VISION-BASED PARKING-SLOT DETECTION: A BENCHMARK AND A LEARNING-BASED APPROACH

Linshen Li[1], *Lin Zhang*[1,2*], *Xiyuan Li*[1], *Xiao Liu*[1], *Ying Shen*[1], *Lu Xiong*[2]

[1]School of Software Engineering, Tongji University, Shanghai, China
[2]Collaborative Innovation Center of Intelligent New Energy Vehicle, Tongji University, Shanghai, China

ABSTRACT

Recent years have witnessed a growing interest in developing automatic parking systems in the field of intelligent vehicle. However, how to effectively and efficiently locating parking-slots using a vision-based system is still an unresolved issue. In this paper, we attempt to fill this research gap 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. For each image in this database, the marking-points and parking-slots are carefully labelled. Such a database can serve as a benchmark to design and validate parking-slot detection algorithms. Secondly, a learning based parking-slot detection approach is proposed. With this approach, given a test image, the marking-points will be detected at first and then the valid parking-slots can be inferred. Its efficacy and efficiency have been corroborated on our database. The labeled database and the source codes are publicly available at http://sse.tongji.edu.cn/linzhang/ps/index.htm.

Index Terms— Parking assistance systems, parking-slot detection, AdaBoost, decision tree

1. INTRODUCTION

An automatic parking system starts by target position designation. To resolve this problem, various solutions have been proposed 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 [1]. 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 parks 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. 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 adjacent vehicles. This is the most widely used approach as it can be implemented using various range-finding sensors, such as ultrasonic sensors [2, 3], laser scanners [4], short-range radars [5, 6], etc. However, the free-space-based approach has an inherent drawback in that it cannot find free spaces when there is no adjacent vehicle 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 and thus its performance does not depend on the existence or poses of adjacent vehicles. Moreover, slot-markings can provide more accurate parking information than "free-space". 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.

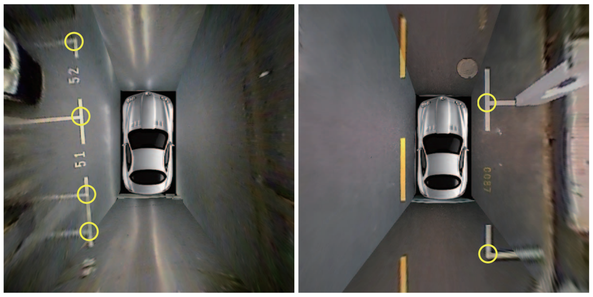
1.1. Related work

In this paper, we focus on how to detect parking-slots from a surround-view image. Representative work in this area will be reviewed here.

The research in this area started from Xu *et al.*'s pioneer work [7]. In [7], 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-work 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 [8], 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.* [9] extended it to a two-touch method that can recognize various types of parking-slots based on two manually designated points. The apparent drawback of the methods in [8] and [9] is that they are not fully automatic. Fully automatic parking-slot detection methods

[*] Corresponding author. Email: cslinzhang@tongji.edu.cn

978-1-5090-6067-2/17/$31.00 ©2017 IEEE



(a) (b)

Fig. 1. (a) and (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.

are developed along two main streams, line-based ones and corner-based ones, based on the primitive visual features they extract. Hough transform, Radon transform, and RANSAC (RANdom SAmpling Consensus) are commonly used techniques to detect parking-lines [10, 11, 12] while the Harris corner detector is usually adopted by corner-based approaches to detect corners on a surround-view image [13, 14]. When the primitive features are ready, semantic parking-slots can be inferred from them.

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, nearly all the existing state-of-the-art methods in this field are based on low-level visual features, such as lines and corners, detected by some low-level vision algorithms. These features actually are not quite distinguishable and even worse, they are unstable and unrepeatable to changes in environment aroused by noise, clutter, or illumination variation. Hence, how to efficiently and accurately detect parking-slots using vision-based methods in uncontrolled environment is still a great challenge. Secondly, vision-based parking-slot detection is actually a "pattern classification" problem. To cope with such a problem, a publicly-available large-scale benchmark dataset is desired, which in fact is indispensable for researchers to design and compare the detection algorithms. Unfortunately, such a dataset is still lacking in this area.

