

Order-independent Matching with Shape Similarity for Parking Slot Detection

Ziyi Yin¹

yzy19980922@stu.xjtu.edu.cn

Ruijin Liu¹

lrj466097290@stu.xjtu.edu.cn

Zejian Yuan¹

yuan.ze.jian@xjtu.edu.cn

Zhiliang Xiong²

leslie.xiong@forward-innovation.com

¹ Institute of Artificial Intelligence and
Robotics,
Xi'an Jiaotong University,
China

² Shenzhen Forward Innovation Digital
Technology Co. Ltd,
China

Abstract

Current mainstream methods adopt point regression and prior sizes to detect parking slots. Although these methods are approved to be useful in majority of circumstances, they have many limitations when they adapt to size and shape variations in more general slots because of a pre-defined fixed order, lack of holistic constraints and adequately diverse data. To address them, we propose an order-independent matching strategy with shape similarity to handle the more general slot sizes and shapes. The matching strategy adopts a two-level procedure: the point-level and the parking slot-level, that finds optimal order and association adaptively. More importantly, we adopt shape similarity to represent holistic geometry to rank slots so as to suppress the missshapen ones. Furthermore, we collected a large-scale and remote-view parking slot dataset (LRPS) to improve data diversity. It contains a large number of general parking environments, as well as slots of various shapes and sizes across different cities, daytime and interior-exterior scenes. The proposed approach is evaluated on the LRPS dataset and achieves superior performance to previous methods.

1 Introduction

Vision-based parking slots detection aims to predict the generic shapes of parking slots in a surrounded-view image, which is critical for the navigating of the Autonomous Valet Parking (AVP) system [1, 2, 3, 4, 5]. The task requires the model to capture complex visual cues of highly variable parking slot patterns, such as shapes or visibility in distinct traffic scenarios.

Mainstream methods [6, 7, 8, 9, 10, 11] first detect two entrance marking points then generate the whole parking slots based on limited prior sizes and post-processing. Recently, the single-stage pipeline [12] replaces the upper pipeline by directly regressing four corner points to form more flexible slots. However, they still face three main challenges. The first is the order-induced rotation problem that false rotated detection would be generated in critical poses. The second is the lacking holistic representation when the model is

only trained with corner regression. With the same corner errors, the shape of slots would change dramatically. The third is the lack of sufficient data to represent the complex light environments and various slot shapes, sizes and appearances.

To tackle these issues, we propose an order-independent matching with shape similarity to train a single-stage transformer-based network. The matching strategy consists of two levels. The point-level seeks optimal order to assign each predicted corner with certain ground truth. Except for the common distance cost of four corners, it incorporates shape similarity to represent holistic geometry for further aligning different slots. By combining distance and shape costs, the network will benefit from samples of similar shapes to predict more accurate parking slots. Based on optimal orders, the parking slot-level constructs correspondence between prediction and ground truth to find positives and negatives for network learning. Such matching processes can be formulated as two bipartite graph matching problems and solved by the Hungarian algorithm efficiently. Besides, the used transformer-based architecture utilizes the self-attention mechanism to model non-local dependencies, allowing the network to better capture complex parking environment typologies and various slot shapes.

Furthermore, we collect a large-scale and remote-view parking slots dataset (LRPS) that contains more diverse environments to benefit real applications. Images are synthesized from four surrounded cameras at a remote view to include numerous parking slots. In addition, the LRPS dataset considers more general scenarios and parking slot types to improve diversity compared with the previous datasets. Finally, experiment results on the LRPS dataset demonstrate that our method achieves superior performance than previous methods.

2 Related Work

Current vision-based methods of parking slot detection can be divided into two categories: (1) marking points-based and (2) corner points-based. The marking points-based methods [1, 2, 3, 4, 5, 6, 7, 8] firstly estimate the two entrance marking points of each parking slot, then use prior size standards to generate the whole slot. DeepPS [9] and DMPR-PS [8] leverage geometric rules designed by the physical prior of real driving scenarios to post-process the detected marking points of different entrances. APS [10] uses an additional graph neural network to fine-tune the detected marking points and model the internal relationships. Such methods achieve considerable performance but suffer from limited prior sizes, making them inefficient to be applied for natural diverse driving environments.

Recently, the corner-based method Faster-PS [11] has been proposed. It directly predicts four corners to generate flexible parking slots without any post-processing and achieves convincing performance. However, it pre-defines the order of four corners to be fixed aligned with ground truth, resulting in unsuspected rotated false detection for certain poses. Besides, it only adopts location error, flawed to distinguish holistic deformations, leading to bad generality.

Different from them, our method is an end-to-end one that estimates four corners while eliminating the false rotation and shape deformation. The proposed order-independent matching drops the pre-defined fixed order by adaptively assigning each predicted corner with a slot corner, enabling the network to seek the best-matched order automatically. Besides, the incorporated shape cost further punish deformed predictions which have the same location errors but are not similar in shape. This filters out ineffective training samples to benefit the corner regression for generalized slot shapes.

