KAM-Net: Keypoint-Aware and Keypoint-Matching Network for Vehicle Detection From 2-D Point Cloud

Tianpei Zou, Guang Chen, *Member, IEEE*, Zhijun Li, *Senior Member, IEEE*, Wei He, Sanqing Qu, Shangding Gu, and Alois Knoll, *Senior Member, IEEE*

***Abstract*—Two-dimesional (2-D) LiDAR is an efficient alternative sensor for vehicle detection, which is one of the most critical tasks in autonomous driving. Compared to the fully developed 3-D LiDAR vehicle detection, 2-D LiDAR vehicle detection has much room to improve. Most existing state-of-the-art works represent 2-D point clouds as pseudo-images and then perform detection with traditional object detectors on 2-D images. However, they ignore the sparse representation and geometric information of vehicles in the 2-D cloud points. To address these issues, in this article, we present a novel *keypoint-aware and keypoint-matching network* termed as KAM-Net, which focuses on better detecting the vehicles by explicitly capturing and extracting the sparse information of L-shape in 2-D LiDAR point clouds. The whole framework consists of two stages—namely, keypoint-aware stage and keypoint-matching stage. The keypoint-aware stage utilizes the heatmap and edge extraction module to simultaneously predict the position of L-shaped keypoints and inflection offset of L-shaped endpoints. The keypoint-matching stage is followed to group the keypoints and produce the oriented bounding boxes with axis by utilizing the endpoint-matching and L-shaped-matching methods. Further, we conduct extensive experiments on a recently released public dataset to evaluate the effectiveness of our approach. The results show that our KAM-Net achieves a new state-of-the-art performance. The source code is available at https://github.com/ispc-lab/KAM-Net.**

***Impact Statement*—This article is motivated by the problem of detecting and locating surrounding vehicles using 2-D LiDAR. Existing 2-D images-based and 3-D LiDAR-based methods of vehicle detection are less robust or too expensive. This article proposes a novel approach which is based on the cheaper but accurate 2-D LiDAR. In summary, our method can be considered as a process, which detects and groups the key-points of L-shape to produce the bounding boxes of vehicles via deep learning. The experimental results verify the effectiveness and robustness of the proposed approach in vehicle detection. For the future work, we will extend our KAM-Net to other object detection applications.**

***Index Terms*—Artificial intelligence algorithmic design and analysis, artificial intelligence in transportation, deep learning, supervised learning.**

I. Introduction

With the rapid development of autonomous driving, surrounding vehicle perception/detection attracts more and more attention, especially for collision avoidance [1], mobile parking robots [2], and safer autonomous driving [3]–[5]. Most existing vehicle perception methods are based on 3-D LiDAR point clouds [6]–[9] or RGB images [10]–[14]. Generally, 3-D LiDAR-based and 2-D image-based detectors are suitable for vehicle detection due to their strong capability of capturing 3-D structures or semantic information. However, the 3-D LiDAR is plagued by the expensive price and the 2-D image-based methods are not robust for environment illumination changing, which make them cannot fully satisfy the practical application. To alleviate above issues, some researchers attempt to utilize 2-D LiDAR as supplementation or alternative. The earlier presented methods leverage predesigned heuristics [15]–[20] while such hand-crafted features are less robust and easily confused. Recently, some deep learning networks based on pseudoimage have been proposed [21]–[23], and a specialized dataset and network for vehicle detection are proposed by Chen *et al.* [24] which achieves great success. However, these deep learning methods directly utilize the traditional image-based object detection network and ignore the differences between 2-D pseudoimages and 2-D images. 1) The 2-D pseudoimages only contain objects edge information but the traditional convolutional neural network (CNN) detectors pay more attention to the center of object. 2) The expressions of vehicles in 2-D pseudoimages are unified L-shape. Thus, the conventional anchor-based methods which directly predict the center and orientation angle are hard to interpret and have unsatisfactory performance. And the pointwise methods [25], [26] which are widely used in dense 3-D LiDAR point clouds are imprecise in the 2-D LiDAR detection task as proposed by Chen *et al.* [24].

To address the above challenges, we propose a 2-D point clouds-based method named keypoint-aware and keypoint-matching network (KAM-Net). Considering that the current

Manuscript received March 26, 2021; revised June 4, 2021 and July 12, 2021; accepted August 15, 2021. Date of publication September 16, 2021; date of current version March 24, 2022. This work was supported in part by Anhui Provincial Natural Science Foundation, Anhui Energy-Internet joint Program under Grant 2008085UD01 in part by the National Natural Science Foundation of China under Grant U1913601 and Grant 61906138, and in part by Shanghai Rising Star Program under Grant 21QC1400900. This paper was recommended for publication by Associate Editor Qinglai Wei upon evaluation of the reviewers' comments. *(Corresponding author: Guang Chen.)*

Tianpei Zou and Sanqing Qu are with the Department of Automotive Engineering, Tongji University, Shanghai 200092, China (e-mail: 2011459@tongji.edu.cn; 20114444@tongji.edu.cn).

Guang Chen and Shangding Gu are with the School of Automotive Studies, School of Automotive Studies, Tongji University, Shanghai 200092, China, and also with the Technical University of Munich, 80333 Munich, Germany (e-mail: guangchen@tongji.edu.cn; gshangd@163.com).

Zhijun Li is with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230071, China, and also with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: zjli@ieee.org).

Wei He is with the Institute of Artificial Intelligence and the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: weihe@ieee.org).

