Article

Vision-Based Parking-Slot Detection: A Benchmark and A Learning-Based Approach

Lin Zhang (ID)**, Xiyuan Li, Junhao Huang, Ying Shen* and Dongqing Wang**

School of Software Engineering, Tongji University, Shanghai 201804, China; cslinzhang@tongji.edu.cn (L.Z.);
1641465@tongji.edu.cn (X.L.); 1731557@tongji.edu.cn (J.H.); wangdongqing@tongji.edu.cn (D.W.)
* Correspondence: yingshen@tongji.edu.cn; Tel.: +86-21-6958-9976

Abstract: Recent years have witnessed a growing interest in developing automatic parking systems in the field of intelligent vehicles. However, how to effectively and efficiently locating parking-slots using a vision-based system is still an unresolved issue. Even more seriously, there is no publicly available labeled benchmark dataset for tuning and testing parking-slot detection algorithms. In this paper, we attempt to fill the above-mentioned research gaps to some extent and our contributions are twofold. Firstly, to facilitate the study of vision-based parking-slot detection, a large-scale parking-slot image database is established. This database comprises 8600 surround-view images collected from typical indoor and outdoor parking sites. For each image in this database, the marking-points and parking-slots are carefully labeled. Such a database can serve as a benchmark to design and validate parking-slot detection algorithms. Secondly, a learning-based parking-slot detection approach, namely PSD_L, is proposed. Using PSD_L, given a surround-view image, the marking-points will be detected first and then the valid parking-slots can be inferred. The efficacy and efficiency of PSD_L have been corroborated on our database. It is expected that PSD_L can serve as a baseline when the other researchers develop more sophisticated methods.

Keywords: parking assistance systems; parking-slot detection; AdaBoost; decision tree

1. Introduction

Owing to the increased demand for autonomous driving [1,2], the development of automatic parking assistant systems (PAS) has become a topic of intense study. A typical automatic parking system starts by target position designation, i.e., locating a vacant parking space for the vehicle. To resolve the problem of target position designation, various solutions have been proposed in the literature and they roughly fall into two categories, infrastructure-based ones and in-vehicle sensor based ones.

A typical infrastructure-based method usually resorts to a pre-built map and infrastructure-level sensors. Desirable target positions are designated based on the infrastructure and the vehicle receives the parking information through vehicle-infrastructure communication [3–8]. Obviously, the infrastructure-based methods have an advantage of managing all parking-slots; however, they may not be applicable in a short time due to the requirement of additional hardware installation on current park-sites and vehicles.

When the infrastructure is not available, the PAS may need to depend on an in-vehicle sensor-based method to identify an appropriate parking space. Such methods can perceive the available parking spaces during the movement of the vehicle by making only use of sensors it carries. These methods can be categorized into two groups, the free-space-based ones and the parking-slot-marking-based ones.

A free-space-based approach designates a target position by recognizing a vacant space between two adjacent vehicles. This is the most widely used approach as it can be implemented using various range-finding sensors, such as ultrasonic sensors [9–15], laser scanners [16–18], short-range radars [19–21], structured light [22], depth cameras [23], stereo cameras [24–29], etc. However, the free-space-based approach has an inherent drawback in that it cannot find free spaces when there is no adjacent vehicles and its accuracy depends on the positions and poses of adjacent vehicles.

Parking-slot-markings refer to the regular line segments painted on the ground, indicating the valid areas for parking. Unlike the free-space-based approaches, a parking-slot-marking-based approach finds parking spaces by recognizing visual slot-markings from image sequences acquired by cameras mounted on the vehicle and thus its performance does not depend on the existence or poses of adjacent vehicles. Moreover, in most cases, slot-markings can provide more accurate parking information than "free-space". Meanwhile, most car manufacturers have started to produce vehicles equipped with wide FOV (field of view) imaging sensors, usually used in an AVM (around view monitoring) system. For these reasons, the parking-slot-marking-based approach has began to draw a lot of attention recently in the field of parking space detection, which is also our focus of this paper.

1.1. Related Work

In this paper, we focus on studying the problem of in-vehicle vision-based parking-slot detection. Usually to fulfill this task, a bird's-eye-view image or a surround-view image (obtained by synthesizing multiple bird's-eye-view images) is generated in real-time first and then the parking-slots are detected from it. Representative work in this area will be reviewed here.

The research in this area started from Xu et al.'s pioneer work [30]. In [30], Xu et al. claimed that the colors of parking-slot's markings are quite uniform and different from the background and thus they trained a neural network to segment parking-slot-markings. Then, they estimated two perpendicular lines as the parking-slot contour. The drawback of this simple parking-slot model is that the type (perpendicular or parallel) of the parking-slot cannot be obtained. In [31], Jung et al. presented a one-touch method that recognizes the line segments of a parking-slot by checking the directional gradient based on a manually designated point inside the target parking-slot. Since this method can handle only a single type of parking-slot, Jung et al. [32] extended it to a two-touch method. This method can recognize various types of parking-slot-markings based on two points of a parking-slot entrance provided by the driver. In [33], Du and Tan developed a reverse parking system. To detect the parking-slot, they first applied a ridge detector on the image and then medial axes of the slot markings can be obtained after the steps of noise filtering, connected components labeling, and removal of components with a small number of pixels. However, their system relies on human drivers to identify an empty parking-slot first before initiating the parking process. The apparent drawback of the methods in [31–33] is that they are not fully automatic.

Fully automatic parking-slot detection methods are developed along two main streams, the line-based ones and the corner-based ones, according to the primitive visual features they extract. In [34], Jung et al. assumed that marking-lines consist of lines with a fixed width and recognized them by applying peak-pair detection and clustering in Hough space [35]. Separating marking-line segments were at last recognized by a T-shaped template matching. Following the similar idea as Jung et al.'s work [34], Wang et al. [36] proposed to detect marking-line segments in Radon space [37] as they considered that Radon transform has a better noise-tolerance ability and is more robust than Hough transform. One potential drawback of the methods in [34,36] is their sensitiveness to the marking-line width. In [38], after obtaining the edge map of the surround-view image, Hamada et al. made use of the probabilistic Hough transform [39] to extract all line segments and then they inferred the valid parking-slots based on some geometric constraints. In [40], Suhr and Jung designed a parking-slot detection approach specially for underground and indoor environments. In their approach, the guide line is detected first and then the separating lines are detected. To detect the guide line, they utilized the RANSAC (RANdom SAmpling Consensus) [35] algorithm for robust

line fitting from edge pixels and to detect the separating lines, they used a distance transform-based chamfer matching [41]. The limitations of Suhr and Jung's method mainly lie in two points: (1) it can only detect perpendicular parking-slots but cannot detect parallel ones; (2) it requires that the guide line of the parking-slots should be visible. In [42], Lee and Seo proposed the so-called "cone-hat" filter for line-marking extraction and then the extracted line features were assigned to parking-line segments via entropy-based clustering. After that, the sequential RANSAC algorithm [43] was utilized to fit parking-lines from clusters. At last, parking-slot candidates were generated and then validated by a Bayesian network model [44].

Different from line-based ones, a few other parking-slot detection methods are based on corners and among them Suhr and Jung's work is a representative one [45,46]. The method proposed in [45,46] detects corners via the Harris corner detector [47] first and then generates junctions by combining these corners; finally, parking-slots are inferred from junction pairs. Thus, the success rate of this method highly depends on the robustness of the Harris corner detector.

1.2. Our Motivations and Contributions

Having investigated the literature, we find that in the field of vision-based parking-slot detection, there is still large room for further improvement in at least two aspects.