In addition to training strategy, sufficiently diverse data is also indispensable. Previous

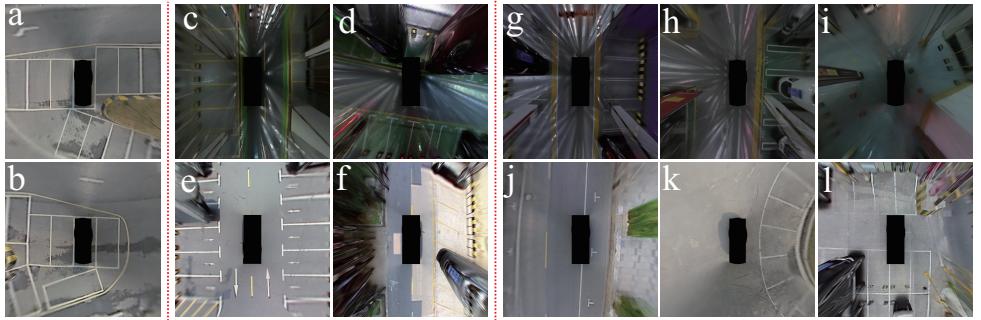


Figure 1: A visualization of LRPS dataset. *a* ~ *b* represent the complex topology among different parking slots. *c* ~ *f* are the common types of parking slots, where *c*, *d* denote indoor and *e*, *f* denote outdoor. More general conditions are displayed in *g* ~ *l*, including U-shape with arc (*g*), U-shape with right angle (*h*), T-shape with dashed line (*j*), arc aligned (*k*) and abrasion (*i* and *l*).

PS1.0 [8], PS2.0 [19], and PSDD [10] have collected lots of images and annotated parking slots. However, most images only include three or fewer slots on average and represent regular parking slots, resulting in a huge gap between them and the real driving scenarios.

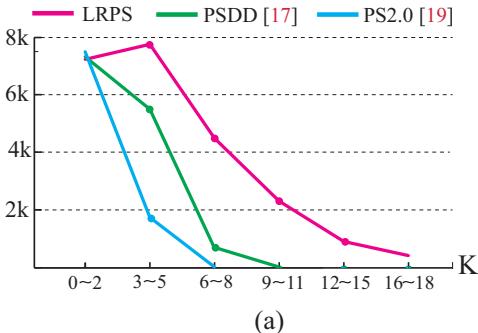
To boost the development of real-world driving applications on automated parking, we have collected and released the largest dataset named large-scale and remote-view parking slots (LRPS) that contains 26k images (2x) and 87k annotated slots (3x). The LRPS dataset comprises scenes with significant lights, road textures, and slot appearance changes. Besides, each image has more general slots with random sizes, shapes, and typology.

3 LRPS Dataset

A large-scale and remote-view parking slots detection dataset is present in this paper. We firstly record more than twenty hours of video through four wide-angle cameras mounted around the ego-vehicle. The videos are collected in Xi'an and Shanghai, China, during daytime from typical scenarios and they include different kinds of parking slots.

The LRPS dataset synthesizes the images from four cameras at a remote view. Unlike existing datasets are usually collected by only concentrating on parking slots near the ego-vehicle, LRPS provides more images containing numerous parking slots, thus increasing the challenge during detection. A more specific comparison is depicted in Tab. 1 (a). Furthermore, these images also exhibit complicated topology among the numerous parking slots, as illustrated in Fig. 1 (a and b).

Besides, the LRPS dataset demonstrates a wide diversity of parking slot shapes. Except for the common types of parking slots, the LRPS dataset additionally introduces U-shape with right angle (*g*) and arc (*h*), T-shape with dashed line (*j*), arc aligned parking slots (*k*) and abrasion (*i* and *l*), as Fig.1 shows. A statistical comparison is shown in Tab. 1 (b). In detail, the LRPS dataset totally comprises 25929 surround-view images and 87121 parking slots, which is nearly 2× larger and contains 3× parking slots than the PSDD dataset [10]. For each surround-view image, the spatial resolution is 1024×1024 , which corresponds to a $10m \times 10m$ physical region. We have divided the dataset into 22132 images for training, 1282 images for validation and 2515 images for testing.



(a)

	PS 2.0	PSDD	LRPS
samples	12 K	14 K	25 K
parking slots with annot	14 K	31 K	87 K
common types	✓	✓	✓
U-shape	✗	✗	✓
T-shape w/o line	✗	✗	✓
deformation	✗	✗	✓
abrasion	✗	✗	✓

(b)

Table 1: (a) is the line chart corresponding to the number of parking slots in the image (K). (b) is the property of existing dataset and our LRPS. The third row to the seventh row demonstrates different types of parking slots with the same definitions as shown in Fig. 1.