Alois Knoll is with the Technical University of Munich, 80333 Munich, Germany (e-mail: knoll@in.tum.de).

Digital Object Identifier 10.1109/TAI.2021.3112945

state-of-the-art 3-D vehicle detection algorithms, such as [6] and [27], tend to use "To BEV' '(to bird's eye view) or pillar method which is a method that transforms 3-D to 2-D, our network follows previous work applying CNN on the pseudoimages and achieves new state-of-the-art performance by fully exploring the L-shaped geometric information of pseudoimages of 2-D LiDAR point clouds.

Our KAM-Net is designed as a keypoint predicted anchor-free framework, which consists of a keypoint-aware stage and a keypoint-matching stage. Specifically, after the feature extractor, our keypoint-aware stage firstly learns to predict the keypoints and inflection offset of L-shape to harvest candidate vehicle bounding boxes. Following CornerNet [28], we utilize an Hourglass network as a feature extractor to extract descriptive features and leverage the prediction branches to predict heatmaps for keypoints. Parallel to the prediction branches, we adopt an inflection shift branch to explicitly capture and extract the sparse edge information through using deformable convolution with supervision. It also learns and predicts the inflection shift, which is the displacement vectors from endpoint to the corresponding inflection point. Although predicting the offsets of point clouds is widely used in 3-D point cloud prediction, here we present for the first time a method that explicitly captures and extracts the information of sparse 2-D point clouds. Moreover, different from normal keypoint predicted anchor-free prediction architectures, we propose an additional classification branch which integrates the edge information from the inflection shift branch and an adaptive keypoints selection strategy to model inference self-adaption. With the positions of keypoints and the inflection shifts, the keypoint-matching stage then adopts the spatial relationship of three keypoints to compose an oriented bounding box with axis. Nonmaximum suppression is finally adopted to remove duplicate predictions. We conduct extensive comparison and ablation studies on the recently released dataset [24], and the results show that our KAM-Net framework achieves a new state-of-the-art performance.

Our main contributions lie in the following.

1) We propose a *keypoint-aware and keypoint-matching network* termed KAM-Net, which can better capture the keypoints and edge information to achieve interpretable and reasonable offsets prediction in 2-D point cloud pseudoimages.
2) We provide a unique anchor-free oriented object detection architecture, which integrates an adaptive keypoints selection method and can avoid messy and challenging oriented bounding boxes regressions.

II. Related Work

A. Two-Dimensional Point Clouds-Based Vehicle Detection

Existing 2-D point clouds-based vehicle methods can be divided into two categories, nondeep learning methods and deep learning. The earlier present methods extract the hand-crafted feature from point clouds with conventional nondeep learning methods. Most of them leverage predesigned heuristics such as the L-shape of cars to detect vehicles [15]–[20] or the shape of legs for human detection [29]–[32]. Considering the occlusion problem, in [18], a weighted least-squares method is leverage to fit the line and right angle corner. In [19], the L-shape fitting is considered as an optimization problem, and an efficient search method is proposed to find the optimal solution. However, such hand-crafted features based methods are less robust and easily confused when objects are occluded or deformed.

With the development of deep learning, some works based on convolutional networks have been proposed [21]–[23] for some specific tasks. Beyer *et al*. [21] propose a depth preprocessing step and a voting scheme that significantly improves CNN performance. Beyer *et al*. [22] provides a small, fully online temporal window in the network to further boost detection performance. Guerrero-Higueras *et al.* [23] describe a tool named PeTra based on an off-line trained full CNN capable of tracking pairs of legs in a cluttered environment. Recently, a specialized model based on 2-D point clouds pseudoimage for vehicle detection has been devised by Chen *et al.* [24] and obtained a good result. The main method adds a cascade pyramid and a direction detection head on faster R-CNN [33] to better predict the oriented bounding box. The deep learning methods are more effective and robust through automatic diversity feature learning from vast amounts of annotation data instead of particular hand-crafted features. Although there have been several works to research 2-D point cloud detection based on pseudoimage with deep learning, most of them directly apply the traditional image-based object detection and ignore the difference between 2-D point cloud pseudoimage and vision-image, which makes them unsuitable for pseudoimage. Therefore, we propose our KAM-Net for vehicle detection in the 2-D point cloud pseudoimage.

B. Image-Based Deep Learning Object Detection

1) Anchor-Based Detectors: Anchor-based detectors try to predict the category confidence of existing objects in each preset anchor box and regress the box to match the object tightly. Anchor-based methods generally fall into two categories, two-stage and one-stage methods.

Derived from R-CNN [34]–[36], two-stage methods apply selective search [37] method to judge and regress the region of interest (RoI) candidates. Two of the most classic methods are faster-RCNN [38] and mask-RCNN [39]. The former work employs the RPN to obtain RoIs and focuses on the object detection, while the latter work applies RoIAlign instead of RoIPool and focuses on the segmentation. Compared with two-stage methods, one-stage methods predict the category confidence and regress the preset anchor boxes directly. YOLOv2 [40] improves YOLOv1 [41] in several aspects, such as batch normalization, high resolution classifier, and so on. SSD [42] extracts different scales of feature maps to detect different scales of objects. Focal loss [43] is also proposed to solve the remaining problem, the imbalance number between negative and positive samples.

2) Anchor-Free Detectors: Without preset anchor boxes, anchor-free detectors try to directly predict the center with four sides of a bounding box or the keypoints which can group bounding boxes.