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

(1) A data-driven learning-based approach, PSD_L (*P*arking-*S*lot *D*etection based on *L*earning), is proposed for parking-slot detection. Given a surround-view image, PSD_L detects the marking-points using a pre-trained detector at first and then infers the valid parking-slots from them. Marking-points are defined as the cross-points of parking-lines. Examples of marking-points are shown in Fig. 1. The advantage 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 the 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.

(2) To facilitate the study of parking-slot detection, we have established a large-scale benchmark dataset and will make it publicly available. This dataset comprises 8600 surround-view images collected from typical parking sites with our experimental car and all the images are manually labeled with care. Such a dataset can be employed for training and testing new parking-slot detection algorithms. Please refer to Sect. 4 for more details about this dataset.

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 for parking-slot detection. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2. SURROUND-VIEW GENERATION

The parking-slot detection algorithm is applied on the surround-view image. Thus, in this section, we briefly introduce the steps for surround-view generation.

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.

Actually, the surround-view is the composite view of the four bird's-eye-views. The key step for generating the bird's-eye-view image is to build an off-line lookup table $T_{B \rightarrow F}$, mapping a point $\mathbf{x}_B$ on the bird's-eye-view image to a position $\mathbf{x}_F$ on the input fish-eye image. To determine $T_{B \rightarrow F}$, we need to determine $P_{B \rightarrow W}$ the transformation matrix from the bird's-eye-view coordinate system to the world coordinate system, $P_{W \rightarrow U}$ the transformation matrix from the world coordinate system to the undistorted input image coordinate system, and $T_{U \rightarrow F}$ the lookup table mapping a point on the undistorted input image to a position on the original fish-eye image. $P_{B \rightarrow W}$ is a similarity transformation matrix, which can be determined straightforwardly if the size of the bird's-eye-view image and the corresponding physical visible range are determined beforehand. $P_{W \rightarrow U}$ is a homography matrix which can be determined via a calibration field [10]. The distortion coefficients of the fish-eye camera can be estimated by

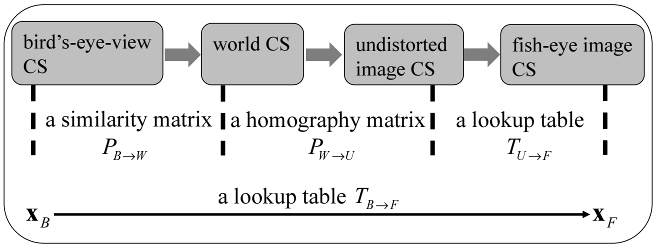

bird's-eye-view CS
world CS
undistorted image CS
fish-eye image CS
a similarity matrix $P_{B \to W}$
a homography matrix $P_{W \to U}$
a lookup table $T_{U \to F}$
a lookup table $T_{B \to F}$
$\mathbf{x}_B$
$\mathbf{x}_F$

Fig. 2. The relations among the coordinate systems involved in surround-view generation.

a standard calibration process [15] and accordingly the mapping table $T_{U \to F}$ can be obtained. When the matrices $P_{B \to W}$ and $P_{W \to U}$, and the mapping table $T_{U \to F}$ are ready, the mapping table $T_{B \to F}$ can be finally determined.

Fig. 2 illustrates the relationships among the coordinate systems involved in bird's-eye-view generation. In Fig. 3 we show an example for surround-view generation. Figs. 3(a)∼3(d) are four fish-eye images and Fig. 3(e) is the surround-view image synthesized from Figs. 3(a)∼3(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 Fig. 4. PSD_L actually comprises two components, marking-points detection and parking-slot inference.