4 Methodology

In this section, we define the output and explain the details of our detection approach, including a transformer-based architecture and a matching strategy for training.

4.1 Parking slot representation

We denote $\{c, \mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3, \mathbf{p}^4\}$ as an available parking slot. $c \in \{0, 1, 2\}$ is the class label of the parking slot where 0 is background, 1 is a free parking slot and 2 is an occluded parking slot as shown in Fig. 2 (a). $\{\mathbf{p}^i = (x^i, y^i) | i \in \{1, 2, 3, 4\}\}$ are the coordinates of four corner points, which we do not stipulate the order of these corner points. For the ground truth corner points $\{\hat{\mathbf{p}}^1, \hat{\mathbf{p}}^2, \hat{\mathbf{p}}^3, \hat{\mathbf{p}}^4\}$ of a parking slot, we only arrange them in an anti-clockwise order and do not designate the starting point as Fig 2 (b) shows, which is different from [20].

4.2 Prediction Network

The network is built by a transformer structure following DEtection TRansformer (DE-TR) [20] with only two encoder layers and two decoder layers. As is illustrated in Fig. 3, giving an input image $\mathbf{I} \in \mathbb{R}^{3 \times H_0 \times W_0}$, we apply a CNN to extract the feature map $\mathbf{F} \in \mathbb{R}^{C \times \frac{H_0}{32} \times \frac{W_0}{32}}$ from input image, where the number of feature map channels C is set to 128. Then, \mathbf{F} is added by a sinusoidal positional embedding E_p and transmitted to the transformer encoder (TRE). E_p owns the same shape as \mathbf{F} . After that, the transformer encoder outputs a representation sequence $\mathbf{S}_{enc} \in \mathbb{R}^{C \times \frac{HW}{32 \times 32}}$, which contain the topological relations of clustered parking slots. Next, the transformer decoder (TRD) receives \mathbf{S}_{enc} and a set of learned query embedding $E_l \in \mathbb{R}^{C \times N}$ as inputs, and outputs the hidden states $\mathbf{S}_{dec} \in \mathbb{R}^{C \times N}$. Here we set the length of queries N as 50. Finally, a fully-connected layer transforms \mathbf{S}_{dec} to predict the categories of parking slots and generate N class labels $\{c_i\}_{i=1}^N$. Meanwhile, a three-layer feed-forward network (FFN) predicts four corner points and also with an additional center point p_i^c to capture the global information of the target parking slot, which results in N parking slots predictions $\{\mathbf{g}_i, \mathbf{p}_i^c\}_{i=1}^N$, $\mathbf{g}_i = (\mathbf{p}_i^1, \mathbf{p}_i^2, \mathbf{p}_i^3, \mathbf{p}_i^4)$.

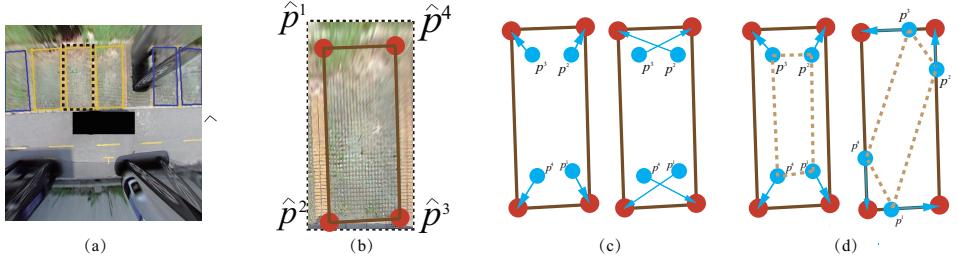


Figure 2: Problem illustration. (a) The yellow lines represent free parking slots and the blue lines represent occluded slots. For (b)-(d), predictions are displayed in blue points and yellow dashed lines. (b) The corner annotation of a free parking slot. Four corners are marked in red. Notice that we do not stipulate the first point and just arrange them in an anti-clockwise order. (c) The points matching problem of a single pair of prediction and ground truth (left). (d) The shape discrepancy problem. With the same distance cost, the shape would changed.

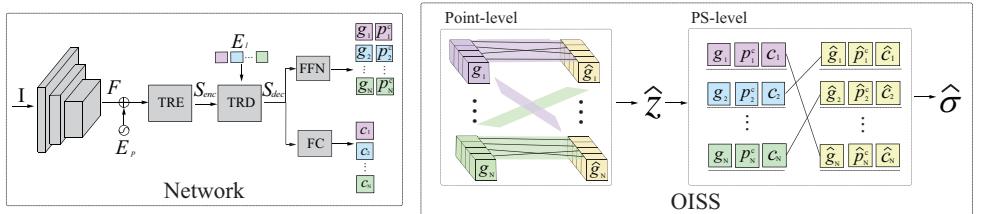


Figure 3: Overall structure. The structure consists of a transformer based network and an order-independent matching strategy with shape similarity.

