Intelligent Vehicles Symposium, Ypsilanti, MI, USA, 8–11 June 2014.
28. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. Structure analysis based parking slot marking recognition for semi-automatic parking system. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, Hong Kong, China, 17–19 August 2006.
29. Jung, H.G. Semi-automatic parking slot marking recognition for intelligent parking assist systems. *J. Eng.* **2014**, *2014*, 8–15. [[CrossRef](#)]
30. Xu, J.; Chen, G.; Xie, M. Vision-guided automatic parking for smart car. In Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 5 October 2000.
31. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. Parking slot markings recognition for automatic parking assist system. In Proceedings of the IEEE Intelligent Vehicles Symposium, Tokyo, Japan, 13–15 June 2006.
32. Wang, C.; Zhang, H.; Yang, M.; Wang, X.; Ye, L.; Guo, C. Automatic Parking Based on a Bird's Eye View Vision System. *Adv. Mech. Eng.* **2014**, *2014*, 1–13. [[CrossRef](#)]
33. Tanaka, Y.; Saiki, M.; Katoh, M.; Endo, T. Development of image recognition for a parking assist system. In Proceedings of the 14th World Congress on Intelligent Transport Systems, London, UK, 9–13 October 2006.
34. Houben, S.; Komar, M.; Hohm, A.; Luke, S.; Neuhausen, M.; Schlipfing, M. On-Vehicle Video-Based Parking Lot Recognition with Fisheye Optics. In Proceedings of the 16th IEEE International Conference on Intelligent Transportation Systems, Hague, Netherlands, 6–9 October 2013.
35. Du, X.; Tan, K.K. Autonomous Reverse Parking System Based on Robust Path Generation and Improved Sliding Mode Control. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1225–1237. [[CrossRef](#)]
36. Hamada, K.; Hu, Z.; Fan, M.; Chen, H. Surround View based Parking Lot Detection and Tracking. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015.
37. Li, L.; Zhang, L.; Li, X.; Liu, X.; Shen, Y.; Xiong, L. Vision-based parking-slot detection: A benchmark and a learning-based approach. In Proceedings of the IEEE International Conference on Multimedia and Expo, Hong Kong, China, 10–14 July 2017.
38. Suhr, J.K.; Jung, H.G. Full-automatic recognition of various parking slot markings using a hierarchical tree structure. *Opt. Eng.* **2013**, *52*, 037203. [[CrossRef](#)]
39. Suhr, J.K.; Jung, H.G. Fully-automatic recognition of various parking slot markings in Around View Monitor (AVM) image sequences. In Proceedings of the 15th IEEE International Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012.
40. Lee, S.; Seo, S.W. Available parking slot recognition based on slot context analysis. *IET Intell. Transp. Syst.* **2016**, *10*, 594–604. [[CrossRef](#)]
41. Lee, S.; Hyeon, D.; Park, G.; Baek, I.; Kim, S.W.; Seo, S.W. Directional-DBSCAN: Parking-slot Detection using a Clustering Method in Around-View Monitoring System. In Proceedings of the IEEE Intelligent Vehicles Symposium, Gothenburg, Sweden, 19–22 June 2016.
42. Hyundai, Azera. Available online: <https://www.hyundai.com/worldwide/en/cars/azera/highlights> (accessed on 21 February 2018).
43. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 2008.
44. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
45. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 1–6 September 1988.

