The Visual Computer (2021) 37:2657–2669
https://doi.org/10.1007/s00371-021-02199-y

ORIGINAL ARTICLE

GPSD: generative parking spot detection using multi-clue recovery model

Zhihua Chen[1] · Jun Qiu[1] · Bin Sheng[2] · Ping Li[3] · Enhua Wu[4,5]

Accepted: 5 June 2021 / Published online: 19 June 2021

Abstract

Due to various complex environmental factors and parking scenes, there are more stringent requirements for automatic parking than the manual one. The existing auto-parking technology is based on space or plane dimension, where the former usually ignores the ground parking spot lines which may cause parking at a wrong position, while the latter often costs a lot of time in object classification which may decreases the algorithm applicability. In this paper, we propose a Generative Parking Spot Detection algorithm which uses a multi-clue recovery model to reconstruct parking spots. In the proposed method, we firstly dismantle the parking spot geometrically for marking the location of its corresponding corners and then use a micro-target recognition network to find corners from the ground image taken by car cameras. After these, we use the multi-clue model to correct the fully pairing map so that the reliable true parking spot can be recovered correctly. The proposed algorithm is compared with several existing algorithms, and the experimental result shows that it has a higher accuracy than others which can reach more than 80% in most test cases.

Keywords Auto-parking · Parking spot detection · Multi-clue recovery model · Corner recognition

1 Introduction

According to the equipment type, the existing automatic parking methods can be roughly divided into two categories: the radar-based one and the camera-based one. The most widely used radar-based method includes: laser radars [13,21,29], ultrasonic radars [20,22], and short-wave radars. Though these methods are good at detecting vehicles and obstacles, or planning and tracking routes, they can neither judge the detected object type well nor obtain the information for parking guidance. Contrarily, the camera-based method can grasp the ground guidance information better and have a lower requirement for hardware and the image quality. Now, the most popular four car cameras are optical flow cameras, depth cameras, stereo cameras [26], and fish-eye cameras [2]. However, these approaches also just pay attention to the space calculation, without fully utilizing the ground information. Thus, Zhang et al. [8,30] proposed a parking spot detection algorithm using the vertex angle as detection object for determining a parking spot, which takes much time to the vertex angle classification and is not enough universal.

In this paper, we propose a Generative Parking Spot Detection (GPSD) algorithm using a multi-clue recovery model. First, we design an illumination balance algorithm for ensur-

✉ Bin Sheng
shengbin@sjtu.edu.cn

✉ Enhua Wu
ehwu@um.edu.mo

Zhihua Chen
czh@ecust.edu.cn

Jun Qiu
2267638193@qq.com

Ping Li
p.li@polyu.edu.hk

[1] Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China

[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

[3] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

[4] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

[5] Faculty of Science and Technology, University of Macau, Macau, China

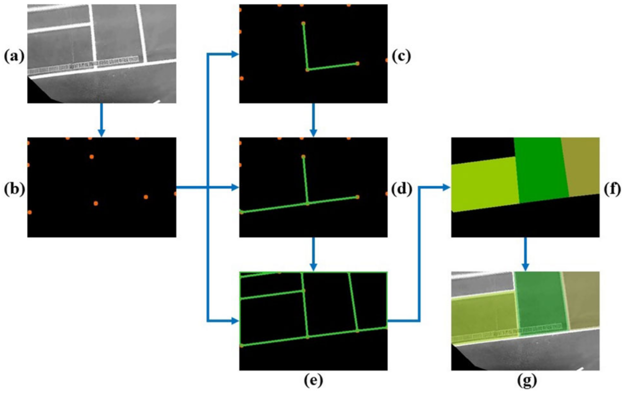


Fig. 1 A schematic diagram of the core content. **a** is a processed ground image; **b** is the result of corner detection; **c**–**f** are, respectively, the generative result by sideline clues, occlusion clues, edge clues, and domain clues; **g** is the final result with detected parking spots

ing detection accuracy, which splits the original image into multiple areas and specifies the balance strategy according to the illumination difference of each area. Then, we propose a micro-target detection algorithm, which strengthens the weight of underlying semantics and expands the spatial pyramid pooling layer. Finally, we propose the multi-clue model using the detected corner to recover the parking spot and adding several clues to correct the recovery result. The result of each stage about proposed algorithm is shown in Fig. 1.

Our contributions can be summarized as followings:

- *Illumination balance* We gradually split the original image with complex scene information into multiple layers, in order to enhance the locally blurred target and ensure the global illumination continuity more conveniently.
- *Micro-target detection* We adjust the network by strengthening the weight of underlying semantics and expanding the spatial pyramid pooling layer to enhance its detection ability.
- *Parking spot location* We geometrically dismantle the object within the original image into several meta-elements as the input of detection process, and for eliminating the interference of the complex scene information, we use the multi-clue model to recover the parking spot, as well as correct the final result.

2 Related works

Automatic Parking Assistance System (APAS) The early APAS tries to find a empty space for parking and guide routes by sensors or cameras. Song et al. [21] proposed a laser-based Simultaneous Localization and Mapping (SLAM) automatic parallel parking and tracking control scheme. Scheunert et al. [18] used a photonic mixer-type depth camera to collect the spatial parking lot information. Suhr et al. [25] designed a three-dimensional point cloud reconstruction based on motion stereo. However, these methods ignore the ground parking spot line and may cause wrong parking. With the deep learning rapidly growing, the new APAS starts using ground images for detecting parking spots, which are captured by four-way fish-eye cameras and finally transformed to an around view images [9,10,32]. In addition, Yamamoto et al. [28] proposed a parking control system by only a monocular camera, while Athira et al. [1] presented an image processing based on Optical Character Recognition (OCR). Besides, the Internet of Thing (IoT) has also been used for smart parking lot managements [5] and real-time information exchanges [6].

Parking spot detection (PSD) The existing PSD method mainly detects the vertex angle [8,23,24,27,30] or the sideline [7,19]. Zhang et al. [30] proposed a spot detection algorithm based on deep convolutional neural network and built a large-scale labeled dataset. Inspired by their works, Suhr and Jung [24] proposed an end-to-end trainable one-stage parking slot detection method, and Wu et al. [27] annotated and released the large-scale benchmark dataset PSDD. Sedighi and Kuhnert [18] presented a parking strategy for vision-based autonomous parking systems in which the ego-vehicle could complete its auto-park by one maneuver, or up to maximum three required maneuvers. Although the existing method can catch the ground information wall, it takes much time in classification tasks about vertex angles or sidelines, and taking different recovery strategies in different situations improves the algorithm complexity. Based on the existing car camera system and inspired by the YOLO [3,14–16], we propose a parking spot detection algorithm using corners for the detection target.