4.3 Training with OISS

The Order-Independent matching with Shape Similarity (OISS) aims at adaptively matching the corner order and parking slot between predictions and ground truth to handle general slot shapes and sizes. In our setting, the corner order of a predicted slot is independent of certain ground truth, shown in Fig. 2 (c). More importantly, the shape similarity is introduced to represent the holistic geometry to rank predicted slots. In Fig. 2 (d), both slots have the same distance error, but the right-most would be suppressed due to its wrong deformation. After that, the association between the set of predictions and ground truth is also unknown. Therefore, OISS adopts two levels matching, including a point-level matching and a parking slot (PS)-level matching. The point-level firstly associates a single pair of prediction and ground truth to find an optimal order of corners by jointly considering distance and shape costs. The PS-level constructs correspondence between all predictions and ground truth to select training positives and negatives. The following step is that the loss function adopts a linear combination of classification loss and corner regression loss.

Point-level matching. We now denote $\mathbf{g}_i = (\mathbf{p}_i^1, \mathbf{p}_i^2, \mathbf{p}_i^3, \mathbf{p}_i^4)$ and $\hat{\mathbf{g}}_j = (\hat{\mathbf{p}}_j^1, \hat{\mathbf{p}}_j^2, \hat{\mathbf{p}}_j^3, \hat{\mathbf{p}}_j^4)$ as the corner points of i -th predicted parking slot and j -th ground truth. It should be noticed that two sets of corner points are not aligned. In order to search for the optimal matching relations between the predicted and ground truth corner points, we define the following bipartite matching problem:

$$\hat{z} = \arg \min_z d_{i,j}(\mathbf{g}_i, \hat{\mathbf{g}}_j, z), \quad (1)$$

where $\hat{z} : \mathbf{g}_i \rightarrow \hat{\mathbf{g}}_j$ is the optimal permutation of 4 points and $d_{i,j}$ is the matching cost function. Here we define $d_{i,j} = L_d + L_s$, where L_d and L_s are defined as follows:

$$L_d = \sum_{n=1}^4 L_1 \left(\hat{\mathbf{p}}_j^n, \mathbf{p}_i^{z(n)} \right), L_s = \sum_{n=1}^4 \left(1 - \cos \theta_{ij}^n \right), \quad (2)$$

$z(n)$ denotes the index of the predicted points to be matched with n -th ground truth corner points, $n \in \{1, \dots, 4\}$. Eq. 2 is composed of two terms to calculate the position and shape differences. (1) Distance cost L_d . L_d is calculated by using the average absolute error L_1 function to obtain the distance between the ground truth and the corresponding predicted points under a certain permutation. (2) **Shape Cost** L_s . L_s measures the shape difference between the quadrilaterals that enclosed by \mathbf{g}_i and $\hat{\mathbf{g}}_j$. The θ_{ij}^n is $\sqrt{\hat{\mathbf{p}}_j^n \hat{\mathbf{p}}_j^{S(n)} \cdot \mathbf{p}_i^{z(n)} \mathbf{p}_i^{z(S(n))}}$, which represents the angle of two edges. Here $S(n) = n \bmod 4 + 1$. The proposed shape cost is necessary to distinguish deformed predictions since only a distance cost can not punish the wrongly deformed slots. After solving Eq. 1 by using the Hungarian algorithm [2], we define $d_{i,j}^* = d_{i,j}(\mathbf{g}_i, \hat{\mathbf{g}}_j, \hat{z})$.

PS-level matching. Consider the parking slot prediction as $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$, where N is larger than the maximum parking slot number in the image of the dataset. The ground-truth set of parking slots are represented as $\hat{\mathcal{Y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^N$, which are padded with \emptyset (non-instance). The matching between $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$ and $\hat{\mathcal{Y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^N$ can be formulated as a bipartite matching problem to search for an injective function $\hat{\sigma} : \mathcal{Y} \rightarrow \hat{\mathcal{Y}}$ with the lowest costs:

$$\hat{\sigma} = \arg \min_{\sigma} \sum_i^N l_i \left(\hat{\mathbf{y}}_i, \mathbf{y}_{\sigma(i)} \right), \quad (3)$$

where $\sigma(i)$ represents the index of the predicted parking slots to be matched with the i -th ground truth and l_i is the matching cost function. For the i -th ground truth parking slot $\hat{\mathbf{y}}_i = (\hat{c}_i, \hat{\mathbf{g}}_i, \hat{\mathbf{p}}_i^c)$, where \hat{c}_i and $\hat{\mathbf{g}}_i$ are the target class label and corner points. The ground truth center point $\hat{\mathbf{p}}_i^c$ is calculated by averaging the corresponding four corner points. For the $\sigma(i)$ -th predicted parking slot, $p_{\sigma(i)}(\hat{c}_i)$ is the probability of class \hat{c}_i and $\mathbf{p}_{\sigma(i)}^c$ is the predicted center point. With above notations we define the matching cost function l_i as follows:

$$l_i = -\omega_1 p_{\sigma(i)}(\hat{c}_i) + \mathbb{1}_{\{\hat{c}_i \neq 0\}} \omega_2 L_1 \left(\hat{\mathbf{p}}_i^c, \mathbf{p}_{\sigma(i)}^c \right) + \mathbb{1}_{\{\hat{c}_i \neq 0\}} \omega_3 d_{\sigma(i), i}^*. \quad (4)$$