Firstly, vision-based parking-slot detection is not an easy problem. The backgrounds can vary significantly in different parking sites as the ground surfaces can have various textures. More seriously, the illumination condition can change greatly, especially at the outdoor parking sites. Though a plethora of solutions have been proposed, nearly all the existing state-of-the-art ones are based on low-level visual features, such as lines and corners, detected by some low-level vision algorithms (such as Canny edge detector, Hough transform, Radon transform, Harris corner detector, etc.). These features actually are not quite distinguishable and even worse, they are unstable and unrepeatable to changes in environment aroused by noise, clutter, illumination variation, etc. These serious drawbacks inevitably limit the performance of methods depending on low-level features. Actually, how to efficiently and accurately detect parking-slots using vision-based methods in uncontrolled environment is still a great challenge. In addition, none of the researchers have ever published their source codes, which undoubtedly hinders the development of this field.

Secondly, vision-based parking-slot detection is actually a "pattern classification" problem. To cope with it, a publicly-available large-scale benchmark dataset is highly desired, which in fact is indispensable for researchers to design and compare the detection algorithms. Unfortunately, previous researchers have never published their datasets and thus such a dataset is still lacking in this area. Without a common benchmark dataset, it is impossible to make fair comparisons among different parking-slot detection algorithms.

In this work, we make an attempt to fill the aforementioned research gaps to some extent. Our contributions in this paper are summarized as follows:

(1) A data-driven learning-based approach, namely PSD_L (*Parking-Slot Detection* based on *Learning*), is proposed for parking-slot detection. Given a surround-view image, PSD_L detects the marking-points using a pre-trained detector first and then infers the valid parking-slots from them. Marking-points are defined as the cross-points of parking-lines. In Figure 1, examples of marking-points on two surround-view images are marked by yellow circles. The advantages of PSD_L over the existing parking-slot detection methods are twofold. Firstly, PSD_L is built upon marking-point patterns, which are more distinguishable and stable than primitive visual features, such as lines or corners. Secondly, for detecting marking-points, PSD_L adopts a data-driven learning-based strategy, which is much more robust to changes of imaging conditions than low-level vision algorithms. To our knowledge, our work is the first to use learning-based techniques to detect visual patterns in the field of parking-slot detection. PSD_L can detect both the perpendicular and the parallel parking-slots. Besides, it can work equally well in indoor and outdoor environments. Its efficacy and efficiency have been thoroughly

evaluated in experiments. As we have already published its source code, PSD_L can serve as a baseline when the other researchers develop more sophisticated algorithms.

(2) To facilitate the study of vision-based parking-slot detection, we have established a large-scale benchmark dataset and have made it publicly available to the research community. This dataset comprises 8600 surround-view images collected from typical indoor and outdoor parking sites with our experimental car and all the images were manually labeled with care. Such a dataset can be employed for training and testing new parking-slot detection algorithms. To our knowledge, this dataset is the first of its kind in the field of vision-based parking-slot detection. Please refer to Section 4.1 for more details about this dataset.
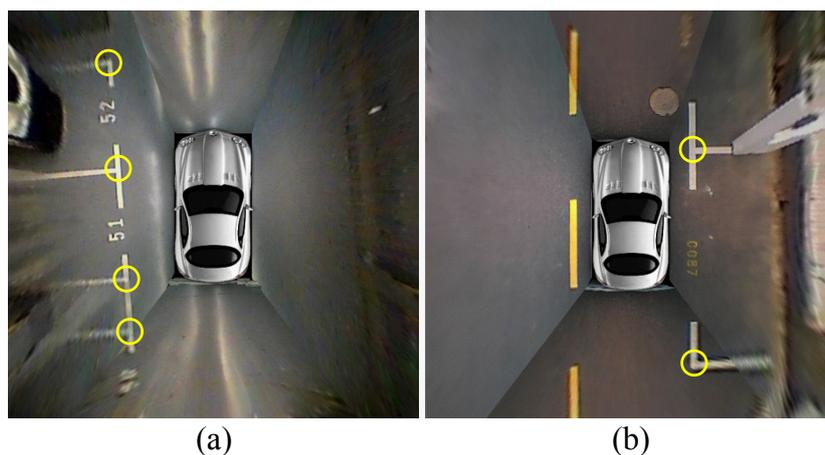

(a) (b)

Figure 1. (**a**,**b**) are two surround-view images taken from two typical parking sites. (**a**) is taken from an underground parking site while (**b**) is taken from an outdoor parking site. Parking-slots in (**a**) are perpendicular while the ones in (**b**) are parallel. Yellow circular marks indicate the positions of marking-points.

To make the results reported in this paper fully reproducible, the collected benchmark dataset and all the relevant source code have been made publicly available at https://cslinzhang.github.io/ps/.

The remainder of this paper is organized as follows. Section 2 presents the steps to generate the surround-view image. Section 3 introduces our novel approach PSD_L for parking-slot detection. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Surround-View Generation

The high-level structure of a typical vision-based PAS system is shown in Figure 2. In such a system, the surround-view image sequence is synthesized in real-time from outputs of multiple wide-angle cameras mounted on the vehicle. Taking the surround-view image as input, the parking-slot detection module detects the valid parking-slot(s) and then send the coordinates of parking-slot(s) with respect to the vehicle coordinate system to the decision module for further process (For example, usually the decision module will use an ultrasonic radar to test whether the parking-slot returned by the parking-slot detection module is vacant or not). In this section, we briefly introduce the steps for surround-view generation.

The automotive surround-view camera system normally consists of 4 to 6 wide-angle cameras mounted around the vehicle, each facing a different direction. On our experimental car, 4 low-cost fish-eye cameras are mounted. From these camera inputs, a 360° surround-view image around the vehicle can be synthesized.
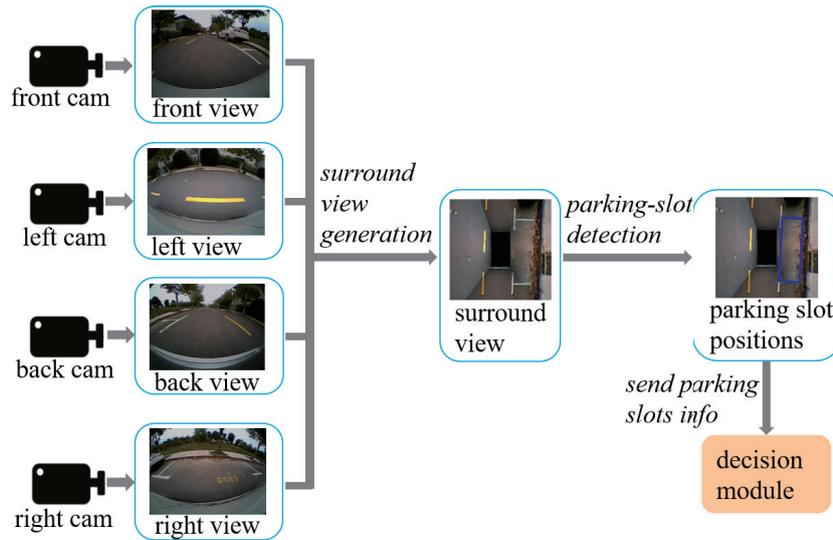
Figure 2. The high-level structure of a typical vision-based parking assistance system. It usually comprises two key modules, the surround-view synthesis module and the parking-slot detection module taking the surround-view image as input.

Actually, the surround-view is the composite view of four bird's-eye-views, the front-view, the left view, the rear view, and the right view. Two adjacent bird's-eye-views overlap with each other. The key step for generating the bird's-eye-view image is to build an off-line lookup table $T_{B \to F}$, mapping a point x_B on the bird's-eye-view image to a position x_F on the input fish-eye image, by conducting a set of calibration operations. To determine $T_{B \to F}$, we need to determine $P_{B \to W}$ the transformation matrix from the bird's-eye-view coordinate system to the world coordinate system, $P_{W \to U}$ the transformation matrix from the world coordinate system to the undistorted input image coordinate system, and $T_{U \to F}$ the lookup table mapping a point on the undistorted input image to a position on the original input fish-eye image. Figure 3 illustrates the relationships among the coordinate systems involved in bird's-eye-view generation.