3 Overview

Different from existing algorithms using sidelines [7,19] or vertex angles [8,30] as the describable characteristic of a parking spot, we focus on the corner, which is more basic but the key constituent element of parking spots, and propose a generative parking spot detection algorithm based on multi-clue recovery model. As shown in Fig. 2, the proposed algorithm contains three main modules: an illumination balance module for image preprocessing, a micro-target detection module for corner detection, and a generative parking spot location module for parking spot recovery. For each captured ground image, it will be firstly transformed to a single-channel picture in a dimension reduction module, then be filtered, and finally be separated into several areas where the view area is transported into the balancer

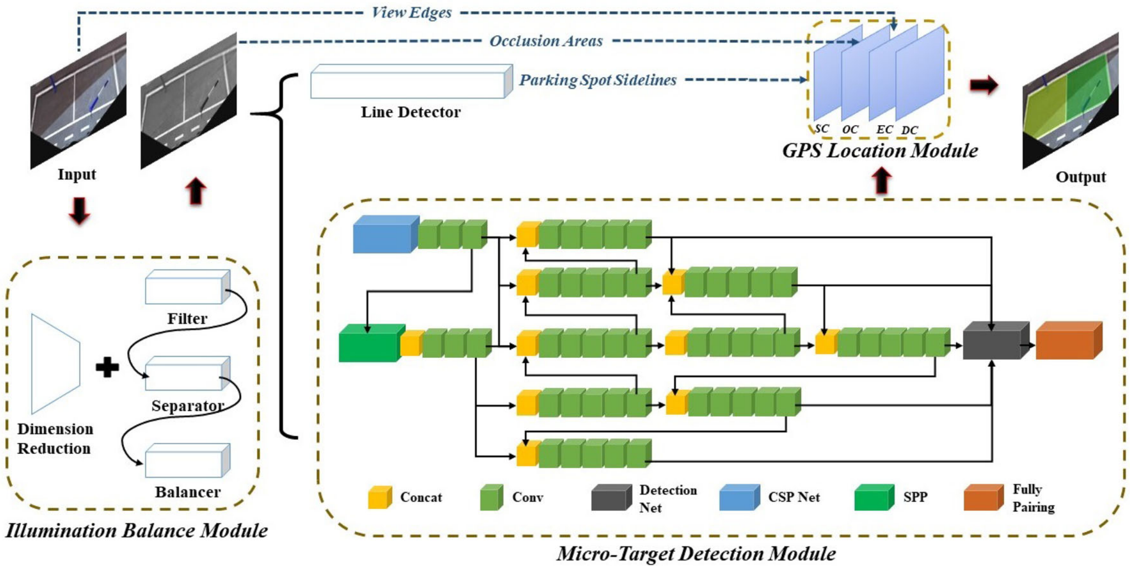


Fig. 2 A schematic diagram of GPSD. The proposed algorithm consists of three main modules: an Illumination Balance Module, a Micro-Target Detection Module, and a Generative Parking Spot (GPS) Location Module. During the illumination balance processing, the original image will be transported to a single-channel one for weakening the influence of the color dimension and then be separated to several areas where the occlusion area will be regarded as Occlusion Clues (OCs) in GPS Location Module. Both Line Detector and Micro-Target Detection Module will take the balanced image as input, and the sideline detected in the former will be regarded as Sideline Clues (SCs) in GPS Location Module; while the result of the latter will be used to construct a fully pairing map. Finally, several clues, like SCs, OCs, Edge Clues (ECs), and Domain Clues (DCs), will be applied to correct the fully pairing map, as well as identify the real parking spot in GPS Location Module

module for completing illumination balance. After these, the preprocessed image will be transported into the micro-target detection module and a line detector. In the former, we take a designed CNN to detect the position of corners and use them to construct a fully pairing map. In the generative parking spot location module, we design a multi-clue recovery model which takes various clues, like sideline clues, occlusion clues, edge clues, and domain clues, to correct the pairing result for locating the real parking spot.

4 Layered analytical illumination balance

In this section, in order to solve the problem of the unbalanced illumination area, partial information missing, and various definitions of parking spot lines, we propose a Layer Analytical Illumination Balance (LAIB) method for image preprocessing.

4.1 Layered analytical model

Color dimension reduction The multiple color type of parking spot lines may cause some unnecessary interference while detecting. Since there is always an obvious color difference between lines and the ground, it is easy to remove the color feature by reducing the color dimension. In our experiment, we just transform the RGB image to a gray one.

Area separation As shown in Fig. 3, we firstly design a Common Area (CA) Extractor to catch the occlusion from several reference images. Because the occlusion is caused by a same camera, it is consistent in all ground images, and we select the reference image following the random principle. Then, we used the extracted occlusion to separate the view from the target image in a View-Occlusion (VO) Separator, which mainly contains the parking spot line and the ground. Finally, we further separate the view into the line area with lights and shadows, and the ground area with lights and shadows in a Line-Ground (LG) Separator.

4.2 Illumination balance strategy

Filter design We improve the original Gaussian filter function by increasing the pixel utilization for reducing the influence of image sharpness dropping:

$$F(p_{x,y},\delta)=\lfloor f(p_{x,y},k,\delta)+\Delta f\rfloor, \qquad (1)$$

where $F(p_{x,y},\delta)$ is the improved Gaussian filter function, consisting of the body $F(p_{x,y},\delta)$ and a relevance item Δf for increasing the correlation of surrounding pixels. In $F(p_{x,y},\delta)$, $p_{x,y}$ represents a pixel at location (x, y) in ground images; δ is a retracting coefficient for adjusting the

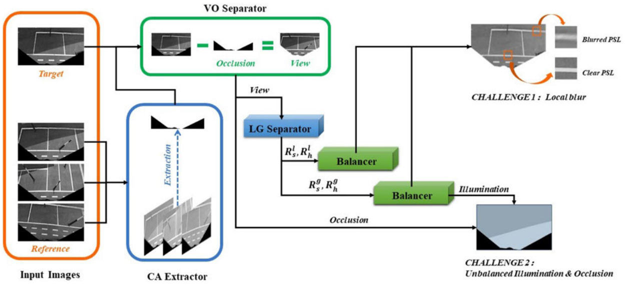


Fig. 3 A frame analysis diagram of Layered Analytical Illumination Balance (LAIB). In this figure, several ground images are randomly selected from the original dataset as the reference images, and they are used to extract the common area in the Common Area (CA) Extractor. The extracted common area is the occlusion area caused by the view of a car camera itself and will be regarded as the reference to separate the occlusion from the view for each target image in the View-Occlusion (VO) Separator. The separated view area is transported into the Line-Ground (LG) Separator to catch the independent parking spot line and ground, which are used to calculate the illumination value, separately

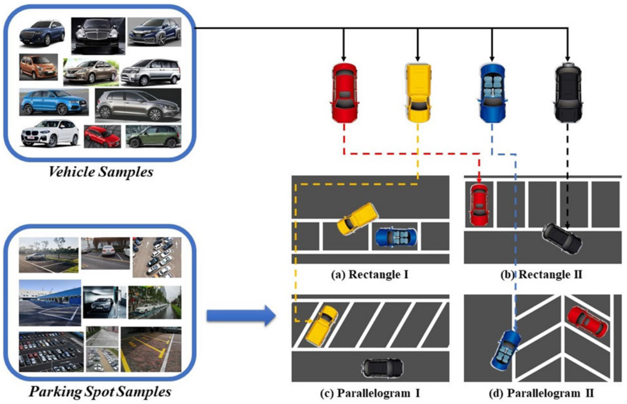


Fig. 4 A design schematic diagram of parking spot types. Four categories are selected from numerous vehicle samples and adjusted to the top view. At the same time, four common PS categories are summarized from various parking spot examples. From the top-down perspective, all vehicles can be regarded as several approximate rectangles, and this feature has affected the design of parking spot types