The first term $-\omega_1 p_{\sigma(i)}(\hat{c}_i)$ is the classification cost. The second term $L_1 \left(\hat{\mathbf{p}}_i^c, \mathbf{p}_{\sigma(i)}^c \right)$ measures the distance cost of center points. The third term $d_{\sigma(i), i}^*$ is the optimal point-level matching cost between the $\sigma(i)$ -th predicted and the i -th ground truth corner points, as is mentioned in the previous paragraph. We define $\mathbb{1}(\cdot)$ as an indicator function, and ω_1 , ω_2 and ω_3 are the coefficients to adjust the weights of different terms. After solving Eq. 3 by using the Hungarian algorithm, we acquire the optimal parking slots matching permutation $\hat{\sigma}$.

Loss Function After generating $\hat{\sigma}$ and \hat{z} , we define the loss function as follows:

$$L_P = \sum_{i=1}^N \left[-\omega_1 \log p_{\hat{\sigma}(i)}(\hat{c}_i) + \mathbb{1}_{\{\hat{c}_i \neq 0\}} \omega_2 L_1 \left(\hat{\mathbf{p}}_i^c, \hat{\mathbf{p}}_{\hat{\sigma}(i)}^c \right) + \mathbb{1}_{\{\hat{c}_i \neq 0\}} \omega_3 d_{\hat{\sigma}(i), i}^* \right]. \quad (5)$$

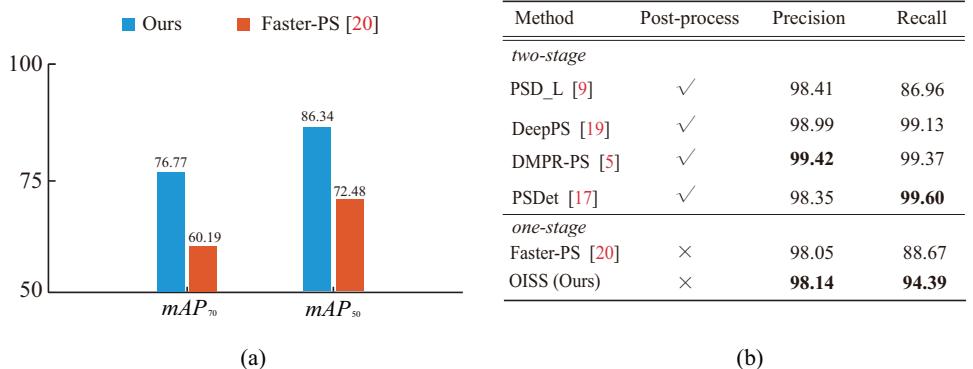


Figure 4: (a) Quantitative evaluation on the LRPS dataset. (b) Evaluation on the PS2.0 dataset [19].

The first term is classification loss and remaining terms are points regression loss. ω_1 , ω_2 and ω_3 are the coefficients to adjust the influence of the loss terms. They have the same value as Eq. 4.

5 Experiments

Datasets. Experiments are conducted on the LRPS dataset described in section 3 and PS2.0 dataset [19]. The widely used PS2.0 dataset [19] contains 12165 samples collected by using an AVM (around view monitoring) system from typical indoor and outdoor scenarios. The resolution of each image is 600×600 . Every parking slot in the PS2.0 dataset is annotated as an entrance line connected by two marking entrance points.

Implementation Details. The height H_0 and width W_0 of an input image are set to 384. Each input image is processed by the augmentation of color jittering, rotating, scaling, flipping and cropping. We adopt the Resnet-18 as the backbone CNN. Loss coefficients ω_1 , ω_2 and ω_3 are set to 1, 3 and 1. The selection of query length N follows LSTR [10] by evaluating from 20 to 80 in increments of 5. We trained the detection network on a single GTX 3090 for 300k iterations. The initial learning rate is set to 0.0001 and decayed ten times every 100k iterations. We adopt the Adam [8] optimizer and the batch size is set to 16. For the experiments conducted on the PS2.0 [19] dataset, the FFN layer of the transformer predicts two marking points for each output query.