The distortion coefficients of the fish-eye camera can be estimated by Zhang's calibration method [48,49] and accordingly, the mapping table $T_{U \to F}$ can be obtained.
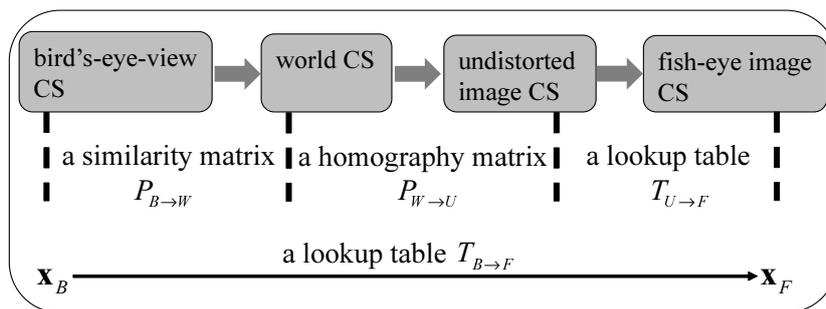

Figure 3. The relations among the coordinate systems involved in surround-view generation. "CS" is short for "coordinate system".

$P_{B \to W}$ is a similarity transformation matrix, which can be determined straightforwardly if the size (measured by pixels) of the bird's-eye-view image and the corresponding physical visible range (measured by millimeters) are determined beforehand. As illustrated in Figure 4, suppose that the size

of the bird's-eye-view image is $M \times N$ and the height of the corresponding physical area is H mm, it is easy to verify that $P_{B \rightarrow W}$ can be expressed as,

$$P_{B \rightarrow W} = \begin{bmatrix} \frac{H}{M} & 0 & -\frac{HN}{2M} \\ 0 & -\frac{H}{M} & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

To estimate the homography matrix $P_{W \rightarrow U}$, a calibration field is used, in which the world coordinates of the feature points are known beforehand. An image of the calibration field is captured and then undistorted using $T_{U \rightarrow F}$. Then, on the undistorted image, N_F feature points (N_F cannot be smaller than 4 and based on our experience it is usually set between 10 and 20.) can be manually selected as illustrated in Figure 5. Of course, for each selected feature point i we know its coordinates x_W^i in the world coordinate system and its coordinates x_U^i in the undistorted image coordinate system. The relationship between x_W^i and x_U^i is $x_U^i = P_{W \rightarrow U} x_W^i$. Therefore, $P_{W \rightarrow U}$ can be estimated by solving a least-square problem based on correspondence pairs $\{x_W^i, x_U^i\}_{i=1}^{N_F}$.
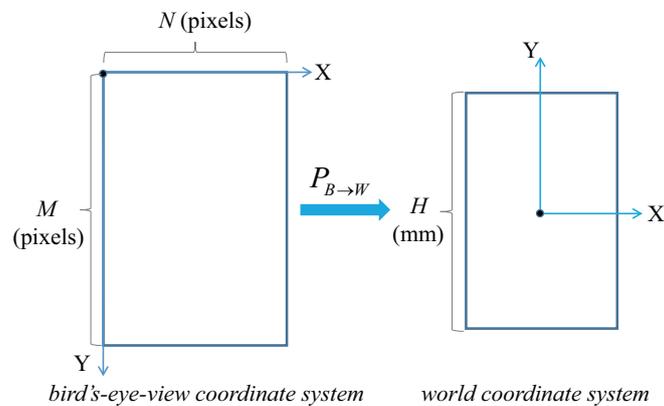

Figure 4. The relationship between the bird's-eye-view coordinate system and the world coordinate system.
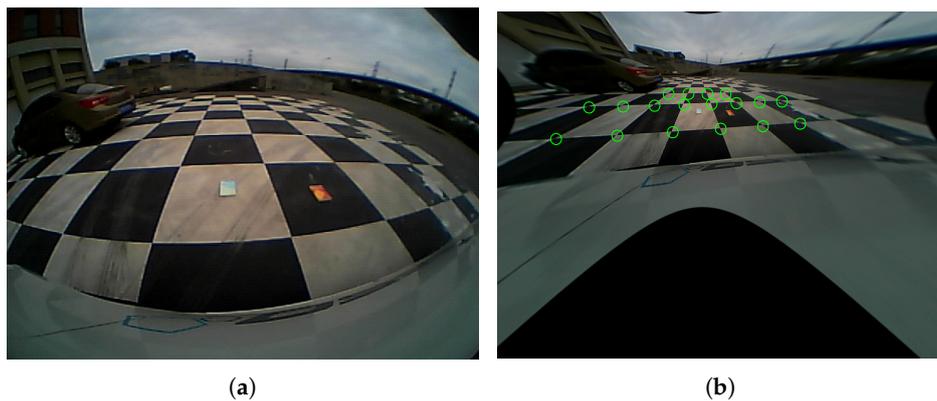

(**a**)   (**b**)

Figure 5. Illustration for the calibration process to obtain the homography matrix $P_{W \rightarrow U}$ between the world coordinate system and the undistorted image coordinate system. (**a**) is the original fish-eye image of the calibration field and its undistorted version is shown in (**b**); On (**b**), N_F feature points are manually selected as marked by yellow circles. For the selected feature points, their coordinates x_W^i in the world coordinate system and their coordinates x_U^i in the undistorted image coordinate system are known.

When the matrices $P_{B \rightarrow W}$ and $P_{W \rightarrow U}$, and the mapping table $T_{U \rightarrow F}$ are ready, the mapping table $T_{B \rightarrow F}$ can be finally determined. In our case, we use four fish-eye cameras and thus we can have four

mapping tables $\left\{ T_{B \to F}^i \right\}_{i=1}^4$, each being responsible for generating one bird's-eye-view. To synthesize the surround-view, at last we need to determine the stitching line between each pair of adjacent bird's-eye-views. With the four stitching lines, for a point on the surround-view, we can know which bird's-eye-view it should come from and accordingly, which mapping table and the associated input fish-eye image should be used to determine its pixel value.

In Figure 6, we show an example for surround-view generation. Figure 6a–d is four fish-eye images and Figure 6e is the surround-view image synthesized from Figure 6a–d.
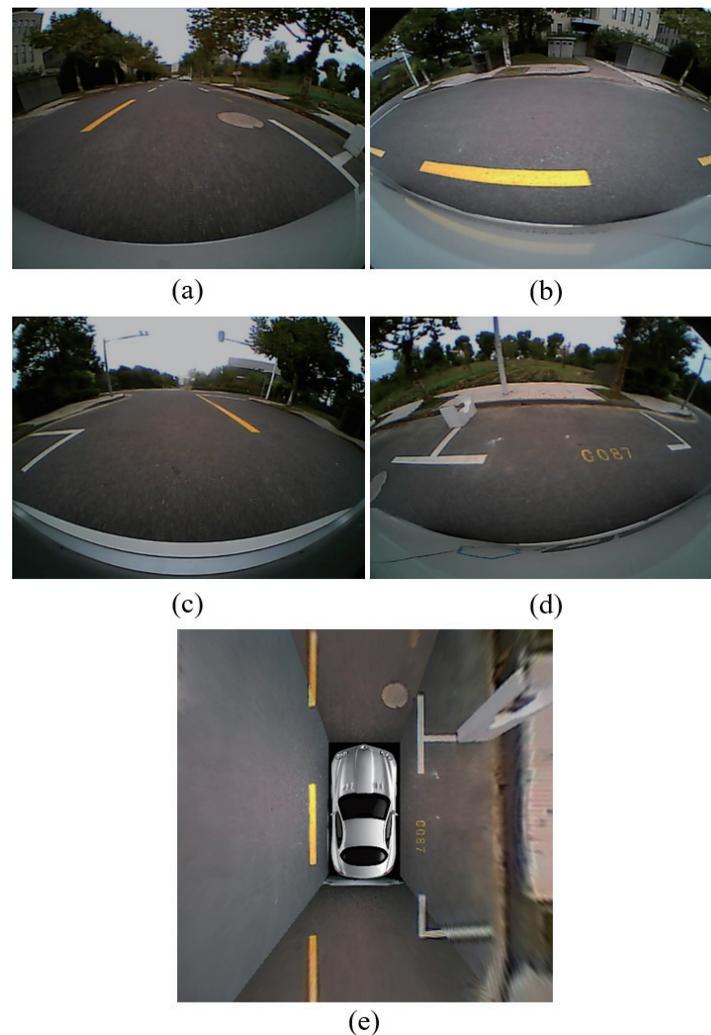