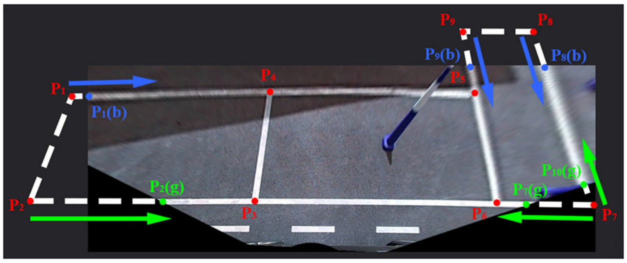


Fig. 5 A schematic diagram of corner classification. In this figure, the reliable corner, $P_1 \sim P_9$, is marked in red, the occlusion hypothetical corners, $P_2(g)$, $P_7(g)$, $P_{10}(g)$, is marked in green, and the edge hypothetical corners, $P_1(b)$, $P_8(b)$, $P_9(b)$, is marked in blue

conversion ratio of pixel values and $\delta \in [0.5, 1.0]$:

$$f(p_{x,y}, k, \delta) = \frac{\sum\sum \frac{p_{i,j}}{|k^2 \cdot \log(\frac{p_{i,j}}{255}+\delta) - \sum\sum \log(\frac{p_{i,j}}{255}+\delta)|}}{\sum\sum \frac{1}{|k^2 \cdot \log(\frac{p_{i,j}}{255}+\delta) - \sum\sum \log(\frac{p_{i,j}}{255}+\delta)|}}, \tag{2}$$

where k is the Gaussian kernel size, satisfying $k > 1$. For each pixel $p_{i,j}$ at location (i, j), it satisfies $1 - k \leq 2 \cdot (i - x), 2 \cdot (j - y) \leq k - 1$. And Δf is as following:

$$\Delta f = \begin{cases} \frac{\sum\{f - p' | p' < f\}}{\sum 1}, \frac{\sum\{p' - f | p' > f\}}{\sum\{f - p' | p' < f\}} \leq 0.2 \\ \frac{\sum\{p' - f | p' > f\}}{\sum 1}, \frac{\sum\{f - p' | p' < f\}}{\sum\{p' - f | p' > f\}} \leq 0.2 \\ \varepsilon_- \cdot \frac{\sum f - p'}{\sum 1} + \varepsilon_+ \cdot \frac{\sum p' - f}{\sum 1}, \text{else} \end{cases} , \tag{3}$$

where ε_+ and ε_- are both proportional coefficients, satisfying $\varepsilon_+ + \varepsilon_- = 1$. And for simplicity, $f(p_{x,y}, k, \delta)$ is simplified to f. p' represents the surrounding pixel, defined as following:

$$p' \in \left\{\log\left(\frac{p_{x+i,y+j}}{255} + \delta\right) | i, j \neq 0\right\}. \tag{4}$$

Balance strategy After the filter processing, the ground image will be separated to get the view which will be further divided into four parts: the parking spot line within shadow areas R_s^l and light areas R_h^l, and the ground within shadow areas R_s^g and light areas R_h^g, according to several thresholds G_1, G_0 and G_2, sequentially:

$$G_0(V, N, \Gamma_0) = \sum\left(\frac{v_i \cdot \log\frac{n_i}{\sum n_i}}{\sum \log\frac{n_i}{\sum n_i}} + \frac{B \cdot \log(B^2 + \mu_0)}{\sum \log(B^2 + \mu_0)}\right)$$
$$B = \frac{v_i \cdot \sum n_i}{\sum(v_i \times n_i)} - 1, \tag{5}$$

where $G_0(V, N, \Gamma_0)$ is the calculation function. For each pixel i in the view, its pixel value is v_i, and the value set of all pixels is V, satisfying $v_i \in V$; for each v_i, its corresponding pixel number is n_i, and the amount set of v_i is N, satisfying $n_i \in N$; and the size of V is equal to the type amount of all pixel values Γ_0, where $1 \leq i \leq \Gamma_0$. μ_0 is an adjustment parameter and satisfies $\mu_0 \leq 1$. Due to the definition of G_1 and G_2 is like G_0, it will not be repeated for simplicity. In order to find the illumination effect on the parking spot line and the ground, we, respectively, calculate the ground one by R_s^g and R_h^g, and the line one by R_s^l and R_h^l. Specifically, when the illumination is simple or single, we just obtain the median of R_s^g and R_h^g, and the illumination effect is $(R_s^g + R_h^g)/2$; if the illumination is complex, we need to subdivide the original area and extract the average value of the illumination of each sub-area to decide a multi-level illumination balance strategy.

Table 1 A comparison result table of corner detection experiments based on HERV 2018 dataset

Methods		Testing Images								
		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Average
ATSS [31]	FD (10^{-4})	11.00	10.47	10.62	10.85	10.47	10.44	10.67	10.69	10.65
	Precision	0.0600	0.1050	0.0900	0.0700	0.1050	0.1100	0.0900	0.0850	0.0894
	Recall	0.6000	0.9500	**0.8750**	**0.8195**	0.8291	0.6763	0.6917	0.8282	0.7837
	Quality	4.50	7.75	7.00	6.09	6.49	5.74	5.79	6.20	6.20
Faster R-CNN [17]	FD (10^{-4})	4.16	3.63	3.97	3.04	2.63	3.64	3.93	3.27	3.53
	Precision	0.1438	0.2169	0.1594	0.2308	0.3114	0.2724	0.2174	0.1898	0.2178
	Recall	0.5500	0.9500	0.8125	0.7014	0.7351	0.6795	0.6834	0.7786	0.7357
	Quality	4.50	7.60	6.50	6.50	5.69	5.36	5.27	5.89	5.91
Retina Net [11]	FD (10^{-4})	9.65	8.83	9.16	10.04	9.77	9.97	10.25	9.17	9.61
	Precision	0.1345	0.1704	0.1069	0.0892	0.1392	0.1415	0.1250	0.1246	0.1294
	Recall	0.7500	0.9375	0.6875	0.6389	0.6582	0.5609	0.5917	0.7282	0.6941
	Quality	6.20	7.60	5.13	4.63	5.31	4.65	4.84	6.07	5.55
SSD [12]	FD (10^{-4})	0.76	0.65	0.35	0.53	0.76	0.41	0.18	0.58	0.53
	Precision	0.4191	0.6275	0.6905	0.6905	0.6362	0.7921	0.8250	0.6058	0.6608
	Recall	0.6000	**0.9617**	0.8125	0.8125	0.9231	0.7565	0.6334	0.7286	0.7785
	Quality	4.70	7.88	6.13	6.14	7.12	5.90	5.17	5.36	6.05
Our FMTD	FD (10^{-4})	0.57	0.28	0.19	0.39	0.47	0.27	0.17	0.31	0.33
	Precision	0.6250	0.7189	0.8421	0.7657	0.8526	0.8993	0.8827	0.8514	0.8047
	Recall	0.6000	0.9107	0.7921	0.8053	0.9137	0.9141	0.9146	0.8861	0.8421
	Quality	4.90	7.52	6.36	6.03	7.41	6.88	7.39	6.96	6.68
Our FMTD + LAIB	FD (10^{-4})	**0.14**	**0.01**	**0.06**	**0.18**	**0.12**	**0.06**	**0.12**	**0.12**	**0.10**
	Precision	**0.8675**	**0.9817**	**0.9167**	**0.8258**	**0.9091**	**0.9616**	**0.9499**	**0.9169**	**0.9162**
	Recall	**0.7690**	0.9431	0.8125	0.8125	**0.9316**	**0.9789**	**0.9561**	**0.9286**	**0.8915**
	Quality	**6.43**	**8.33**	**6.63**	**6.28**	**7.63**	**7.85**	**8.00**	**7.29**	**7.31**

5 Fast micro-target detection for corners

The existing parking spot detection algorithm focuses on identifying the type of detected sidelines or vertex angles. Usually, these various types will not only need large numbers of training cases for deep convolutional neural network, but also make the subsequent identification cumbersome. In order to solve this problem, we consider using corners as the alternative and propose a Fast Micro-Target Detection (FMTD) algorithm.