The first type of detectors predicts the center. To some extent, YOLOv1 [41] belongs to this category due to its direct prediction of size and shape. Besides, DenseBox [44], UnitBox [45], FCOS [46] all belong to this type. Among them, FCOS [46] considers all the points in the ground truth as positive samples to cope with low recall, and it has become one of the state-of-the-art detectors. As the second type, detectors locate and group the keypoints of the bounding boxes. CornerNet [28] predicts the position of top left point and down right point of the target object including the embedding vector of the corresponding point to group them as a bounding box. CenterNet [47] adds a center detection into CornerNet. ExtremeNet [48] predicts and groups five keypoints. RepPoints [52] first proposes the deformable convolution with supervision to improve the performance.

Unlike most prior detection tasks where the target has a clear center, the 2-D point clouds pseudoimages are hard to extract effective knowledge of center from sparse points, which means the anchor-based approaches and the first type of anchor-free approaches are not adequate. The second anchor-free methods are also less leveraged in the oriented bounding box prediction. To cope with the above problems, we propose our anchor-free approach adjusted from the second anchor-free method. Furthermore, our approach is also extensible to more general detection tasks through its symbiotic architecture with existing object detectors.

III. Dataset Preliminaries

The training and testing of our model are conducted on the large-scale recent dataset proposed by [24]. As we predict the keypoints of the bounding boxes, the original dataset and annotations need to be adjusted to meet our approach.

A. Pseudoimage Preprocess

We convert 2-D point cloud frames into pseudoimages by projecting points onto the ground plane and resizing them to 512×512. To increase robustness, we also apply flip augmentation and rotation data augmentation. We mainly consult the preprocess method proposed by Chen [24].

B. Annotation Preprocess

To make original annotation suit our method, we pretreat the annotation without relabeling by defining I-point, D-point, and A-point. The term "I-point" refers to the inflection point of L-shape, which is the closest point to the self-vehicle in the ground truth box. The term "D-point" means the point in the same direction as the target vehicle axis. The "A-point" means another endpoint of L-shape. We predict the direction of the vehicle axis instead of the heading of the vehicle. Because if we know the vehicle axis, it is easy to predict the heading when the vehicle is in motion. The final ground truth of each instance includes I-point (x_i, y_i), A-point (x_a, y_a), and D-point (x_d, y_d) as shown in Fig. 1 and (x_c, y_c, w, h), which are the x and y axis coordinates of the center of the annotated vehicle bounding box, and the corresponding width and height, respectively.

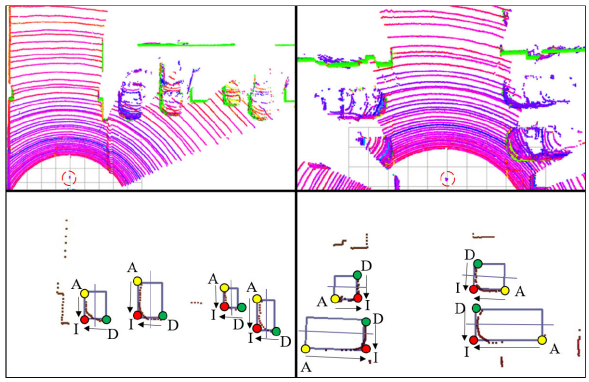


Fig. 1. Dataset visualization. Upper row: The visualization of 3-D cloud points; Lower row: The visualization of 2-D cloud points. The bounding boxes mean the ground truth. The lines in the bounding boxes are the target vehicle axes. The dotted circle is the self-car and the arrows are the inflection shifts. The "I," "D," and "A" mean I-point, D-point, and A-point which are additional ground truths.The lower row shows the corresponding 2-D cloud points zoomed in to achieve a better view, which causes the self-car to be invisible.

IV. Methods

As we mentioned before, the anchor-based and center predicted anchor-free approaches are unsuitable for this task since there is only edge information in the 2-D point clouds pseudoimage. To address this challenge, we present our keypoint predicted anchor-free method, KAM-Net, which can better capture edge information and has a reasonable architecture. The framework is presented in Fig. 2. As the framework shows, our KAM-Net mainly consists of two stages: keypoint-aware stage and keypoint-matching stage. The keypoint-aware stage predicts and classifies the keypoints and corresponding inflection shifts. The keypoint-matching stage proposes two methods to matching the keypoints and producing the final bounding boxes. In the following, we will present more details about our KAM-Net.

A. Keypoint-Aware Stage

The Keypoint-aware stage consists of four branches: endpoint prediction branch, inflection point prediction branch, inflection shift prediction branch, and endpoint classification branch. In this stage, the network learns to predict the locations of three keypoints and the inflection shifts of two endpoints. Then it leverages the extracting feature from inflection shifts prediction to adaptively classify and select the two endpoints for determining the orientation angle.

1) Endpoint Prediction Branch: We feed the pseudoimage feature map into the endpoint prediction branch to locate the endpoints. Since it is hard to distinguish the endpoints of L-shape directly, the endpoint prediction branch predicts the confidences of both A-points and D-points in one heatmap. The classification of them is in another branch. As we only predict vehicles, the heatmap has only one channel to represent the confidences of endpoints. Other settings are similar to CornerNet [28].

2) Inflection Point Prediction Branch: The inflection prediction branch is similar to the endpoint prediction branch. This branch predicts the confidences of inflection points, other sets are the same as the endpoint prediction branch.