(a)　　　　　　　　　　(b)

(c)　　　　　　　　　　(d)

(e)

Figure 6. Images (**a**–**d**) is captured from the front, the left, the back, and the right fish-eye cameras, respectively; (**e**) is the surround-view image synthesized from (**a**–**d**).

3. PSD_L: A Learning Based Approach for Detecting Parking-Slots

In this section, our proposed parking-slot detection approach PSD_L will be presented in detail. It is designed to detect typical perpendicular and parallel parking-slots as illustrated in Figure 7. Based on Figure 7, it can be seen that the entrance-lines of parking-slots that can be correctly detected by our PSD_L approach are composed of "T-shaped" or "L-shaped" marking-point patterns. PSD_L actually comprises two phases, detecting marking-points and then inferring valid parking-slots from detected marking-point patterns.
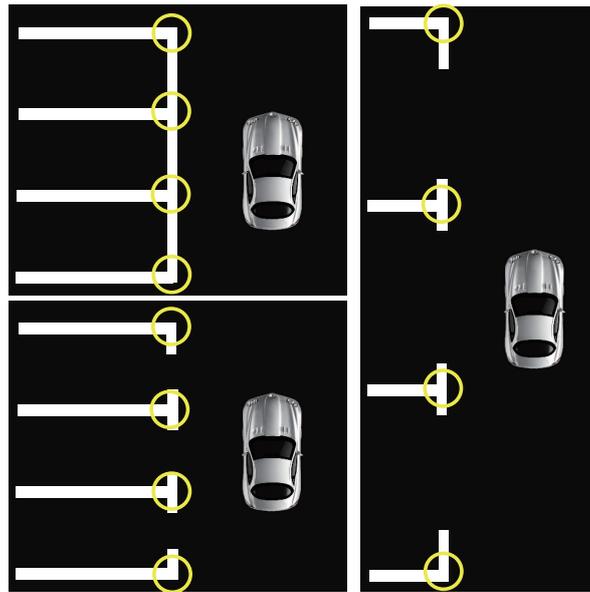
Figure 7. Typical types of parking-slots that the proposed algorithm PSD_L can detect.

3.1. Marking-Point Detection

A marking-point pattern refers to a local image patch centered at a cross-point of parking-line segments, as indicated by the yellow circular marks in Figure 7. To detect marking-point patterns, a binary classifier is designed, which takes a local image patch as input and outputs a binary value indicating whether the input is a marking-point pattern or not. Based on the labeled benchmark dataset (see Section 4.1 for details), the positive training image patches can be simply extracted from labeled surround-view images while the negative training patches are extracted using a bootstrapping process during training. It needs to be stressed that based on our experience, chrominance information is unstable for parking-slot detection since the lighting conditions can vary significantly in different parking occasions. Hence, the colorful training surround-view images are first converted to their gray-scale versions. Similarly, at the testing stage, the input surround-view image will also be converted to gray-scale.

To train the marking-point detector, features and the classifier model need to be determined. With respect to features, three types of features are used. The first feature is the normalized intensity. The second feature is the gradient magnitude. Given an image $I(\boldsymbol{x})$, its partial derivatives $G_x(\boldsymbol{x})$ and $G_y(\boldsymbol{x})$ can be computed by filtering $I(\boldsymbol{x})$ with Sobel gradient operators [35]. Then, the gradient magnitude of $I(\boldsymbol{x})$ is computed as $GM(\boldsymbol{x}) = \sqrt{G_x^2(\boldsymbol{x}) + G_y^2(\boldsymbol{x})}$. The last type of feature is the oriented gradient magnitudes [50]. The ith oriented gradient magnitude map $Q_i(\boldsymbol{x})$ of $I(\boldsymbol{x})$ is defined as,

$$Q_i(\boldsymbol{x}) = GM(\boldsymbol{x}) \cdot \mathbf{1}\left(\Theta(\boldsymbol{x}) == i\right), i = 1, 2..., N_O \tag{2}$$

where $\mathbf{1}\left(\cdot\right)$ is the indicator function, $\Theta(\boldsymbol{x})$ is the quantized gradient angle of $I(\boldsymbol{x})$ and the range of $\Theta(\boldsymbol{x})$ values is $[1, N_O]$. Given an image patch, its normalized intensity map, its gradient magnitude map, and its oriented gradient maps are vectorized and then concatenated together as the final feature vector.

With respect to the classifier, we adopt the popular AdaBoost framework [51]. A boosted classifier H consisting of M weak classifiers can be represented as,

$$H(\boldsymbol{x}) = \sum_{t=1}^{M} \alpha_t h_t(\boldsymbol{x}) \tag{3}$$

where each h_t is a weak classifier and α_t is its associated weight. $\boldsymbol{x}$ is classified as positive if $H(\boldsymbol{x}) > 0$ and $H(\boldsymbol{x})$ serves as the confidence. In terms of the weak classifier, we use the shallow decision tree. The training is conducted in several stages and after each stage a bootstrapping process is performed to extract negative samples for the next stage.

At the detection stage, if the base AdaBoost classifier is utilized, it would be quite slow. A cascade structure is a common way to reduce the computational burden of evaluating a complex classifier over an entire image [52]. To simplify training, we use the "constant soft-cascade" strategy [53] instead of a real cascade structure. During training, for the node i of the tree h_t, we record its weighted log-ratio defined as,