5.1 Corner properties

As Fig. 4 shown, though vehicles have various appearances, their ground projection can always be abstracted to some rectangles, which makes the common parking spot is a rectangle or parallelogram. For a regular parking spot, Zhang et al. [30] used the vertex angle as the detection target, but pay much attention to consider different types of them. Considering that the corner is the basis of sidelines and vertex angles, we choose it as an alternative.

5.2 Corner selection

Although it is difficult to catch the position of missing corners outside the view, we can find some alternatives: when these missing corners approach the view boundary along the sideline they belong to, they will eventually intersect with the boundary. We call the real corner on the parking spot line the reliable corner and the intersection the hypothetical corner, which can be further subdivided into the occlusion one and the edge one according to their positions in the ground image. The occlusion hypothetical corner is not only caused by the view limitation of car cameras, but also from the occlusions of vehicles or other obstacles. In Fig. 5, we show several samples of above corners. With introducing of hypothetical corners, we improve the single-image utilization, though some parking spots, like $P_1(b)P_2(g)P_3P_4$, have the deformation.

5.3 Corner detection

Since corners are relatively micro-targets relative to the whole parking spot, it is necessary to improve the detection ability of the detection model for micro-target. Thus, we have proposed the following improvement strategy: We add a spatial pyramid pooling layer with a five-window structure. In this way, we can use more local small images to train our network, so that the network's learning ability of local features can be greatly enhanced. At the same time, we also use the pooling result obtained by different windows for convolution processing.

6 Generative parking spot location

After the corner detection, we propose a Generative Parking Spot Location (GPSL) method to recover the effective parking spot, which will construct a fully pairing map using a pairwise pairing method and extract useful information from the original image as clues for correcting the result.

6.1 Sideline clue

We extract parking spot sidelines from the captured ground image and use them to depict parking spots. Specifically, we will compare the connection of each corner pair (i, j), defined as $K^s(i, j)$, with the group of parking spot sidelines Q^s:

$$D^s(i, j, \rho, b) = \omega^s + \left\| \gamma_1^s \cdot \left| \log \frac{\Delta\rho}{\xi^s} \right| + \gamma_2^s \cdot \sum_{i,j} \log \frac{\rho \cdot x + b}{y} \right\|_{-\infty}, \tag{6}$$

where $D^s(i, j, \rho, b)$ is the coincidence degree between $K^s(i, j)$ and Q^s. ρ and b are the parameter of Q^s, satisfying $y = \rho \cdot x + b$. ω^s is a coincidence degree threshold and $\omega^s < 1$. γ_1^s and γ_2^s are, respectively, the line proportion and the point proportion, satisfying $0 < \gamma_1^s, \gamma_2^s < 1$. $\Delta\rho$ is the slope difference between $K^s(i, j)$ and Q^s. ξ^s is a basic threshold for controlling the line coincidence degree and $\xi^s < 1$. If and only if $D^s(i, j, \rho, b) \leq 0$, we consider the connection $K^s(i, j)$ is a part of one reliable sideline.

6.2 Occlusion clue

Since the boundary of occluded areas is usually irregular, we directly scan each pixel on the connection of an occlusion hypothetical corner pair $K^o(i, j)$:

$$D^o(i, j) = \omega^o + \gamma_1^o \cdot \frac{\Delta L}{\xi^o} + \gamma_2^o \cdot \frac{\sum_{i,j} \left\{1 \middle| \log\left(\frac{v}{255} + q\right) \leq 0\right\}}{\sum_{i,j} 1}, \tag{7}$$

where $D^o(i, j)$ is the confidence of connections $K^o(i, j)$. ω^o is a confidence threshold and $\omega^o < 0$. γ_1^o and γ_2^o are, respectively, the line length proportion and the pixel value proportion, satisfying $0 < \gamma_1^o, \gamma_2^o < 1$. ΔL is the Euclidean distance of the occlusion hypothetical corner i and j. ξ^o is a basic threshold for controlling the line length confidence and $\xi^o > 1$. The pixel value is expressed as $\log(v/255 + q)$, where $v \in [0, 255]$ and q is a preset threshold determined by the occlusion area's type. If and only if $D^o(i, j) \leq 0$, we think the connection $K^o(i, j)$ is a part of the occlusion area boundary.

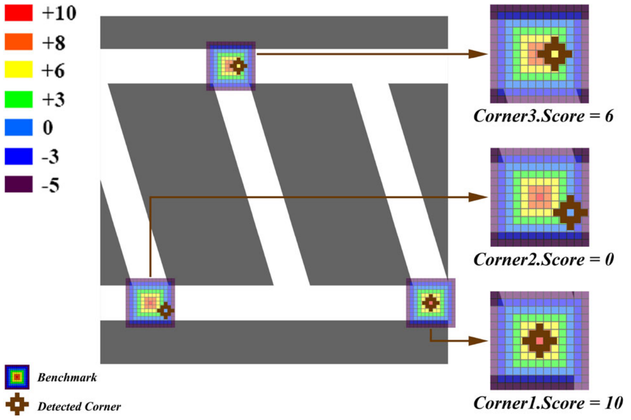


Fig. 6 A schematic diagram of the rating benchmark. In this figure, different distribution areas are marked with various colors, and each one has its own score. It is noted that the score is allowed to be negative. Three samples have been shown in the right column

6.3 Edge clue

For the connection problem of an edge hypothetical corner pair $K^e(i, j)$, we use the following equation:

$$D^e(i, j, \rho) = \omega^e + \left\| \log\left(\left| \frac{\Delta\rho}{\xi^e} \right| \right) \right\|_{-\infty}. \tag{8}$$

where $D^e(i, j, \rho)$ is the coincidence degree between the connection of $K^e(i, j)$ and the image edge group Q^e. ρ is the slope of Q^e. ω^e is a coincidence degree threshold and $\omega^e < 1$. $\Delta\rho$ is the slope difference between $K^e(i, j)$ and Q^e. ξ^e is a basic threshold for controlling the line coincidence degree and $\xi^e < 1$.

As for the connection problem between two types of hypothetical corners, we add the marginal assistance point and the vertex for correcting the deformation, where the former is the intersection of occlusion areas and the view. Specifically, there are two cases: when targets are edge hypothetical corners, vertices, and marginal assistance points, we just adjust the type of i and j in Eq. (8); and when targets are occlusion hypothetical corners and marginal assistance points, we can replace i or j with the marginal assistance point in Eq. (7).