Evaluation Metrics. For the LRPS dataset, the performance of parking slots detection is evaluated through the mean average precision (mAP) introduced in [20]. A predicted parking slot \hat{s}_i is considered as a true positive when it is matched with a ground truth parking slot s_j under the following conditions: $\hat{c}_i = c_i$, $IOU(\hat{s}_i, s_j) > \delta$. We adopt intersection over union (IOU) to measure the deviation between \hat{s}_i and s_j . δ is the corresponding IOU threshold. For the PS2.0 dataset, the performance is verified through euclidean distances between ground truth and detected marking points of entrance lines [19]. Specially, for a labeled ground truth $\{\hat{\mathbf{p}}_i^1, \hat{\mathbf{p}}_i^2\}$, a certain predicted parking slot $\{\mathbf{p}_j^1, \mathbf{p}_j^2\}$ is determined as a true positive when $L_2(\hat{\mathbf{p}}_i^t, \mathbf{p}_j^t) < \delta_l$, $t \in \{1, 2\}$. Here we adopt the square error function L_2 to obtain the eu-

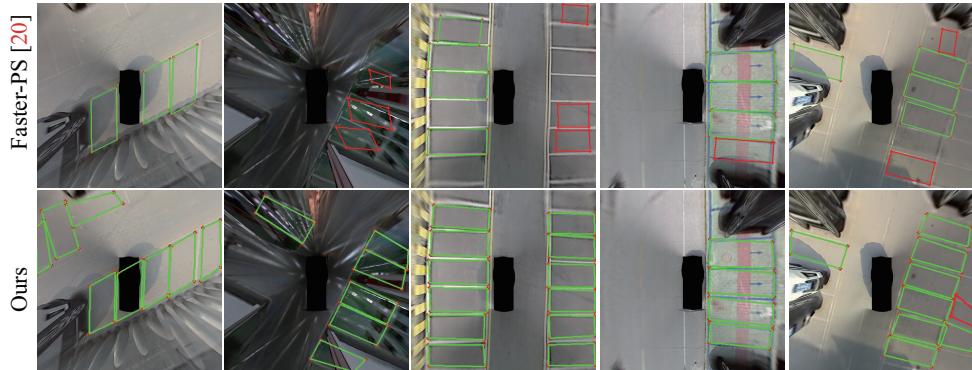


Figure 5: Comparative results on LRPS test set. Red lines represent the false positive and green lines are the true positive.

clidean distance and δ_l is the distance threshold. δ_l is set to 12 in the following experiments.

5.1 Comparision with the state-of-the-art method

Fig. 4 (a) shows the performance on the LRPS dataset. We adopt Faster-PS [20] as our baseline, which is a corner points-based parking slot detection method modified on a faster-rcnn structure [13]. From the results, our approach fully surpasses Faster-PS by 16.58% on mAP_{70} , 13.86% on mAP_{50} .

Fig. 4 (b) shows the performance of the PS2.0 dataset. OISS outperforms Faster-PS [20] by 5.67% on the recall and 0.09% on the precision. Comparing with current two-stage frameworks, OISS implemented in an end-to-end (one-stage) manner at the cost of comparatively lower precision and recall values. Furthermore, OISS is capable of directly predicting the entrance lines without any extra post-processing operations like filtering or matching in previous methods [8, 9, 10, 11].

Visualization results on the LRPS dataset are illustrated in Fig. 5. Our method performs more accurate predictions under different scenarios than Faster-PS, especially for clustered parking slots. We attribute this to (1) the OISS strategy makes our method more robust to the deformation of input images. Details will be claimed in the next section. (2) The transformer network can capture non-local dependencies to benefit detection in complex slot topology. *More examples are illustrated in supplementary materials.*

5.2 Ablation study

Effect of the OISS matching strategy. To verify the effect of OISS strategy, we compare it with the consistent ordering matching (COM) strategy proposed in [20]. The COM strategy makes the model regress the four corner points of each parking slot in a clockwise direction that starts from a specific point according to their angles with respect to the center of the parking slot. For the training of COM, we preprocess the label to fix the order. Comparing to OISS, the consequence of COM strategy is 2.55 % lower on mAP_{70} , 3.14 % lower on mAP_{50} . We also display the predicted results to better understand how OISS works. As is illustrated in Fig. 6, when the vehicle is steering, the OISS strategy can accurately predict

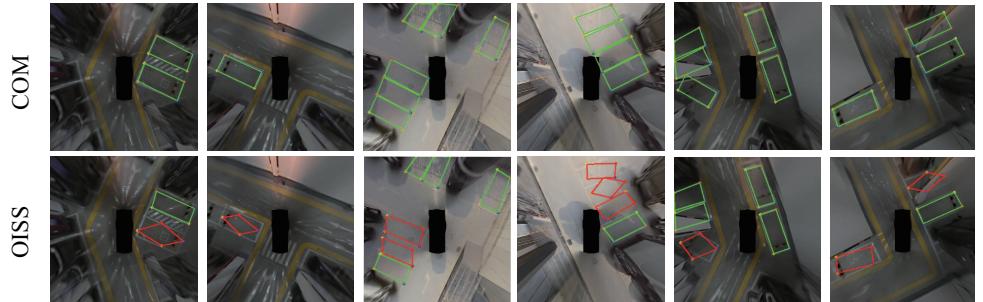


Figure 6: Predicted parking slots of OISS and COM strategies when vehicle is steering. Both of them are generated based on a transformer network.