$$l_t^i = \frac{1}{2}\alpha_t \ln \frac{p_i}{1 - p_i} \tag{4}$$

where p_i is the ratio of positive samples to all the samples reaching this node. l_t^i can measure the "positiveness" of samples reaching the node i of the tree h_t. At the testing stage, when a testing sample t reaches a node whose associated weighted log-ratio is smaller than a pre-defined constant threshold θ, the testing stops since the probability of t being positive will be quite low. Major steps for training the marking-point detector and applying it on a given surround-view image to detect marking-points are illustrated in Figure 8.
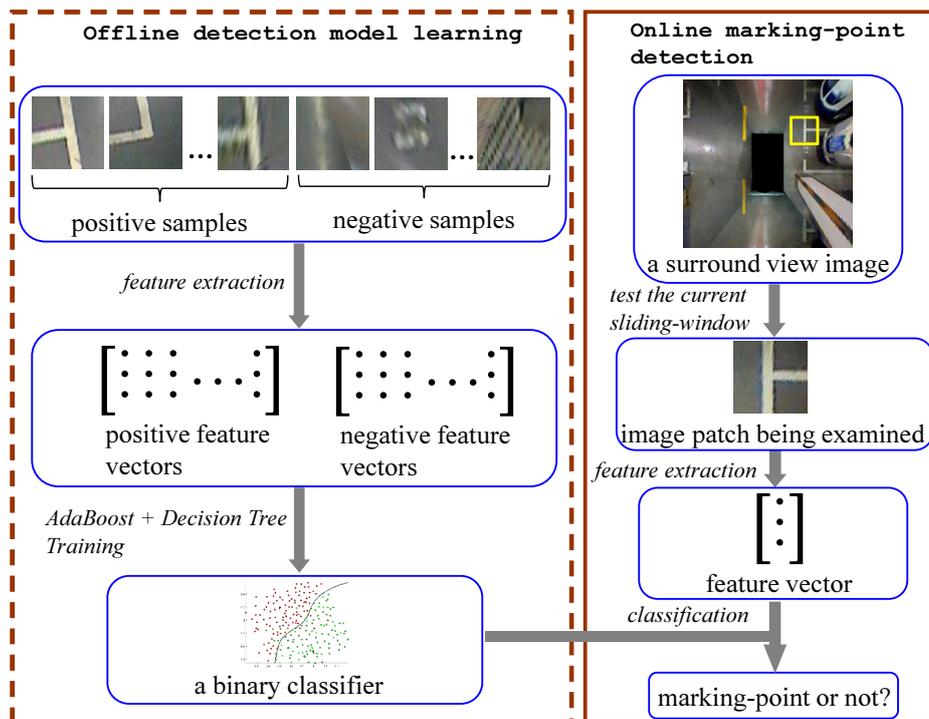

Figure 8. Processing flows for training the marking-point detector and applying it on a surround-view image to detect marking-points.

Another issue needs to be considered is that since marking-point patterns can be of any directions, a single detector would not be accurate enough. Thus, we train multiple detectors, each of them being responsible for detecting marking-point patterns whose directions are within a specific range. To keep the balance between the detection accuracy and the speed, we train four detectors $\{det_j\}_{j=1}^{4}$ and det_j is responsible for detecting marking-point patterns whose directions are within the range $\left[-\frac{3\pi}{4} + \frac{\pi}{2}j, -\frac{\pi}{4} + \frac{\pi}{2}j\right]$. In order to train multiple detectors, when we label positive samples of marking-point patterns, their directions are also labeled besides their positions. The directions of marking-slot patterns are labeled in a way as illustrated in Figure 9. When positive image patches

are extracted, they are first rotated to make their directions equal to 0. To train det_j, we rotate each positive image patch with a set of angles $\left\{ \frac{j-1}{2}\pi + \frac{\pi}{4}r_k \right\}_{k=1}^{K}$, where r_k is a random number uniformly distributed over $[-1, 1]$ and K defines the number of possible rotations. It can be known that if N_P positive samples are labeled, there will be KN_P positive samples for training det_j.

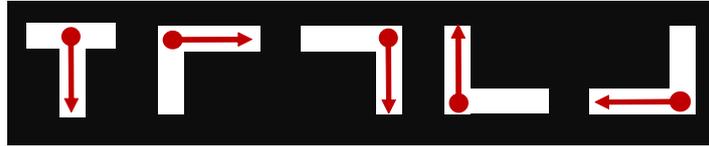In Figure 10, marking-point detection results on two typical surround-view images are shown.

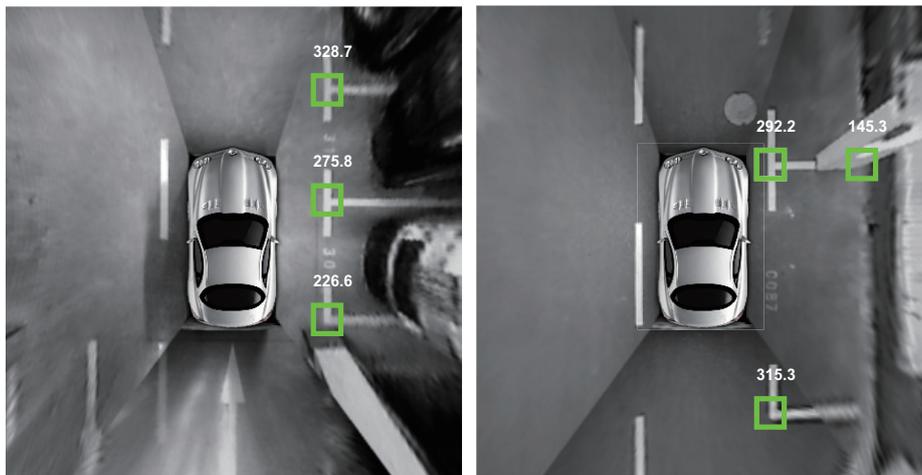**Figure 9.** Directions of the marking-point patterns.

Figure 10. Marking-point detection results on two typical surround-view images.

3.2. Parking-Slot Inference

Having detected the marking-points, we then can infer valid parking-slots from them based on some rules.

Given two marking-points P_1 and P_2, we need to check whether the line $\overrightarrow{P_1 P_2}$ can be a valid entrance-line of a parking-slot. At first, the length of $\overrightarrow{P_1 P_2}$ should satisfy some length constraints obtained from the prior knowledge. Then, we check whether the image patterns around P_1 and P_2 conform to the parking-slot model. By examining the ideal parking-slot models shown in Figure 7, it can be seen that to be a valid parking-slot entrance-line, image patterns around P_1 and P_2 should conform to one of the pattern models as shown in Figure 11. Inspired by this analysis, we propose to use Gaussian line templates to check the image patterns around P_1 and P_2. Six Gaussian line templates are used and their positions and orientations relative to P_1 and P_2 are illustrated in Figure 12. By checking the six responses $r_1 \sim r_6$ with a set of simple rules, whether $\overrightarrow{P_1 P_2}$ can be a valid entrance-line can be determined. Besides, whether the parking-slot is at the clockwise side or the anti-clockwise side of $\overrightarrow{P_1 P_2}$ can also be determined. The checking-rules are summarized in Algorithm 1.

After completing the abovementioned steps, we can get a set of "entrance-line" candidates. However, there is one case that they may contradict with each other as illustrated in Figure 13. In Figure 13, $\overrightarrow{P_1 P_2}$ and $\overrightarrow{P_2 P_3}$ are two entrance-line candidates for two perpendicular parking-slots while $\overrightarrow{P_1 P_3}$ is an entrance-line candidate for a parallel parking-slot. However, in fact, $\overrightarrow{P_1 P_3}$ is an invalid entrance-line. Hence, when the entrance-line candidates are ready, we need to remove the ones passing any valid marking-points to annihilate contradictions.
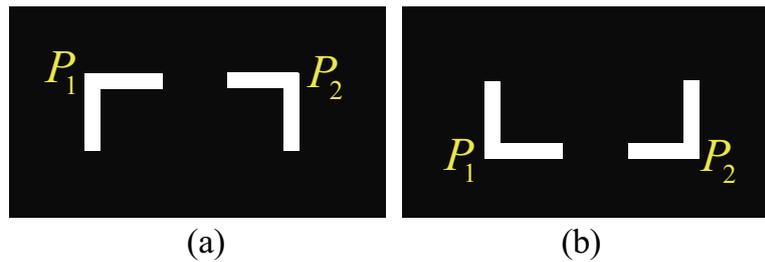
(a)   (b)