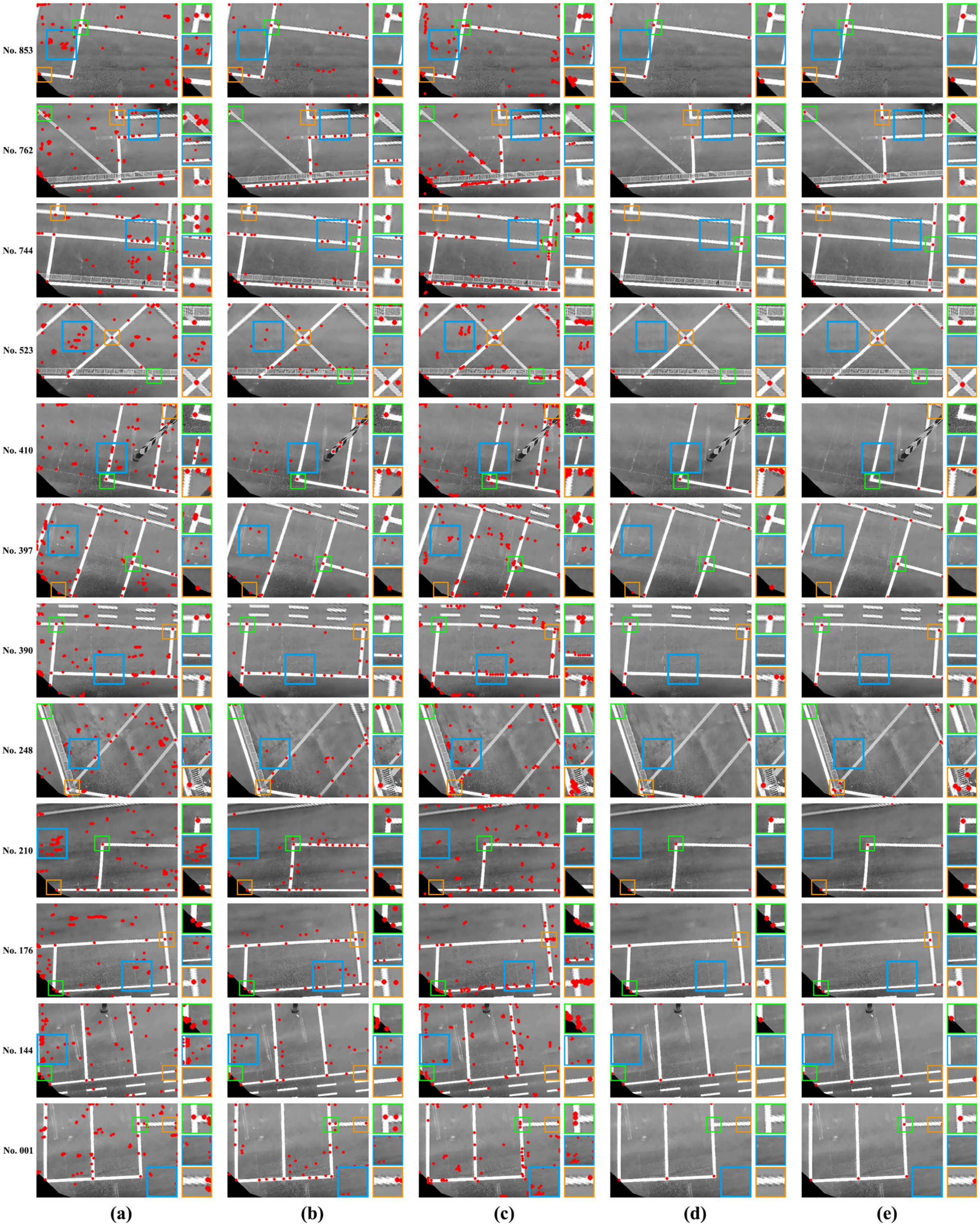
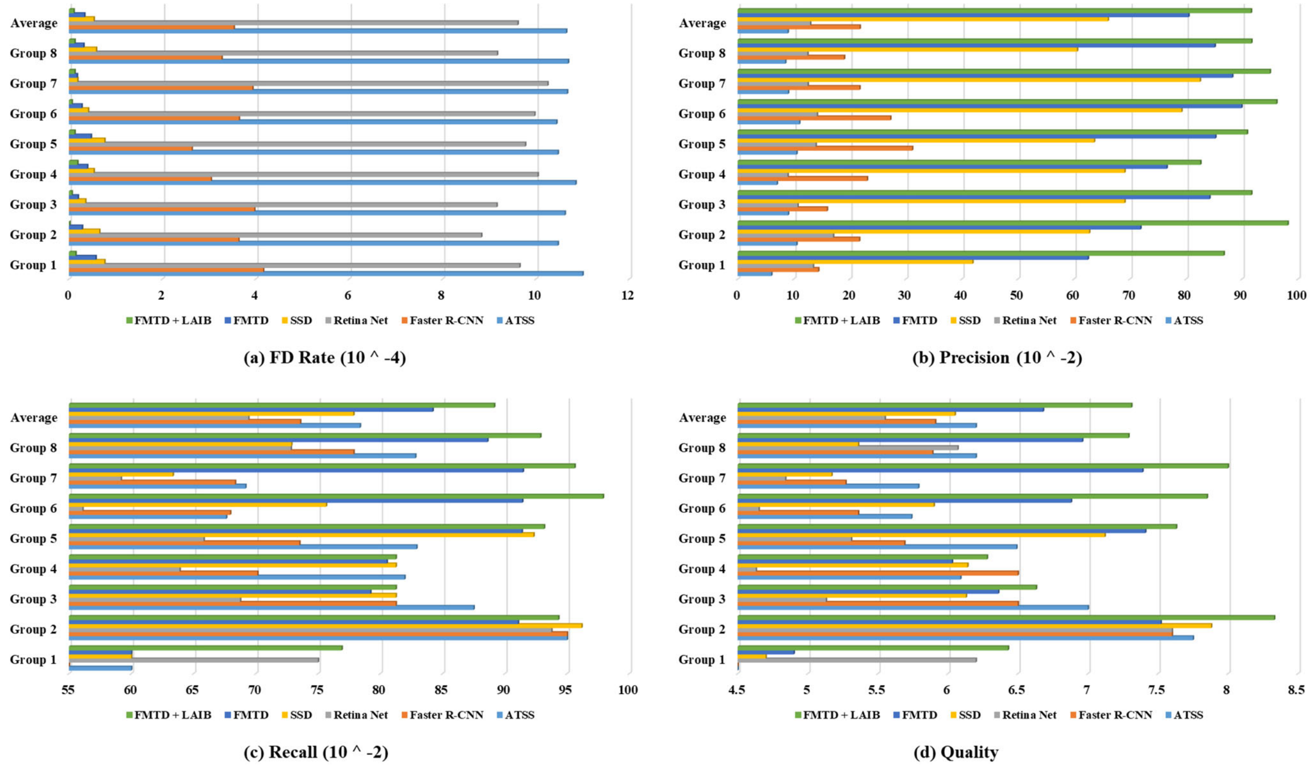


Fig. 7 A comparison result diagram of different corner detection algorithms on HERV 2018 dataset. In this figure, columns 1 to 5 are the result of **a** ATSS [31], **b** Faster R-CNN [17], **c** Retina Net [11], **d** SSD [12], and **e** our FMTD, respectively



(a) FD Rate (10 ^ -4)

(b) Precision (10 ^ -2)

(c) Recall (10 ^ -2)

(d) Quality

Fig. 8 A statistical graph of comparative experimental results on HERV 2018 dataset. In this figure, **a** is the FD rate, **b** is the precision rate, **c** is the recall rate, and **d** is the quality

In the former case, the marginal assistance point is also the occlusion hypothetical corners.