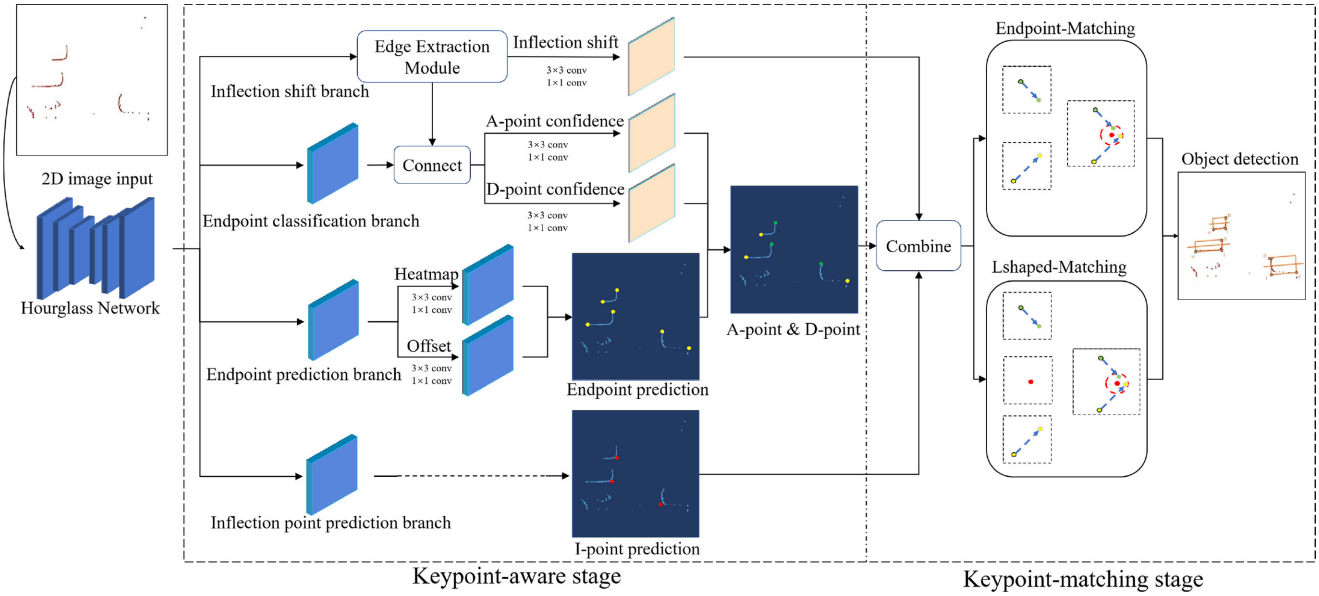


Fig. 2. KAM-Net architecture. Pseudoimage first feed into the Hourglass Network to extract feature maps. The keypoint-aware stage then predicts accurate keypoint locations and classifies the endpoints with inflection shifts of L-shape via four branches from feature maps. Finally the keypoint-matching stage matches the three keypoints to produce the bounding box by two proposed methods.

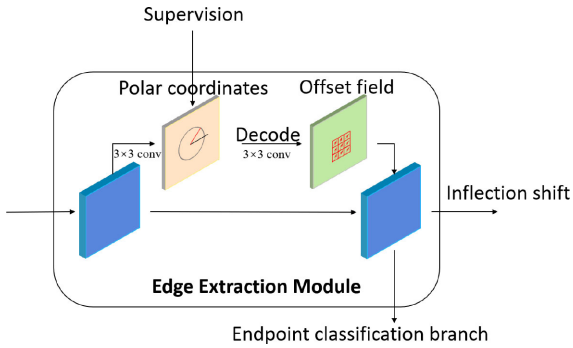


Fig. 3. Edge extraction module.

3) Inflection Shift Prediction Branch: The edge information of 2-D point cloud pseudoimage introduces challenges both on meaningful feature selection and the disturbance of similar objects. Inspired by the articles presented in [50]–[52], we proposed the inflection shift prediction branch to explicitly extract the edge information and output the inflection shift via deformable convolution with supervision. We will describe it in detail as follows.

Edge extraction module: Unlike the traditional image object detection, whose main information usually concentrates in the center of objects, the edge information in 2-D point clouds pseudoimage is uncertain and deformable. Thus, we construct an edge extraction module to explicitly capture the variable vehicle edges, as shown in Fig. 3. Inspired by the articles presented in [50]–[52], a deformable convolution with supervision is applied to capture the edge information. Specifically, we first feed the feature map into the 3×3 convolution layer to obtain the guiding map as f_1. Because of both positive and negative vectors existing, we design f_1 to predict the offset in the corresponding position with polar coordinates, which guides the deformable convolution to capture the edge information. We compute offsets between ground truth endpoints and inflection points and regard offsets as targets of f_1 to supervise in training

$$\alpha_{ai}^{i} = (\theta(A'^{i}, I'^{i}), d(A'^{i}, I'^{i}))$$
$$\alpha_{di}^{i} = (\theta(D'^{i}, I'^{i}), d(D'^{i}, I'^{i})) \quad (1)$$

where A'^{i}, D'^{i}, and I'^{i} means the A-point, D-point, and I-point in the feature map, respectively, $\theta(,)$ denotes the angle of the vector, $d(,)$ means the distance of two points in the feature map and α is the supervision shift. We train $\widehat{\alpha}$ applying smooth L1 loss with the ground truth offset α

$$L_{\alpha} = \frac{1}{N}\sum_{i=1}^{N}[SmoothL_1(\alpha_{ai}, \widehat{\alpha}_{ai}) + SmoothL_1(\alpha_{di}, \widehat{\alpha}_{di})]. \quad (2)$$

Then, we decoded the offset from polar coordinates to cartesian coordinates to fit the traditional deformable convolutional proposed in [50]. After decoded to cartesian coordinates, another convolution layer is used to obtain the offset of deformable convolution. As a result, the deformable convolution tries to capture and utilize the edge feature.