Figure 11. If P_1P_2 is a valid parking-slot entrance-line, the local image patterns around P_1 and P_2 should satisfy the pattern models (**a**) or (**b**). In (**a**), the parking-slot is at the clockwise side of P_1P_2 while in (**b**) the parking-slot is at the anti-clockwise side of P_1P_2.
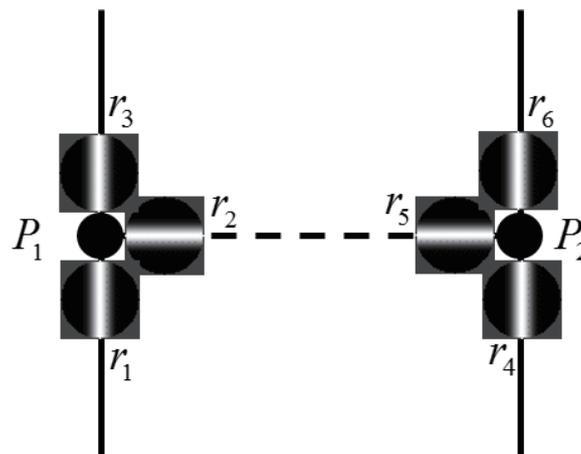

Figure 12. Six Gaussian line templates are used to examine the local image patterns around the marking-points P_1 and P_2.
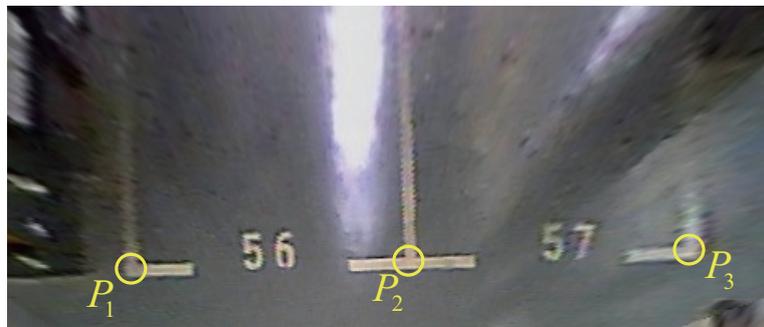

Figure 13. P_1P_2 and P_2P_3 are two entrance-line candidates for two perpendicular parking-slots while P_1P_3 is an entrance-line candidate for a parallel parking-slot. Actually, P_1P_3 is invalid and should be removed.

Finally, valid entrance-lines are remained, each of which represents a valid parking-slot. Their information is then sent to the decision-making module. It needs to be noted that the "depth" of the parking-slot is determined by prior knowledge. Moreover, if multiple parking-slots are detected on the current surround-view image, it is the responsibility of the decision-marking module to choose the most appropriate one. In Figure 14, six typical surround-view images with marked parking-slots detected by PSD_L are shown. From these samples, it can be seen that the proposed algorithm PSD_L has a strong capability for accurately detecting different types of parking-slots under various conditions.

Algorithm 1: Checking-Rules for Determining the Validity of P_1P_2 for Being an Entrance-Line and the Parking-Slot Orientation

---

t is a predefined threshold;

if $r_1 >> r_3$ *and* $r_2 >> r_3$ *and* $r_4 >> r_6$ *and* $r_5 >> r_6$ *and* $r_1 > t$ *and* $r_2 > t$ *and* $r_4 > t$ *and* $r_5 > t$
  then
    | $\overrightarrow{P_1P_2}$ is an entrance-line candidate;
    | The parking-slot is at the clockwise side of $\overrightarrow{P_1P_2}$;
  end
else if $r_3 >> r_1$ *and* $r_2 >> r_1$ *and* $r_6 >> r_4$ *and* $r_5 >> r_4$ *and* $r_3 > t$ *and* $r_2 > t$ *and* $r_6 > t$ *and* $r_5 > t$ **then**
    | $\overrightarrow{P_1P_2}$ is an entrance-line candidate;
    | The parking-slot is at the anti-clockwise side of $\overrightarrow{P_1P_2}$;
  end
else
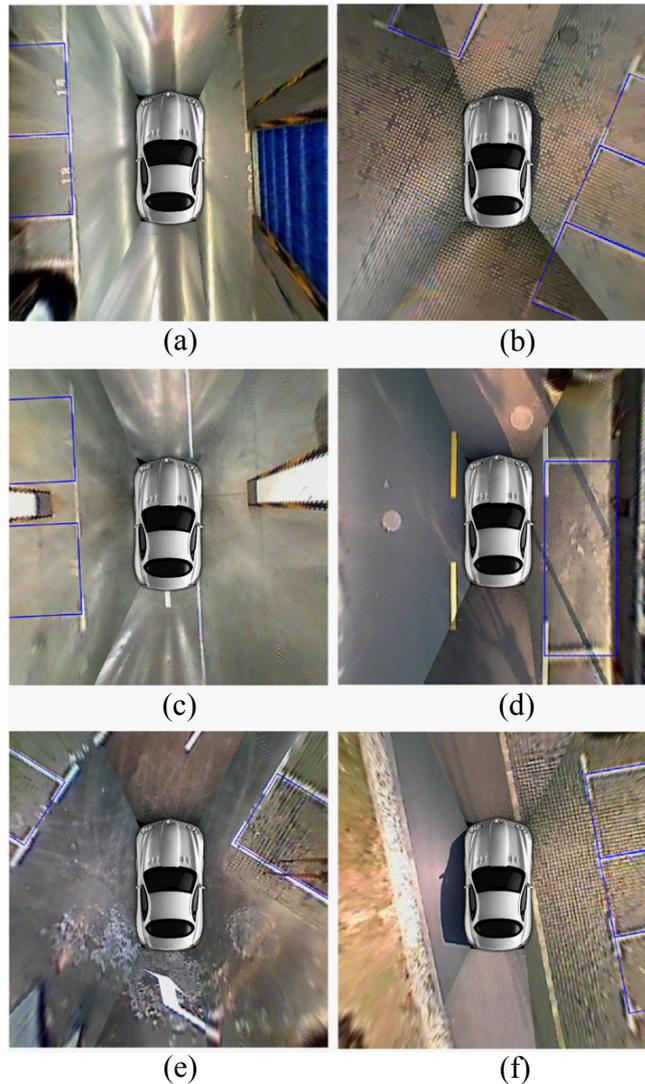 | $\overrightarrow{P_1P_2}$ is not an entrance-line candidate.
end

(a)   (b)

(c)   (d)

(e)   (f)

Figure 14. Six surround-view image samples (**a**–**f**) with marked parking-slots detected by PSD_L.

4. Experimental Results

4.1. Benchmark Dataset

In order to train and test our proposed parking-slot detection approach PSD_L, we have established a large-scale benchmark dataset, which is publicly available at https://cslinzhang.github.io/ps/. Surround-view images contained in this dataset were collected from typical indoor and outdoor parking sites using our self-developed AVM system equipped on a SAIC Roewe E50 (SAIC MOTOR, Shanghai, China) electric car [54]. Two types of parking-slots, the perpendicular ones and the parallel ones, are included. The spatial resolution of each surround-view image is 600×600, corresponding to a 9600 mm $\times$ 9600 mm square region on the physical ground. This dataset comprises two subsets, the training set and the testing set.

In the training set, we labeled 5100 surround-view images for extracting positive marking-point patterns. For each marking-point pattern, we recorded its center and its local orientation (as illustrated in Figure 9). Altogether, we have 13,364 positive marking-point pattern samples. In addition, we labeled 2400 images for extracting negative samples used for training the marking-point detector. Specifically, on each image, all the possible marking-points were marked by bounding-boxes. During training, patches sampled from these images can be used as negative samples if they do not overlap with any bounding boxes.

The testing set comprises 500 labeled surround-view images. It can be used for testing the accuracy of a marking-point detection algorithm and also can be used for testing the final accuracy of a parking-slot detection algorithm.

4.2. Evaluating the Performance of Marking-Point Detection

In our proposed parking-slot detection approach PSD_L, marking-point detection is a crucial step. In this experiment, we evaluated the performance of our marking-point detection algorithm and also compared it with several other classical methods in the field of object detection, VJ [52], HoG + SVM [55], HoG + LBP [56], PLS [57], HIKSVM [58], MultiFtr [59] and Roerei [60].