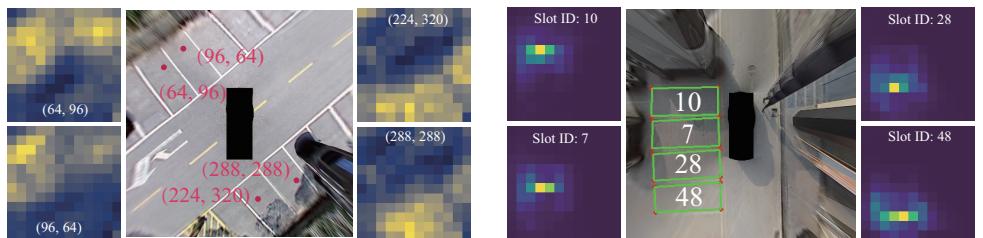


Figure 7: Attention maps in the transformer network. Left: encoder attention maps for different sampling points. Right: decoder attention maps between output slots and encoder feature.

the parking slots at some critical observation angles, while the COM strategy fails to predict the corner points between the ground truth.

Shape cost and center point prediction. The effect of shape cost and center point are also evaluated. Without the addition of shape cost, the prediction performance of OISS dropped by 1% of mAP_{70} , 1.5% of mAP_{50} . We visualize the predictions under some challenging shapes in Fig. 8 to better understand the shape cost. From Fig. 8, OISS strategy without shape cost sometimes performs poorly on inclined parking slots detection. Moreover, the addition of center point prediction also slightly promotes the performance by 1.32% of mAP_{70} and 2.54% of mAP_{50} , which demonstrates that the center prediction is beneficial to capture the global pattern of each parking slot.

Attention maps in the transformer network. Fig. 7 displays the intermediate attention maps in the transformer network. The left part visualizes the attention maps for several reference points in the last encoder stage. Two points on the upper part of the image are more interested to the nearby parking slots, thus making the encoder to capture the topological relations in the inclined area more effectively. A similar pattern is also exhibited for the remaining points at the bottom of the image. The right part demonstrates the attention maps in the cross-attention module of the last decoder stage. Each attention map interprets the relations between the encoder features and an output query slot. We notice that the attention is clearly reflected on the local regions of different parking slots. As a result, specific slots in the adjacent area are separated and distinguished.

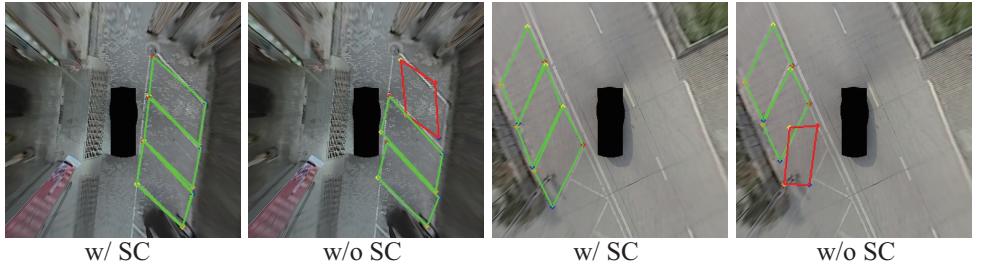


Figure 8: Effect of the shape cost (SC). By introducing the shape cost (SC), our method can predict the aligned shape of parking slots.

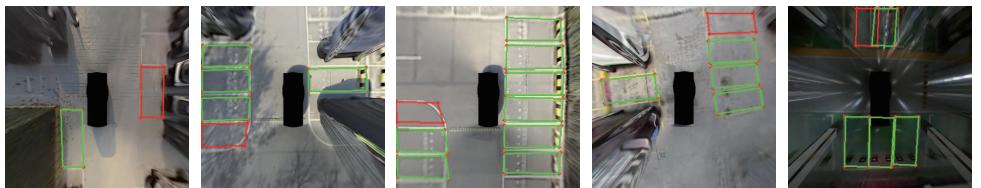


Figure 9: Failure cases under some special situations. The same color coding (see Fig. 5) is adopted.

Failure cases. Some failure cases are displayed in Fig. 9. Our OISS fails in predicting parking slots under unique situations. For example, the left three images show the OISS fails to identify the slot-like pattern in traffic scenarios. Such slot-like patterns are usually formed with the signs or lanes on the road. The remaining images show the prediction errors under some extremely adverse conditions, like abrasion (fourth) and underground (fifth).