Inflectional shift: We feed the feature map obtained from edge extraction module into a new convolution layer to predict the inflection shift from endpoint to inflection point. For

$$oriented\_bbox^{i} = (Ix^{i}, Iy^{i}, Dx^{i}, Dy^{i}, Ax^{i}, Ay^{i})$$

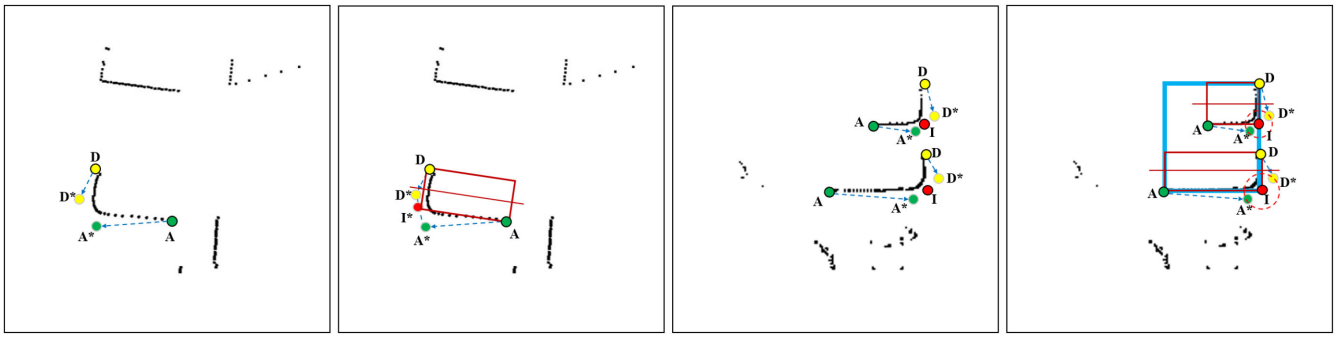


Fig. 4. Two matching methods. The "A," "D," and "I" mean A-point, D-point, and I-point. "A*" and "D*" mean assumed I-points (decode points) derived by "A" and "D". The boxes shown here have been projected to get rectangle. (a) First method, endpoint-matching, whose input is the location of two endpoints and corresponding inflection shifts. (b) Endpoint-matching allows the "A" and "D" to compose the bounding box alone if the decode point is close enough to each other. If so, we account the "I*" on the line which consists by "A*" and "D*". The confidences of "A" and "D" also influence the position of "I*" ("I*" should be close to the high confidence int). (c) Second matching method, L-shaped-matching, whose inputs are the location of A-point, D-point, and I-point and inflection shift. (d) We only show the box that can form the approximate right-angle. Here are three candidate boxes, and the biggest one will be screened out due to the decode points is far away from the I-point.

we define the ground truth inflection shift for endpoint in the pseudoimage as

$$\delta_{ai}^{i} = (\theta(A^i, I^i), d^*(A^i, I^i))$$
$$\delta_{di}^{i} = (\theta(D^i, I^i), d^*(D^i, I^i)) \tag{3}$$

where A^i, D^i, and I^i means the A-point, D-point, and I-point in the pseudoimage, respectively, δ consisted $\theta(,)$ function with normalization and $d^*(,)$ with $log()$ function for narrowing the range of shift, and A, D, and I means the A-point, D-point, and I-point on the pseudoimage, respectively. During training, we apply smooth L1 loss to approach ground truth shift

$$L_\delta = \frac{1}{N}\sum_{i=1}^{N}\left[SmoothL_1\left(\delta_{ai}, \widehat{\delta}_{ai}\right) + SmoothL_1\left(\delta_{di}, \widehat{\delta}_{di}\right)\right] \tag{4}$$

where $\widehat{\delta}$ denotes prediction.

4) Endpoint Classification Branch: As we mentioned before, it is hard to distinguish the endpoint directly. In this branch, we fuse the information of edge from the previous branch to adaptively select and classify the A-point and D-point. We first connect the output of deformable convolution to the original feature map. Then, we apply a convolution layer to classify each endpoint, which is similar to the heatmap convolution. The main difference between endpoint classification branch and normal keypoint predicted anchor-free method is that the latter predicts the fixed top-k numbers of different keypoints in inference. While in our method, we select fixed top-k endpoints number, as the two numbers of A-points and D-points are different, we only maintain the small one and reduce the big one. For example, we assume there are m A-points and n D-points in top-k endpoints. Next, only $Minimum(m, n)$ points of both A-points and D-points will be sent into the keypoint-matching stage.In this way, we can get a self-adaption model, which can maximize the advantages of anchor-free approach. We performed ablation experiments and demonstrated its effectiveness in Section V.