For a ground-truth marking-point g_i, if there is a detected marking point d_i satisfying $\|g_i - d_i\| < \delta$, where δ is a predefined threshold, we deem that g_i is correctly detected and d_i is a true positive. In our experiments, δ was set as 10. To compare various detectors, we plot miss rate against false positives per image (FPPI) using log-log plots by varying the threshold on detection confidence. The plots are shown in Figure 15. As recommended in [61], we use the log-average miss rate (LAMR) to summarize detector performance, computed by averaging miss rate at nine FPPI rates evenly spaced in log-space in the range 10^{-2} to 10^0. Log-average miss rates achieved by different methods are also shown in Figure 15.

From the results shown in Figure 15, it can be seen that for the task of marking-point detection, our proposed method can achieve much higher accuracy than the other widely used methods in the field of object detection. Specifically, the LAMR of our approach is 18.82% while Roerei [60] is the runner-up whose LAMR is 23.77%.
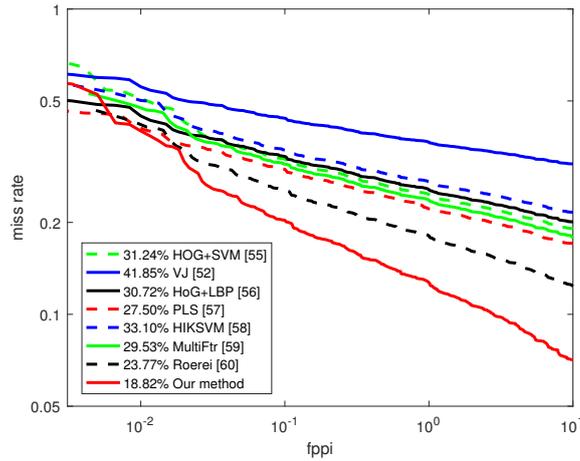
Figure 15. Marking-point detection results by using different methods.

4.3. Evaluating the Performance of Parking-Slot Detection

In this experiment, the detection accuracy of our proposed parking-slot detection algorithm PSD_L was evaluated on the test set. Besides, the performance of several state-of-the-art methods in this field, including Jung et al.'s method [34], Wang et al.'s method [36], Hamda et al.'s method [38], and Suhr&Jung's method [45], was also evaluated for comparison. We use the precision-recall rates as the performance measure, which are defined as,

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$
$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{5}$$

Each labeled parking-slot is represented as $PS_i = \left\{ P_1^i, P_2^i, o^i \right\}$, where P_1^i and P_2^i are the coordinates of the two marking-points forming the entrance-line and o^i represents the parking-slot's orientation. If o^i is 1, it means that the parking-slot PS_i is at the anticlockwise side of $\overrightarrow{P_1P_2}$; if o^i is -1, it means that PS_i is at the clockwise side of $\overrightarrow{P_1P_2}$. Suppose that $PS_d = \left\{ P_1^d, P_2^d, o^d \right\}$ is a detected parking-slot and $PS_l = \left\{ P_1^l, P_2^l, o^l \right\}$ is a labeled ground-truth parking-slot. If P_1^d matches with P_1^l, P_2^d matches with P_2^l, and o^d is equal to o^l, PS_d is regarded as a true positive; if P_1^d matches with P_2^l, P_2^d matches with P_1^l, and o^d is equal to $-o^l$, PS_d is also regarded as a true positive; otherwise, PS_d is a false positive.

In order to make the results achieved by different approaches comparable, we carefully adjusted the parameters of all the competing methods to make them achieve nearly the same high precision rates. Then, we could compare their recall rates. In this case, the approach that can achieve the highest recall rate is the best. The results are summarized in Table 1. From Table 1, it can be observed that when operating at a high precision rate, PSD_L can achieve a much higher recall rate than all the other competitors. The superiority of PSD_L over the other state-of-the-art competitors corroborates that: (1) marking-point patterns are more stable and distinguishable than primitive visual features (lines or corners); and (2) data-driven learning-based detection strategy is more robust to imaging condition variations than low-level vision algorithms.

PSD_L was implemented in C++ and tested on an in-vehicle industrial computer with a 2.4 GHZ Intel Core i5 CPU and 4G RAM. It can process 20~25 surround-view image frames per second and thus it can satisfy the requirements of most automatic PASs.

Table 1. Performance evaluation of vision-based parking-slot detection methods.

Method	Precision Rate	Recall Rate
Jung et al. [34]	98.70%	59.13%
Wang et al. [36]	98.59%	61.76%
Hamada et al. [38]	98.61%	63.82%
Suhr&Jung [45]	98.70%	76.22%
PSD_L	**98.87%**	**92.38%**

4.4. Discussion about the Usability of Our Parking-Slot Detection System

Parking-slot detection is a key component in a self-parking system. Most of the commercial parking-slot detection systems are based on ultrasonic radars. These systems share an inherent drawback that they rely on vehicles which have already been properly parked as references. In addition, they cannot detect parking-slots defined by parking line segments painted on the ground either. The vision-based technology is a useful complement to ultrasonic radar based ones. In our work, a vision-based parking-slot detection system using low-cost imaging sensors was developed. With respect to the parking-slot detection algorithm, a learning-based approach *PSD_L* was proposed. Our system has been internally tested by SAIC MOTOR and it works quite well in practice.

5. Conclusions

Vision-based parking-slot detection is still an unresolved challenging problem. In this paper, we made two contributions to this field. Firstly, we collected and labeled a large-scale surround-view image dataset and have made it publicly available to the research community. Such a dataset will for sure benefit the study of parking-slot detection. Secondly, we proposed a novel learning-based parking-slot detection approach PSD_L. Its high efficacy and efficiency have been corroborated by comprehensive experiments and it has already been deployed in practice on our experimental car. PSD_L can serve as a baseline when the other researchers develop more advanced approaches.

Author Contributions: Lin Zhang and Xiyuan Li conceived and designed the experiments; Junhao Huang performed the experiments; Ying Shen and Dongqing Wang analyzed the data; Xiyuan Li contributed reagents/materials/analysis tools; Lin Zhang wrote the paper.

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

References

1. Jo, K.; Kim, J.; Kim, D.; Jang, C.; Sunwoo, M. Development of autonomous car—Part I: Distributed system architecture and development process. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7131–7140.
2. Jo, K.; Kim, J.; Kim, D.; Jang, C.; Sunwoo, M. Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Trans. Ind. Electron.* **2016**, *62*, 5119–5132.
3. Wada, M.; Yoon, K.S.; Hashimoto, H. Development of advanced parking assistance system. *IEEE Trans. Ind. Electron.* **2003**, *50*, 4–17.
4. Suzuki, Y.; Koyamaishi, M.; Yendo, T.; Fujii, T.; Tanimoto, M. Parking assistance using multi-camera infrastructure. In Proceedings of the IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 106–111.
5. Yan, G.; Yang, W.; Rawat, D.B.; Olariu, S. SmartParking: A secure and intelligent parking system. *IEEE Intell. Transp. Syst. Mag.* **2011**, *3*, 18–30.

6. Sung, K.; Choi, J.; Kwak, D. Vehicle control system for automatic valet parking with infrastructure sensors. In Proceedings of the 2011 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 9–12 January 2011; pp. 567–568.

7. Huang, C.; Tai, Y.; Wang, S. Vacant parking space detection based on plane-based Bayesian hierarchical framework. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1598–1610.

8. De Almeida, P.; Oliveira, L.S.; Britto, A.S., Jr.; Silva, E.J., Jr.; Koerich, A.L. PKLot—A robust dataset for parking lot classification. *Exp. Syst. Appl.* **2015**, *42*, 4937–4949.