6 Conclusion

In this work, we propose the order-independent matching with shape similarity and a large-scale remote-view parking slots towards accurate, generalized, and scale-able parking slot detection. The introduced matching strategy adopts a two-level pipeline that solves the order-induced rotated false detection and shape deformations by adaptive order association and holistic shape cost. Furthermore, the collected LRPS dataset demonstrates more diverse traffic scenarios and shapes of different parking slots, making it more possible and practical for the autonomous driving society to confront more generalized parking slot detection in the real world. In the future, we plan to add the marking points detection for further improving the localization performance.

7 Acknowledgements

This work was supported by the National Natural Science Foundation of China (61976170, 91648121).

References

- [1] Ali Boyali and Simon Thompson. Autonomous parking by successive convexification and compound state triggers. In *ITSC*, pages 1–8. IEEE, 2020.
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.
- [3] Juntao Chen, Lin Zhang, Ying Shen, Yong Ma, Shengjie Zhao, and Yicong Zhou. A study of parking-slot detection with the aid of pixel-level domain adaptation. In *ICME*, pages 1–6. IEEE, 2020.
- [4] Kazukuni Hamada, Zhencheng Hu, Mengyang Fan, and Hui Chen. Surround view based parking lot detection and tracking. In *Intelligent Vehicles Symposium*, pages 1106–1111. IEEE, 2015.
- [5] Junhao Huang, Lin Zhang, Ying Shen, Huijuan Zhang, Shengjie Zhao, and Yukai Yang. DMPR-PS: A novel approach for parking-slot detection using directional marking-point regression. In *ICME*, pages 212–217. IEEE, 2019.
- [6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.
- [7] Minchul Lee, Seokwon Kim, Wonteak Lim, and Myoungho Sunwoo. Probabilistic occupancy filter for parking slot marker detection in an autonomous parking system using AVM. *IEEE Trans. Intell. Transp. Syst.*, 20(6):2389–2394, 2019.
- [8] Soomok Lee, Daejin Hyeon, Gikwang Park, Il-joo Baek, Seong-Woo Kim, and Seung-Woo Seo. Directional-dbscan: Parking-slot detection using a clustering method in around-view monitoring system. In *Intelligent Vehicles Symposium*, pages 349–354. IEEE, 2016.
- [9] Linshen Li, Lin Zhang, Xiyuan Li, Xiao Liu, Ying Shen, and Lu Xiong. Vision-based parking-slot detection: A benchmark and a learning-based approach. In *ICME*, pages 649–654. IEEE Computer Society, 2017.
- [10] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *WACV*, pages 3693–3701. IEEE, 2021.
- [11] Chen Min, Jiaolong Xu, Liang Xiao, Dawei Zhao, Yiming Nie, and Bin Dai. Attentional graph neural network for parking-slot detection. *IEEE Robotics Autom. Lett.*, 6(2):3445–3450, 2021.
- [12] Tong Qin, Tongqing Chen, Yilun Chen, and Qing Su. AVP-SLAM: semantic visual mapping and localization for autonomous vehicles in the parking lot. In *IROS*, pages 5939–5945. IEEE, 2020.
- [13] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

- [14] Russell Stewart, Mykhaylo Andriluka, and Andrew Y. Ng. End-to-end people detection in crowded scenes. In *CVPR*, pages 2325–2333. IEEE Computer Society, 2016.
- [15] Jae Kyu Suhr and Ho Gi Jung. Sensor fusion-based vacant parking slot detection and tracking. *IEEE Trans. Intell. Transp. Syst.*, 15(1):21–36, 2014.
- [16] Jae Kyu Suhr and Ho Gi Jung. End-to-end trainable one-stage parking slot detection integrating global and local information. *CoRR*, abs/2003.02445, 2020.
- [17] Zizhang Wu, Weiwei Sun, Man Wang, Xiaoquan Wang, Lizhu Ding, and Fan Wang. Psdet: Efficient and universal parking slot detection. In *IV*, pages 290–297. IEEE, 2020.
- [18] Zhuoping Yu, Zhong Gao, Hansheng Chen, and Yuyao Huang. SPFCN: select and prune the fully convolutional networks for real-time parking slot detection. In *IV*, pages 445–450. IEEE, 2020.
- [19] Lin Zhang, Junhao Huang, Xiyuan Li, and Lu Xiong. Vision-based parking-slot detection: A dcnn-based approach and a large-scale benchmark dataset. *IEEE Trans. Image Process.*, 27(11):5350–5364, 2018.
- [20] Andrea Zinelli, Luigi Musto, and Fabio Pizzati. A deep-learning approach for parking slot detection on surround-view images. In *IV*, pages 683–688. IEEE, 2019.