B. Keypoint-Matching Stage

By considering the locations of keypoints and the inflection shifts of endpoints predicted from the keypoint-aware stage, it is reasonable to aggregate all the information within the spatial relationship of L-shape to match and produce the oriented bounding box. In the keypoint-matching stage, we propose and fuse two complementary methods to obtain the bounding boxes: endpoint-matching matches the endpoints using their shifts and locations to get high recall under less stringent criterion, L-shaped-matching, which combines locations of three keypoints and shifts to get high precision even under high standards, is the supplement of the first methods.

1) Endpoint-Matching: We propose our first matching method, endpoint-matching, for high recall due to its lower matching condition. As shown in Fig. 4(a) and (b), endpoint-matching groups the endpoints to produce the oriented bounding box utilizing the locations of endpoints and corresponding inflection shifts as input. In this method, we first combine the locations and inflection shifts of endpoints ("A," "D") to generate assumed I-points which we present as "A*" and "D*". Then, it is intuitive and reasonable that two endpoints belonging to a same bounding box should have the related assumed I-points which are close to each other

$$d(D^*, A^*) \times \kappa < d(D, A) \tag{5}$$

where D and A represent the two different endpoints, κ indicates the severity of matching conditions.

If two endpoints satisfy the above inequality, we consider the A-point and D-point can march a bounding box. We screen out the combinations which do not satisfy the inequality and calculate the scores of the bounding box

$$S_B^i = \omega_i(S_A^i + S_D^i)/2 \tag{6}$$

$$\theta_i = log(\frac{d(A^i D^i)/\kappa}{d(A^{*i} D^{*i})}) \quad \omega_i = \theta_i / Max(\theta) \tag{7}$$

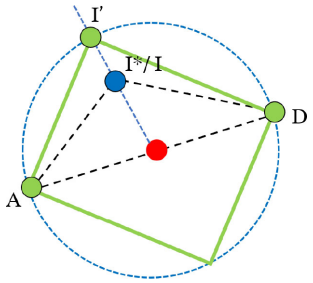


Fig. 5. Method of generating the bounding box. Once we match the points to form a triangle, we project the I-point to the circle which takes two endpoints as its diameter to form a right-angle bounding box. The rectangle is the final output.

where ω_i is the weight to introduce distance in, $Max()$ is the maximum, S_B, S_A and S_D is the score of the bounding box, A-point and D-point. Finally the I-point is inferred on the line consisted by "A*" or "D*." To improve the location quality of I-point we leverage the confidence of A-point and D-point as weight to decide which point the I-point prediction will close to

$$Loc_{I^*} = \frac{S_A Loc_{A^*} + S_D Loc_{D^*}}{S_A + S_D} \tag{8}$$

where Loc is the location of keypoint.

2) L-Shaped-Matching: The endpoint-matching obtain high recall due to the lower matching condition. However, the inflection shift prediction compared with the keypoint prediction is hard and inaccurate, which means it performs badly in strict criteria. We introduce the second method named L-shaped-matching as a supplement method which aims at high precision. The overview of this method is shown in Fig. 4(c) and (d). This method is inspired by following observations. 1) Three keypoints of the same bounding box form a right angle. 2) The assumed I-points should be close to the predicted I-point. Motivated by the above observations, we first calculate the vectors from two endpoint candidates to inflection point candidates with normalization presented as "AI" and "DI". Then we compute the angle of "AI" and "DI" as ν through vector dot products. Finally, the $\nu < \phi$ would be screened out. Here, ϕ is the lower bound of angles.

We also screen out the combinations if the assumed inflection points (A*, D*) are far from predicted inflection points obtained by heatmap and enhance the rest. The restrictions are more strict than the endpoint-matching, and the output uses I-points, A-points, and D-points, directly. That means for the same box, the predictions of L-shaped-matching and endpoint-matching are different, and the prediction of L-shaped-matching will get more significant scores than endpoint-matching generally. As we mentioned before, it is reasonable as the endpoint-matching usually cannot obtain the accurate location of I-point and L-shaped-matching can be a supplement.

3) Generating Bounding Box: However, after two matching methods, we usually cannot obtain a right triangle. To address this challenge, we maintain two endpoints but project the inflection point to the circle which takes two endpoints as its diameter to refine the original bounding box as shown in Fig. 5.

C. Optimization Objective

Points detection loss and local offset loss which use variant focal loss are defined as CornerNet [28]

$$L_{det} = \frac{-1}{N} \sum_{i=1}^{H} \sum_{j=1}^{W} \begin{cases} (1-p_{ij})^{\alpha} log(p_{ij}) & \text{if } y_{ij} = 1 \\ (1-y_{ij})^{\beta} (p_{ij})^{\alpha} log(1-p_{ij}) & \text{if } y_{ij} \neq 1 \end{cases} \tag{9}$$

$$L_{off} = \frac{1}{N} \sum_{k=1}^{N} SmoothL1Loss(o_{ij}, \widehat{o}_{ij}) \tag{10}$$

where N is the number of keypoints in a pseudoimage, and α and β are the hyper-parameters which we set 2 and 4 based on experience. And p_{ij} and o_{ij} are the score and location offset of keypoint at location (i, j) in predict heatmap and offsetmap. We use the two different ground truths of endpoint and inflection point to calculate the two different L_{edet}, L_{idet}, L_{eoff}, and L_{ioff}, respectively, and leverage them to supervise the two prediction branches distinguishing the endpoint and inflection point. The classifier of endpoints also leverages the same detection loss L_{det} as L_{cs}. Stated thus, total training loss is obtained by summing the above loss items

$$L_{total} = L_{edet} + L_{eoff} + L_{idet} + L_{ioff} + \beta L_{\alpha} + \gamma L_{\delta} + L_{cs} \tag{11}$$

where we set $\beta = 0.05$ and $\gamma = 0.5$. Following CornerNet [28], we add intermediate training and supervision, but we only choose the second prediction as output.