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 Fig. 4. 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 Sect. 4 for details), the positive training samples can be simply extracted from labeled surround-view images while the negative training samples are extracted using a bootstrapping process during training.

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, including the normalized intensity, the gradient magnitude, and the oriented gradient histograms [16]. For classifier, we adopt the popular AdaBoost framework. A boosted classifier *H* consisting of *M* weak classifiers can be represented as,

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

where each h_t is a weak classifier and α_t is its associated weight. $\mathbf{x}$ is classified as positive if $H(\mathbf{x}) > 0$ and $H(\mathbf{x})$

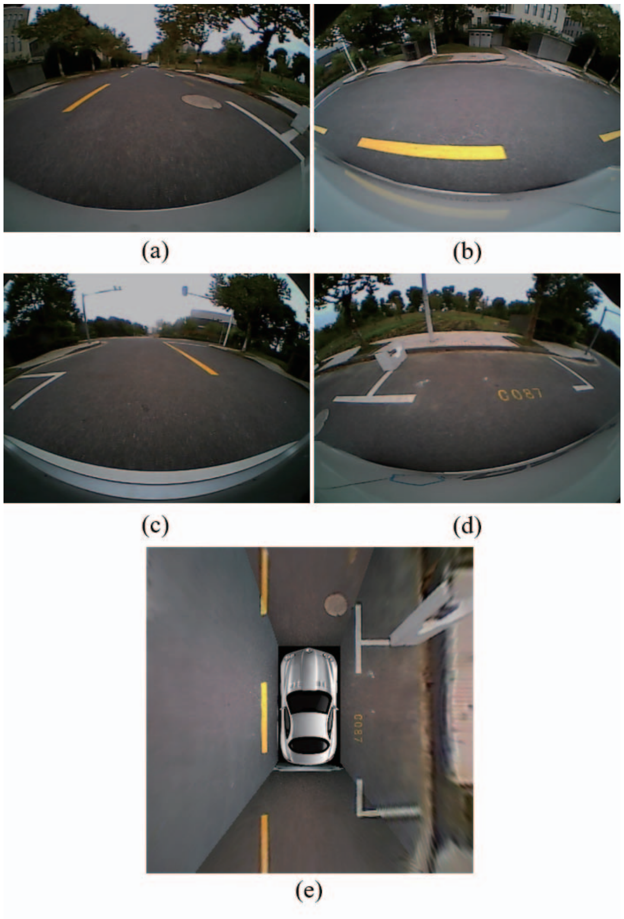


Fig. 3. Images (a)∼(d) are captured from the front, the left, the back, and the right fish-eye cameras, respectively. (e) is the surround-view image synthesized from (a)∼(d).

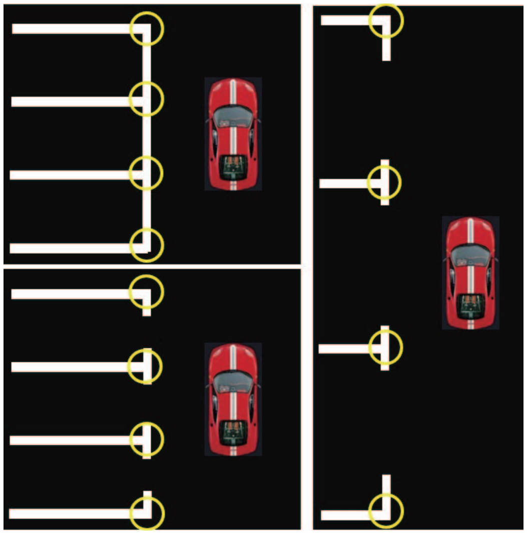


Fig. 4. Typical types of parking-slots that the proposed algorithm PSD_L can detect.

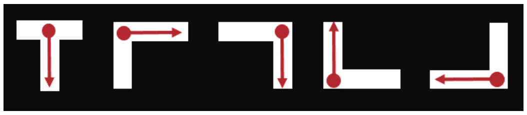


Fig. 5. Directions of the marking-point patterns.