6.4 Domain clue

In order to select the effective parking spot from these polygonal domains divided by above clues, we take the level of points, lines, and areas for consideration:

$$D^d(P^n, L^m, S) = \min\{D_p^d(P^n), D_l^d(L^m), D_a^d(S)\}, \qquad (9)$$

where $D^d(P^n, L^m, S)$ is the final discriminator containing three sub-discriminators: the point one $D_p^d(P^n)$, the line one $D_l^d(L^m)$, and the area one $D_a^d(S)$.

In $D_p^d(P^n)$, P^n is the set of n vertices. For each p_i, its type j can be $\{1, 2, 3\}$, respectively, representing reliable corners, hypothetical corners, and the assistance point; the corresponding weight η_j is presented and $\sum \eta_j = 1$, while the corresponding non-negative score v_j is decreasing and no more than 1:

$$D_p^d(P^n) = \frac{1}{n} \cdot \sum_{i=1}^{n} \{\eta_j \cdot v_j | p_i = j, j = 1, 2, 3\}. \qquad (10)$$

In $D_l^d(L^m)$, L^m is the set of m sidelines. l_i is a sideline of L^m and $1 \le i \le m$. ξ_1^d is a standard ratio of long and short sidelines according to the real parking spot type and $\xi_1^d \ge 1.5$:

$$D_l^d(L^m) = \frac{1}{\xi_1^d} \cdot \left| \mathrm{minlog}\left(\frac{l_i}{l_{i\pm1}}\right)\right|. \qquad (11)$$

In $D_a^d(S)$, S is the area and ξ_2^d is a standard value according to the real parking spot type and $\xi_2^d \ge 1.2$:

$$D_a^d(S) = \log\left(\sqrt[4]{1/S}\right)/\xi_2^d. \qquad (12)$$

7 Experimental results

Our algorithm is proposed in the design phase of a real project and is fully verified during the implementation process. Datasets used for comparison experiments are HERV 2018 and HERV 2019, both of which are provided by the foundation engineering supported by the Huayu Automotive Systems Co., Limited (HASCO) and the East China University of Science and Technology (ECUST). Specifically, the HERV 2018 dataset contains more than 800 processed bird's-eye views with a size of 360 × 240; the HERV 2019 dataset contains 440 processed fish-eye camera images with a size of 900 × 350.

7.1 Corner detection

In Table 1, we show the data comparison result of our proposed FMTD algorithm with several classic object detection algorithms which are ATSS [31], Faster R-CNN [17], Retina Net [11], and SSD [12] on HERV 2018 dataset. We randomly split the original dataset into 8 groups where there are 100 images in each group. Every time, we use 7 groups as the training set which also contains the validation set, and 1 group as the testing set. We choose False Detection rate (FD rate), perception rate, recall rate, and the quality as the comparison item. The quality is designed to describe the average distance between detected corners and the real corner, which is abstracted as a score, and the rating rule is shown in Fig. 6.

In Fig. 7, we show the comparison result of above five algorithms. As this figure shown, there are 12 samples, and for each sample, we select several domains to compare the performance in details: blue boxes are used to show error detection results, while green and orange boxes are for the result ought to be detected. According to this figure, it is obvious that our method can greatly reduce the probability of error detection results when compared with ATSS, Faster R-CNN, and Retina Net; while compared with Faster R-CNN and SSD, our method has a higher accuracy. Besides, our method also can achieve a higher score in precision of the detected corner position. In Fig. 8, we show the corresponding statistical graph of comparison results in Table 1. As shown in this figure, our FMTD method always can achieve the better performance than other algorithms.

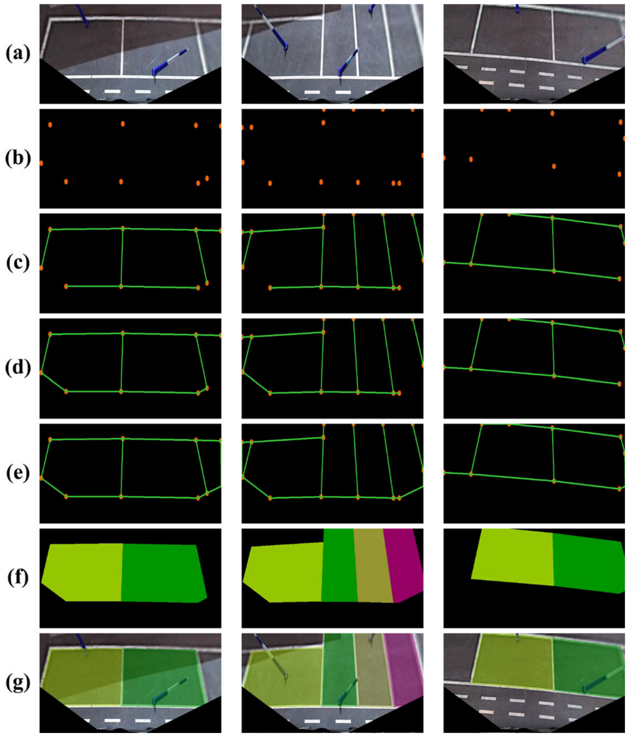


Fig. 9 A comparison result diagram of proposed algorithm on HERV 2019 dataset. In this figure, **a** is the original ground image; **b** is the detection result of corners; rows 3–6 are the recovery result of using **c** SC, **d** OC, **e** EC, and **f** DC, respectively; **g** is the final parking spot marking map

7.2 Parking spot detection

In Fig. 9, we show the experimental result of proposed multi-clue recovery processing on HERV 2019 dataset. In this figure, (b) is the result of FMTD without fully pairing processing, and it is transported into the multi-clue recovery model as the input. Next, four rows are, respectively, the result of recovery processing using sideline clues, occlusion clues, edge clues, and domain clues. Specifically, the order of edge clue processing and occlusion clue processing can be changed. As this figure shown, our multi-clue recovery model can effectively locate parking spots and mark them in this map. At the same time, it can also deal with the deformation of the parking spot well and ensure that the recovered parking spot meets the intuitive perception of human eyes.

In Fig. 10, we compare the proposed GPSL algorithm with the classic instance segmentation algorithm, Yolact [4]. The testing dataset for comparison experiments is HERV 2018 dataset. In this figure, there are a total of 5 different samples, and for each sample, we have shown the original top-view image in row (a), the ground truth in row (b), the location result of Yolact in row (c), and the one of our GPSL in row (d). It is obvious that our result is closer to the given ground truth, and our method can achieve a lower miss detection rate in all samples. Although our method will filter some parking spots which appear at the marginal area, this strategy can fit the principle of proximity, and at the same time guarantee to a certain extent the accuracy and availability of the input used in the vehicle parking guidance algorithm.