V. Experiments and Evaluations

In this section, we conduct extensive experiments on a recently released 2-D point clouds benchmark dataset [24] to evaluate the performance of our proposed KAM-Net.

A. Experimental Setup

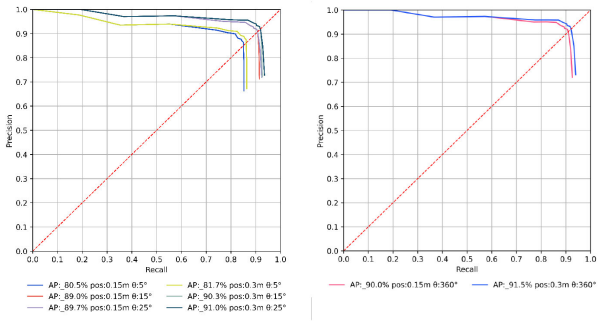
1) Dataset: In [24], there are 10 522 annotated frames of point clouds scans captured from nine types of scenes, including campus road, merging into traffic, T-junction, expressway, country road, crossroad, exiting traffic, parking lots, and underground garage. To permit a fair comparison, we follow the article presented in [24] to divide the dataset into `train`, `validation`, and `test` parts, each with 3604, 212, and 424 scans, respectively. For ablation studies and comparing with other approaches, we train on the training set and report results on the test set.

2) Evaluation Metric: Following the article presented in [24], our experiments and evaluations are conducted with the same metrics that the predictions are evaluated by combining different kinds of vehicle center location and axis direction angle errors. Only when a prediction satisfies both location and angle errors, it will be considered accurate. We apply these metrics instead of IoU to calculate the AP and introduce the AP@$d\&\theta$ to represent the AP under $d\&\theta$ criterion where d is the center location error and θ is the axis direction angle error. The unit of distance is meter and the unit of angle is degree.

TABLE I
OBJECT DETECTION PERFORMANCE COMPARISON

Model	AP@0.15(%)	AP@0.3(%)	AP@0.15&5(%)	AP@0.15&15(%)	AP@0.3&5(%)	AP@0.3&15(%)
Cascade Pyramid RCNN [24]	81.2	89.8	70.6	79.4	76.4	88.2
Faster RCNN [33]	80.6	89.4	-[1]	-	-	-
SSD [42]	74.4	81.6	-	-	-	-
Retinanet [43]	82.0	90.9	-	-	-	-
Hybrid Resnet Lite [24]	60.5	78.9	40.5	53.4	68.2	76.3
CornerNet [28]	84.3	87.4	-	-	-	-
KAM-Net	**90.0**	**91.5**	**80.5**	**89.0**	**81.7**	**90.3**

[1]We do not use angle error criterion because original baseline models like faster RCNN, Retinanet, and CornerNet are not capable of predicting box orientations.
The significance of bold entities in Table I is to emphasize which method in each column (i.e. evaluation indicator) achieves the best performance.



(a) (b)

Fig. 6. (a) PR curves of different center positioning and vehicle axis direction error criterion. (b) PR curves of different center positioning error criterion only, without vehicle axis direction.

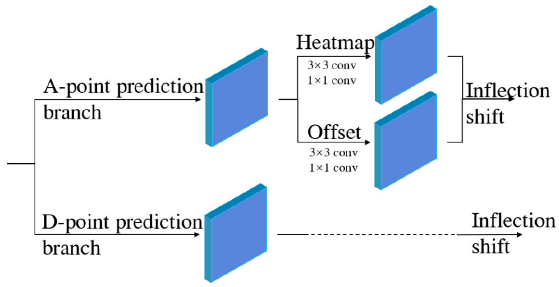


Fig. 7. Method of generating the bounding box. Once we march the points to form a triangle, we project the I-point to the circle which takes two endpoints as its diameter to form a right-angle bounding box.

3) Implementation Details: In the matching part, we set κ to 4, ϕ to 80°. In the training process, our model is trained on an 8GB RTX 2070 GPU with a batch size of 2, and the model is trained with stochastic gradient descent (SGD) for 30 epochs with the initial learning rate of 0.01. The learning rate is decayed after every 5 epochs, and the weight decay and momentum are 0.0001 and 0.9. Four branches are both trained in an end-to-end training neural network, and no other special training strategies are used. During training, we apply the data augmentations mentioned before, and training this detection network on one RTX 2070 GPU takes about 20 h.

During testing, our KAM-Net firstly predicts the locations of keypoints and inflection shifts. Here we only select $top50$ scored endpoints and $top25$ inflection points, then classify the endpoints with the edge information. Finally, we perform matching operations and apply soft-NMS with 0.4 IOU.

B. Method Comparisons

Our precision-recall curves under different center positioning and axis direction error criterion are shown as Fig. 6(a). The precision-recall curves under different center positioning error criterion only are also shown in Fig. 6(b). We report the results of vehicle detection on the `test` set of [24], as presented in Table I. As given in Table I, our KAM-Net achieves around AP@0.15&15 of 89.0%. Compared to the state-of-the-art anchor-based model proposed by Chen *et al.* [24], our KAM-Net achieves a 2% AP@0.3&15 improvement and a significant improvement (10%) under more stringent evaluation criteria. As for inference time, KAM-Net takes about 58 ms per frame in 2080 ti, i.e., above 17 Hz, which is little slow than cascade pyramid RCNN [24] but still acceptable and real-time.