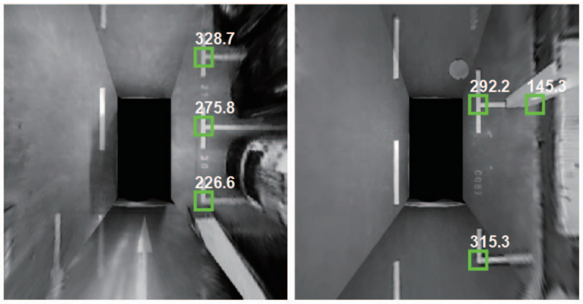


Fig. 6. Marking-point detection results on two typical surround-view images.

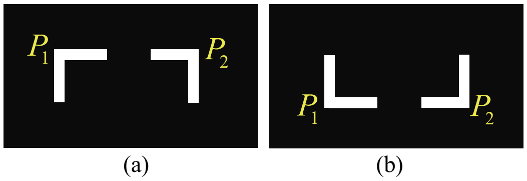
serves as the confidence. In terms of the weak classifier, we use the shallow decision tree. The training is conducted in several stages and after each stage a bootstrapping process is performed to extract negative samples for the next stage.

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 [17]. To simplify training, we use the "constant soft-cascade" strategy [18] 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,

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

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 $\mathbf{t}$ reaches a node whose associated weighted log-ratio is smaller than a pre-defined constant threshold θ, the testing stops.

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 4 detectors $\{det_j\}_{j=1}^4$ and det_j is responsible for detecting marking-point patterns whose directions are within the range $\left[-\frac{3\pi}{4}+\frac{\pi}{2}j, -\frac{\pi}{4}+\frac{\pi}{2}j\right]$. For training 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 Fig. 5. When positive image patches are extracted, they are at 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* positive samples are labeled, there will be *KN* positive samples for training det_j.



(a) (b)

Fig. 7. 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.

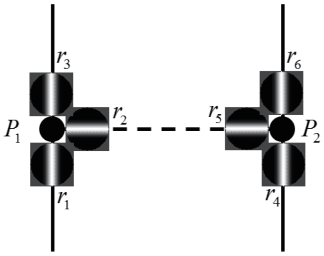


Fig. 8. Six Gaussian line templates are used to examine the local image patterns around the marking-points P_1 and P_2

In Fig. 6, marking-point detection results on two typical surround-view images are shown.

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 $\overrightarrow{P_1P_2}$ can be a valid entrance-line of a parking-slot. At first, the length of $\overrightarrow{P_1P_2}$ should satisfy some length constraints obtained from the prior knowledge. Then,

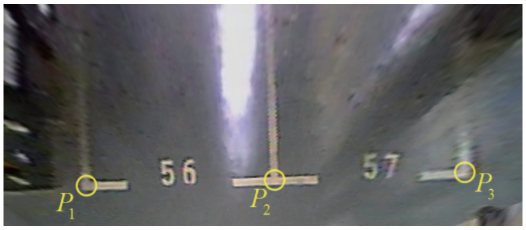


Fig. 9. P_1P_2 and P_2P_3 are two entrance-line candidates for two perpendicular parking-slots while P_1P_3 is an entrance-line candidate for a parallel parking slot. Actually, P_1P_3 is invalid and should be removed.

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 Fig. 4, 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 shown in Fig. 7. Inspired by this analysis, we propose to use Gaussian line templates to check the image patterns around P_1 and P_2. 6 Gaussian line templates are used and their positions and orientations relative to P_1 and P_2 are illustrated in Fig. 8. By checking the 6 responses $r_1 \sim r_6$ with a set of simple rules, whether $\overrightarrow{P_1P_2}$ 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 $\overrightarrow{P_1P_2}$ can also be determined. The checking-rules are summarized in Table 1.