In Table 2, we show the comparison result of the parking spot location experiment by using the HERV 2018 dataset. We choose the precision rate, the recall rate and the score to explain the performance difference among Yolact [4] and our GPSL method. The calculation rule of score is:

$$\#\text{Score} = \sum_i \left(\frac{s_i}{\sum_i s_i} \cdot (\beta \cdot \sum_j p_{ij} + (1-\beta) \cdot \sum_r l_{ir}) \right), \qquad (13)$$

where we use the area ratio $s_i/(\sum_i s_i)$ to indicate the importance of each parking spot, which is scored by their points and sidelines, and β is used to adjust the ratio of point score p_{ij} and sideline score l_{ir}. The final score has been normalized. In Fig. 11, we show the corresponding statistical graph of comparison results in Table 2. As shown in this figure, our proposed GPSL method has achieved a satisfactory performance.

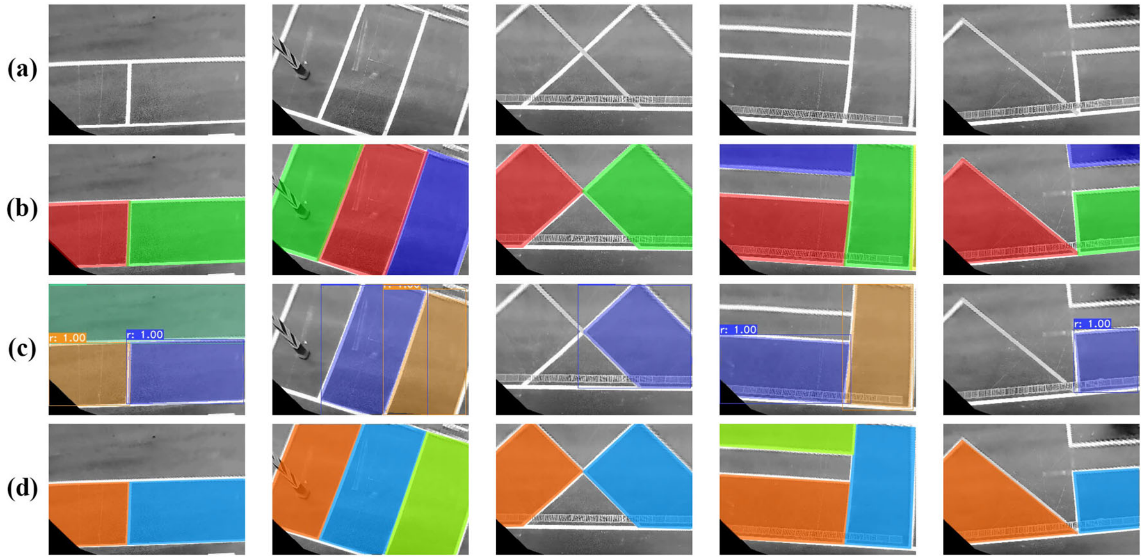


Fig. 10 A comparison result diagram of different parking spot detection algorithms on HERV 2018 dataset. In this figure, rows 1 to 4 are, respectively **a** original image, **b** ground truth, the result of **c** Yolact [4] and the result of **d** our GPSL

Table 2 A comparison result table of parking spot location experiments based on HERV 2018 dataset

Methods		Testing Images								
		Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Average
Yolact [4]	Precision	0.83	0.86	0.54	0.87	0.95	0.96	0.84	0.85	0.84
	Recall	0.72	0.67	0.61	0.61	0.50	0.44	0.72	0.67	0.62
	Score	0.60	0.51	0.46	0.50	0.56	0.50	0.64	0.59	0.55
Our GPSL	Precision	**0.91**	**0.93**	**0.94**	**0.95**	**0.96**	**0.96**	**0.93**	**0.93**	**0.94**
	Recall	**0.91**	**0.82**	**0.84**	**0.77**	**0.77**	**0.79**	**0.84**	**0.86**	**0.83**
	Score	**0.92**	**0.93**	**0.91**	**0.94**	**0.95**	**0.93**	**0.95**	**0.93**	**0.93**

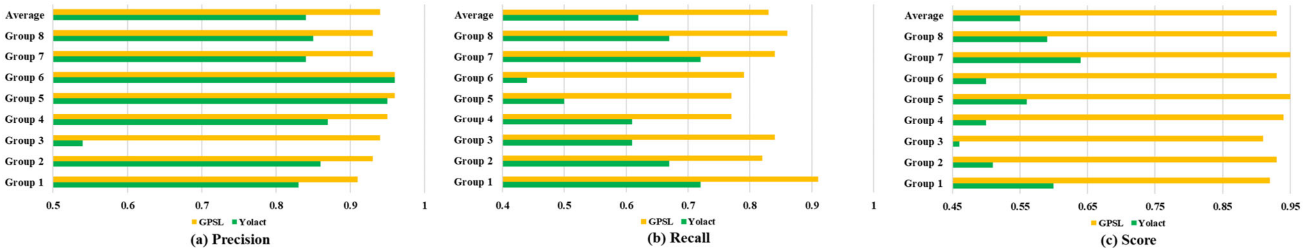


Fig. 11 A statistical graph of comparative experimental results on HERV 2018 dataset. In this figure, **a** is the precision rate, **b** is the recall rate, and **c** is the score

Besides, in Tables 3 and 4, we show the comparison result of the parking spot detection experiment by using two public parking spot datasets, PS 2.0 [30] and PPSD [27]. We also choose the precision rate, recall rate and score to evaluate the performance of each method. Since the around view images in these two datasets contains several areas with various light intensities, we have to increase the range of illumination balance, which increases the algorithm complexity; at the same time, because the area of parking spots in the around view image is small, we need to adjust some parameters of the multi-clue recovery model for adapting the need of generating a small parking spot; in addition, the clarity of around view image also poses a great challenge to our algorithm. As shown in Tables 3 and 4, our GPSD method has the most balanced performance in all aspects.

Table 3 The performance of different parking spot detection algorithm on the PS 2.0 dataset

Methods	Precision	Recall	Score
DeepPS [30]	0.8132	0.7924	0.889
DMPR-PS [8]	0.8857	0.8329	0.913
PSDet [27]	**0.9519**	**0.9793**	0.968
our GPSD	0.9321	0.9684	**0.972**

Table 4 The performance of different parking spot detection algorithm on the PPSD dataset

Methods	Precision	Recall	Score
DeepPS [30]	0.9841	0.9893	0.963
DMPR-PS [8]	**0.9915**	0.9917	0.975
PSDet [27]	0.9881	**0.9936**	0.979
our GPSD	0.9863	0.9897	**0.981**

8 Conclusion and future work

Different from existing methods, this paper proposes a Generative Parking Spot Detection algorithm which focuses on using the corner to recover parking spots. For improving the accuracy of corner detection, we proposed a layered analytical illumination balance method and designed a fast micro-target detection network. And we use the multi-clue model to correct the result of fully pairing processing. According to the experimental result, our method can achieve a higher score both in the corner detection and the parking spot location. Because our proposed algorithm is aimed as the detection task of common parallelogram parking spots, it is very sensitive to the deformation of parking spots. In addition, the sample number of used datasets is small, and the scene type is single. So in the future, we will both improve the parking spot generation strategy for strengthening the algorithm robustness, and extended the dataset by adding more scene types and surrounding environmental factors.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61672228, 62077037, 61872241, 62072449 and 61632003, in part by the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 1837, in part by the Science and Technology Commission of Shanghai Municipality under Grants 18410750700 and 17411952600, and in part by The Hong Kong Polytechnic University under Grants P0030419, P0030929 and P0035358.