9. Pohl, J.; Sethsson, M.; Degerman, P.; Larsson, J. A semi-automated parallel parking system for passenger cars. *J. Autom. Eng.* **2006**, *220*, 53–65.

10. Satonaka, H.; Okuda, M.; Hayasaka, S.; Endo, T.; Tanaka, Y.; Yoshida, T. Development of parking space detection using an ultrasonic sensor. In Proceedings of the 13th World Congress Intelligent Transport Systems and Services, London, UK, 8–12 October 2006; pp. 1–10.

11. Park, W.J.; Kim, B.S.; Seo, D.E.; Kim, D.S.; Lee, K.H. Parking space detection using ultrasonic sensor in parking assistance system. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 1039–1044.

12. Jeong, S.H.; Choi, C.G.; Oh, J.N.; Yoon, P.J.; Kim, B.S.; Kim, M.; Lee, K.H. Low cost design of parallel parking assist system based on an ultrasonic sensor. *Int. J. Autom. Technol.* **2010**, *11*, 409–416.

13. Ford Fusion. Available online: http://www.ford.com/cars/fusion/2017/features/smart/ (accessed on 28 February 2018).

14. BMW 7 Series Sedan. Available online: http://www.bmw.com/com/en/newvehicles/7series/sedan/2015/showroom/driver_assistance.html (accessed on 28 February 2017).

15. Toyota Prius. Available online: http://www.toyota.com/prius/prius-features/) (accessed on 28 February 2017).

16. 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.

17. Zhou, J.; Navarro-Serment, L.E.; Hebert, M. Detection of parking spots using 2D range data. In Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), Anchorage, AK, USA, 16–19 September 2012; pp. 1280–1287.

18. Ibisch, A.; Stumper, S.; Altinger, H.; Neuhausen, M.; Tschentscher, M.; Schlipsing, M.; Salmen, J.; Knoll, A. Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded LIDAR sensors. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 829–834.

19. Schmid, M.R.; Ates, S.; Dickmann, J.; Hundelshausen, F.; Wuensche, H.J. Parking space detection with hierarchical dynamic occupancy grids. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 254–259.

20. Dube, R.; Hahn, M.; Schutz, M.; Dickmann, J.; Gingras, D. Detection of parked vehicles from a radar based on occupancy grid. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 1415–1420.

21. Loeffler, A.; Ronczka, J.; Fechner, T. Parking lot measurement with 24 GHz short range automotive radar. In Proceedings of the 2015 16th International Radar Symposium (IRS), Dresden, Germany, 24–26 June 2015; pp. 1–6.

22. 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.

23. 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 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 154–159.

24. Kaempchen, N.; Franke, U.; Ott, R. Stereo vision based pose estimation of parking lots using 3-D vehicle models. In Proceedings of the 2002 IEEE Intelligent Vehicle Symposium, Versailles, France, 17–21 June 2002; pp. 459–464.

25. Fintzel, K.; Bendahan, R.; Vestri, C.; Bougnoux, S.; Kakinami, T. 3D parking assitant system. In Proceedings of the 2004 IEEE Intelligent Vehicles Symposium, Parma, Italy, 14–17 June 2004; pp. 881–886.

26. 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 IEEE International Conference Intelligent Transportation System, Vienna, Austria, 13–16 September 2005; pp. 131–135.

27. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. 3D vision system for the recognition of free parking site location. *Int. J. Autom. Technol.* **2006**, *7*, 361–367.

28. 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.

29. Unger, C.; Wahl, E.; Ilic, S. Parking assistance using dense motion-stereo. *Mach. Vis. Appl.* **2004**, *25*, 561–581.

30. Xu, J.; Chen, G.; Xie, M. Vision-guided automatic parking for smart car. In Proceedings of the IV 2000 Intelligent Vehicles Symposium, Dearborn, MI, USA, 5 October 2000; pp. 725–730.

31. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. *Structure Analysis Based Parking Slot Marking Recognition for Semi-Automatic Parking System*; Joint IAPR International Workshops Structural and Syntactic Pattern Recognition (SSPR): Colorado Springs, CO, USA, 2006; pp. 384–393.

32. Jung, H.G.; Lee, Y.H.; Kim, J. Uniform user interface for semi-automatic parking slot marking recognition. *IEEE Trans. Veh. Technol.* **2010**, *59*, 616–626.

33. Du, X.; Tan, K. Autonomous reverse parking system based on robust path generation and improved sliding mode control. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1225–1237.

34. Jung, H.G.; Kim, D.S.; Yoon, P.J.; Kim, J. Parking slot markings recognition for automatic parking assist system. In Proceedings of the 2006 IEEE Intelligent Vehicles Symposium, Tokyo, Japan, 13–15 June 2006; pp. 106–113.

35. Sonka, M.; Hlavac, V.; Boyle, R. *Image Processing, Analysis, and Machine Vision*; Thomson-Engineering: Mobile, AL, USA, 2008.

36. 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*, 847406.

37. Deans, S.R. *The Radon Transform and Some of Its Applications*; Dover Publications: Mineola, NY, USA, 1983.

38. Hamada, K.; Hu, Z.; Fan, M.; Chen, H. Surround view based parking lot detection and tracking. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 1106–1111.

39. Matas, J.; Galambos, C.; Kittler, J. Robust detection of lines using the progressive probabilistic Hough transform. *Comput. Vis. Image Underst.* **2000**, *78*, 119–137.

40. 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.

41. Borgefors, G. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 849–856.

42. Lee, S.; Seo, S. Available parking slot recognition based on slot context analysis. *IET Intell. Transp. Syst.* **2016**, *10*, 594–604.

43. Nieto, M.; Salgado, L. Robust multiple lane road modeling based on perspective analysis. In Proceedings of the ICIP 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 2396–2399.

44. Barber, D. *Bayesian Reasoning and Machine Learning*; Cambridge University Press: Cambridge, UK, 2013.

45. Suhr, J.K.; Jung, H.G. Full-automatic recognition of various parking slot markings using a hierarchical tree structure. *Opt. Eng.* **2013**, *52*, 037203.

46. Suhr, J.K.; Jung, H.G. Sensor fusion-based vacant parking slot detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 21–36.

47. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Fourth Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.

48. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334.

49. OpenCV 3.1. Available online: http://opencv.org/opencv-3-1.html (accessed on 28 February 2017).

50. Dollar, P.; Tu, Z.; Perona, P.; Belongie, S. Integral channel features. In Proceedings of the British Machine Vision Conference, London, UK, 7–10 September 2009; pp. 91:1–91:11.

51. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139.

52. Viola, P.A.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154.

53. Dollar, P.; Appel, R.; Kienzle, W. Crosstalk cascades for frame-rate pedestrain detection. In Proceedings of the 12th European conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 645–659.

54. Roewe E50 Review. Available online: http://www.autocar.co.uk/car-review/roewe/e50 (accessed on 28 February 2017).

55. Dalal, N.; Triggs, B. Histogram of oriented gradients for human detection. In Proceedings of the CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.

56. Wang, X.; Han, X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 32–39.

57. Schwarts, W.; Kembhavi, A.; Harwood, D.; Davis, L. Human detection using partial least squares analysis. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 24–31.

58. Maji, S.; Berg, A.; Malik, J. Classification using intersection kernel support vector machines is efficient. In Proceedings of the CVPR 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.

59. Wojek, C.; Schiele, B. A performance evaluation of single and multi-feature people detection. In Proceedings of the 30th DAGM symposium on Pattern Recognition, Munich, Germany, 10–13 June 2008; pp. 82–91.

60. Benenson, R.; Mathias, M.; Tuytelaars, T.; van Gool, L. Seeking the strongest rigid detector. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 3666–3673.

61. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 743–761.