Table 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 $r_5 > t$
    $\overrightarrow{P_1P_2}$ is an entrance-line candidate;
    The parking-slot is on the clockwise side of $\overrightarrow{P_1P_2}$;
elseif $r_3 >> r_1$ 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 $r_5 > t$
    $\overrightarrow{P_1P_2}$ is an entrance-line candidate;
    The parking-slot is on the anti-clockwise side of $\overrightarrow{P_1P_2}$;
else
    $\overrightarrow{P_1P_2}$ is not an entrance-line candidate.
```

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 Fig. 9. In Fig. 9, $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ are two entrance-line candidates for two perpendicular parking-slots while $\overrightarrow{P_1P_3}$ is an entrance-line candidate for a parallel parking slot. Actually, $\overrightarrow{P_1P_3}$ is an invalid entrance line. So, we postprocess the entrance-line candidate set by removing the ones with marking-points on them to annihilate contradictions.

Finally, valid entrance-lines are remained, each of which represents a valid parking-slot. Their information is then sent to the path-planning module.

4. EXPERIMENTAL RESULTS

4.1. Benchmark dataset

We have established a large-scale benchmark dataset, in which surround-view images were collected from typical underground and outdoor parking sites. This dataset comprises two subsets, the training set and the testing set. In training set, we labeled 5100 images for extracting positive patterns and 2400 images for extracting negative samples. For each marking-point pattern, we marked its center and its local orientation (as illustrated in Fig. 5). Altogether, we have 13,364 positive samples. Testing set has two parts, Part1 and Part2. Part1 contains 600 labeled images and is used for testing the accuracy of a marking-point detection algorithm. Part2 contains 500 labeled images and is used for testing the final accuracy of a parking-slot detection algorithm.

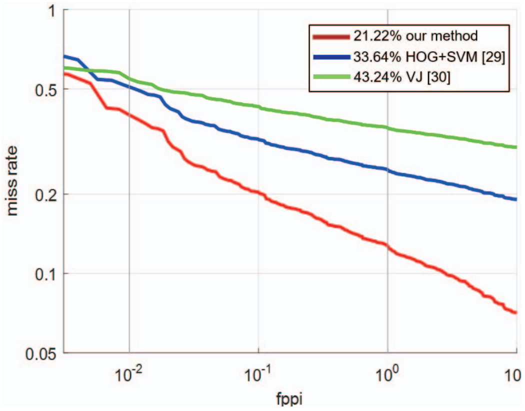


Fig. 10. Marking-point detection results by using different methods.

4.2. Evaluation the performance of marking-point detection

In this experiment, we evaluated the performance of our marking-point detection algorithm and also compared it with the other two classical methods in the field of object detection, HoG+SVM [19] and VJ [17]. Part1 of the test set was used in this experiment.

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 Fig. 10. As recommended in [20], 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 Fig. 10.

From the results shown in Fig. 10, it can be seen that for the task of marking-point detection, our proposed method can achieve much higher accuracy than the other two widely used methods in the field of object detection. Specifically, the LAMR of our approach is 21.22% while the LAMRs of HOG+SVM and VJ are 33.64% and 43.24%, respectively.

4.3. Evaluation the performance of parking-slot detection

In this experiment, the detection accuracy of our proposed parking-slot detection algorithm PSD_L was evaluated on Part2 of the test set. We use the precision-recall rates as the performance measure, which are defined as,

$$\begin{aligned} precision &= \frac{true\ positives}{true\ positives + false\ positives} \\ recall &= \frac{true\ positives}{true\ positives + false\ negatives} \end{aligned} \qquad (3)$$

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 clockwise side of $\overrightarrow{P_1P_2}$; if o^i is -1, it means that PS_i is at the anti-clockwise side of $\overrightarrow{P_1P_2}$. In Part2 test set, there are 761 labeled parking-slots. 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 and P_2^d match with P_1^l and P_2^l within a threshold and o^d is equal to o^l, we regard PS_d as a true positive.