Declaration

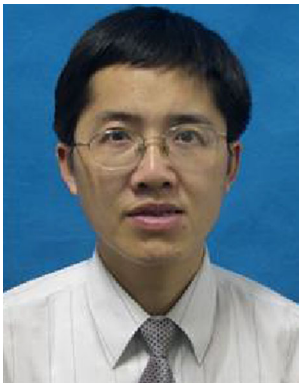
Conflict of interest All authors declare that they have no conflict of interest.

References

1. Athira, A., Lekshmi, S., Vijayan, P., Kurian, B.: Smart parking system based on optical character recognition. In: 3rd International Conference on Trends in Electronics and Informatics, pp. 1184–1188 (2019)
2. Bacchiani, G., Patander, M., Cionini, A., Giaquinto, D.: Parking slots detection on the equivalence sphere with a progressive probabilistic Hough transform. In: IEEE 20th International Conference on Intelligent Transportation Systems, pp. 1–6 (2017)
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
4. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: Real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9157–9166 (2019)
5. Dixit, M., Srimathi, C., Doss, R., Loke, S., Saleemdurai, M.: Smart parking with computer vision and iot technology. In: 43rd International Conference on Telecommunications and Signal Processing, pp. 170–174 (2020)
6. Hakim, I.M., Christover, D., Marindra, A.M.J.: Implementation of an image processing based smart parking system using Haar-cascade method. In: IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 222–227 (2019)
7. Hamada, K., Hu, Z., Fan, M., Chen, H.: Surround view based parking lot detection and tracking. In: IEEE Intelligent Vehicles Symposium (IV), pp. 1106–1111 (2015)
8. Huang, J., Zhang, L., Shen, Y., Zhang, H., Zhao, S., Yang, Y.: Dmpr-ps: a novel approach for parking-slot detection using directional marking-point regression. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 212–217 (2019)
9. Jian, D.H., Lin, C.H.: Vision-based parking slot detection based on end-to-end semantic segmentation training. In: IEEE International Conference on Consumer Electronics, pp. 1–4 (2020)
10. Lee, M., Kim, S., Lim, W., Sunwoo, M.: Probabilistic occupancy filter for parking slot marker detection in an autonomous parking system using AVM. IEEE Trans. Intell. Transp. Syst. **20**(6), 2389–2394 (2018)
11. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
13. Panzani, G., Nava, D., Savaresi, S.M.: An odometry free automatic perpendicular parking strategy for a light urban vehicle based on a low resolution lidar. In: IEEE Intelligent Transportation Systems Conference, pp. 2772–2777 (2019)
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
15. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
16. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
18. Scheunert, U., Fardi, B., Mattern, N., Wanielik, G., Keppeler, N.: Free space determination for parking slots using a 3d pmd sensor. In: 2007 IEEE Intelligent Vehicles Symposium, pp. 154–159 (2007)

19. Sedighi, S., Nguyen, D.V., Kuhnert, K.D.: Implementation of a parking state machine on vision-based auto parking systems for perpendicular parking scenarios. In: 6th International Conference on Control, Decision and Information Technologies, pp. 1711–1716 (2019)
20. Singh, T., Khan, S.S., Chadokar, S.: A review on automatic parking space occupancy detection. In: International Conference on Advanced Computation and Telecommunication, pp. 1–5 (2018)
21. Song, J., Zhang, W., Wu, X., Cao, H., Gao, Q., Luo, S.: Laser-based slam automatic parallel parking path planning and tracking for passenger vehicle. IET Intell. Transp. Syst. **13**(10), 1557–1568 (2019)
22. Suhr, J., Jung, H.: Sensor fusion-based precise obstacle localisation for automatic parking systems. Electron. Lett. **54**(7), 445–447 (2018)
23. Suhr, J.K., Jung, H.G.: Full-automatic recognition of various parking slot markings using a hierarchical tree structure. Opt. Eng. **52**(3), 037,203 (2013)
24. Suhr, J.K., Jung, H.G.: End-to-end trainable one-stage parking slot detection integrating global and local information. IEEE Trans. Intell. Transp. Syst. **6**, 66 (2021)
25. 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. **21**(2), 163–176 (2010)
26. Unger, C., Wahl, E., Ilic, S.: Parking assistance using dense motion-stereo. Mach. Vis. Appl. **25**(3), 561–581 (2014)
27. Wu, Z., Sun, W., Wang, M., Wang, X., Ding, L., Wang, F.: Psdet: efficient and universal parking slot detection. In: IEEE Intelligent Vehicles Symposium (IV), pp. 290–297 (2020)
28. Yamamoto, K., Watanabe, K., Nagai, I.: Proposal of an environmental recognition method for automatic parking by an image-based CNN. In: IEEE International Conference on Mechatronics and Automation, pp. 833–838 (2019)
29. Ye, C., Chen, G., Qu, S., Yang, Q., Chen, K., Du, J., Hu, R.: Self-localization of parking robots using square-like landmarks. In: IEEE International Conference on Robotics and Biomimetics, pp. 1987–1992 (2018)
30. Zhang, L., Huang, J., Li, X., Xiong, L.: Vision-based parking-slot detection: a dcnn-based approach and a large-scale benchmark dataset. IEEE Trans. Image Process. **27**(11), 5350–5364 (2018)
31. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. arXiv preprint arXiv:1912.02424 (2019)
32. Zinelli, A., Musto, L., Pizzati, F.: A deep-learning approach for parking slot detection on surround-view images. In: IEEE Intelligent Vehicles Symposium, pp. 683–688 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Zhihua Chen received his Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently a Full Professor in the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests include image/video processing and computer vision.

Jun Qiu received the B.Eng. degree in computer science from the East China University of Science and Technology, Shanghai, China, in 2018. He is currently pursuing the M.Eng. degree in computer science with the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China. His current research interests include computer vision, image processing, and deep learning.

Bin Sheng received the B.A. degree in English and the B.Eng. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2004, and the M.Sc. degree in software engineering from the University of Macau, Taipa, Macau, in 2007, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2011. He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He is an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology. His current research interests include virtual reality and computer graphics.

Ping Li received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2013. He is currently a Research Assistant Professor with The Hong Kong Polytechnic University, Kowloon, Hong Kong. He has one image/video processing national invention patent and has excellent research project reported worldwide by ACM TechNews. His current research interests include image/video stylization, colorization, artistic rendering and synthesis, and creative media.

Enhua Wu received the B.Sc. degree from Tsinghua University, Beijing, China, in 1970, and the Ph.D. degree from The University of Manchester, Manchester, UK, in 1984. He is currently a Research Professor with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. He is also an Emeritus Professor with the University of Macau, Macau, China. His current research interests include realistic image synthesis and virtual reality. He has served as chair or co-chair for various conferences, including SIGGRAPH Asia 2016. He is an Associate Editor-in-Chief of the Journal of Computer Science and Technology. He is a Member of the IEEE and ACM, and a Fellow of the China Computer Federation (CCF).