With our experimental settings, the number of true positives is 703, the number of false negatives is 58, and the number of false positives is 8. Accordingly, the precision rate is equal to 98.87% and the recall rate is equal to 92.38%. PSD_L was implemented in C++ and tested on an 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.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we made two contributions to the field of vision-based parking-slot detection. Firstly, we collected and labeled a large-scale surround-view dataset and have made it publicly available. 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 experiments and it has already been deployed in practice on our experimental car. In near future, we may resort to deep neural networks to improve the performance of our detector.

6. ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of China under grant no. 61672380 and in part by the ZTE Industry-Academia-Research Cooperation Funds under grant no.CON1608310007.

7. REFERENCES

[1] C. Huang, Y. Tai, and S. Wang, "Vacant parking space detection based on plane-based Bayesian hierarchical framework," *IEEE Trans. CSVT*, vol. 23, no. 9, pp. 1598–1610, 2013.

[2] W.J. Park, B.S. Kim, D.E. Seo, D.S. Kim, and K.H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *IEEE Intell. Veh. Symp.*, 2008, pp. 1039–1044.

[3] S.H. Jeong, C.G. Choi, J.N. Oh, P.J. Yoon, B.S. Kim, M. Kim, and K.H. Lee, "Low cost design of parallel parking assist system based on an ultrasonic sensor," *Int. J. Autom. Technol.*, vol. 11, no. 3, pp. 409–416, 2010.

[4] J. Zhou, L.E. Navarro-Serment, and M. Hebert, "Detection of parking spots using 2D range data," in *IEEE Int. Conf. Intell. Transp. Syst.*, 2012, pp. 1280–1287.

[5] M.R. Schmid, S. Ates, J. Dickmann, F. Hundelshausen, and H.J. Wuensche, "Parking space detection with hierarchical dynamic occupancy grids," in *IEEE Intell. Veh. Symp.*, 2011, pp. 254–259.

[6] R. Dube, M. Hahn, M. Schutz, J. Dickmann, and D. Gingras, "Detection of parked vehicles from a radar based on occupancy grid," in *IEEE Intell. Veh. Symp.*, 2014, pp. 1415–1420.

[7] J. Xu, G. Chen, and M. Xie, "Vision-guided automatic parking for smart car," in *IEEE Intell. Veh. Symp.*, 2000, pp. 725–730.

[8] H.G. Jung, D.S. Kim, P.J. Yoon, and J. Kim, "Structure analysis based parking slot marking recognition for semi-automatic parking system," in *IAPR Int. Workshop Struct. Syntact. Patt. Recog.*, 2006, pp. 384–393.

[9] H.G. Jung, Y.H. Lee, and J. Kim, "Uniform user interface for semi-automatic parking slot marking recognition," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 616–626, 2010.

[10] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic parking based on a bird's eye view vision system," *Adv. Mech. Eng.*, vol. 2014, pp. 847406:1–13, 2014.

[11] X. Du and K. Tan, "Autonomous reverse parking system based on robust path generation and improved sliding mode control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1225–1237, 2015.

[12] J.K. Suhr and H.G. Jung, "Automatic parking space detection and tracking for underground and indoor environments," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5687–5698, 2016.

[13] J.K. Suhr and H.G. Jung, "Full-automatic recognition of various parking slot markings using a hierarchical tree structure," *Opt. Eng.*, vol. 52, no. 3, pp. 037203:1–14, 2013.

[14] J.K. Suhr and H.G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 21–36, 2014.

[15] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. PAMI*, vol. 22, no. 11, pp. 1330–1334, 2000.

[16] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*, 2009, pp. 91:1–11.

[17] P.A. Viola and M.J. Jones, "Robust real-time face detection," *Int'l J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[18] P. Dollar, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrain detection," in *ECCV*, 2012, pp. 645–659.

[19] N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.

[20] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. PAMI*, vol. 34, no. 4, pp. 743–761, 2012.