C. Ablation Study

The ablation studies are given in Table II to exam the validity of our modules, and try putting forward as convincing explanations as possible.

1) Effectiveness of Matching Methods: To analyze the contribution of two matching methods, we exam two models that generate bounding boxes by one matching method. As a result, the endpoint-matching model performs better under low criterion, while the L-shaped matching performs better under rigid criterion. The reason is that as we mentioned before, the I-points inferred by shifts in endpoint-matching are inaccurate. The L-shaped-matching method has high precision because it uses the output I-point of the inflection prediction branch as the input.

2) Effectiveness of Endpoint Discrimination: The endpoint discrimination can be removed for the task without orientation estimation. Here, we try to verify the effectiveness of our architecture by comparing with the architecture which fuses the endpoints classification into the endpoint prediction branch. Specifically, we set up an architecture fusing two branches as shown in Fig. 7, each branch predicts the position and inflection shift of different kinds of endpoints. The AP decreases 8% and the decline is more severe in rigid criterion even after refinement.

TABLE II
SUMMARY OF EXPERIMENT AND ABLATION STUDIES

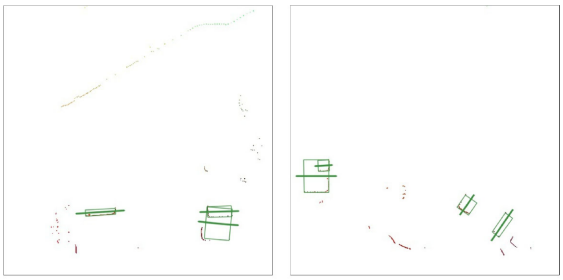
Model	Coordinates	Matching	Endpoint classification	Inflection Shift	AP@0.15(%)	AP@0.15&5(%)	AP@0.15&15(%)
KAM-Net	Polar	Fusion[1]	Classification branch	✓	**90.0**	80.5	**89.0**
KAM-Net	Polar	Endpoint-Matching[2]	Classification branch	✓	89.6	80.2	88.6
KAM-Net	Polar	Lshaped-Matching[3]	Classification branch	✓	87.2	**81.0**	85.9
KAM-Net	Polar	Fusion	-[5]	✓	86.0	69.1	81.0
CornerNet	-	Lshaped-Matching†[4]	Classification branch	-	86.5	74.5	80.0

[1,2,3]Represent the matching method in the inference module.
[4]It means the matching method is similar to the L-shaped-matching, but different.
[5]It means the endpoint classification branch integrates into the prediction branch.
The significance of bold entities in Table II is to emphasize which method in each column (i.e. evaluation indicator) achieves the best performance.

TABLE III
ABLATION RESULTS FOR MATCHING WITH DIFFERENT κ AND ϕ

Method	AP@0.15(%)	AP@0.3(%)	AP@0.15&5(%)	AP@0.15&15(%)	AP@0.3&5(%)	AP@0.3&15(%)
$\kappa = 1$	89.1	90.8	79.3	87.9	80.3	89.2
$\kappa = 2$	89.9	91.4	80.2	88.7	81.3	90.1
$\kappa = 3$	90.1	91.8	80.4	88.8	81.5	90.1
$\kappa = 4$	90.0	91.5	80.5	89.0	81.7	90.3
$\kappa = 5$	90.2	91.4	80.9	89.1	81.7	90.1
$\kappa = 6$	89.8	90.9	81.4	89.3	82.2	90.3
$\kappa = 7$	89.8	91.2	80.9	88.8	81.7	89.9
$\phi = 75°$	88.9	90.4	80.2	87.9	81.4	89.0
$\phi = 80°$	90.0	91.5	80.5	89.0	81.7	90.1
$\phi = 85°$	89.0	90.7	80.1	88.0	81.5	89.4

Setting $\phi = 80°$ when we experiment κ, and setting $\kappa = 4$ when we experiment ϕ.



(a) (b)

Fig. 8. Two wrong detection results by CornerNet [28]. There are some wrong matching pairs and cannot be restrained by NMS. The model may wrongly combine the points of two similar and close objects.

That is because due to removing our classification branch, the self-adaption approach also is removed. Meanwhile, through our observations, there are also some incorrect classifications for the endpoints without edge information from deformable convolution.

3) Deformable Convolution and Inflection Shift: To verify the effectiveness of our inflection shift, we rewrite the code of CornerNet [28]. To make it suit this task to predict the oriented bounding box, we add a module to inflection point with an embedding vector. As CornerNet [28] does, we use "push" and "pull" loss to make sure the three points which belong to the same bounding box have similar embedding vectors. We also apply the endpoint classification branch without deformable convolution. Without inflection shift means that the endpoint-matching method can't be used. As given in Table II, our method based on inflection shift performs better than CornerNet [28]. By observing the output, we also find CornerNet [28] generates a few false corner pairs because of similar embedding vectors caused by similar appearance as shown in Fig. 8.

4) Ablation on the Parameters κ and ϕ: We use different values of k in the keypoint-matching stage. The performance varies as given in Table III. A big k means strict matching condition. Matching with such a k may miss many potential instances, but some slightly larger values perform better at strict criteria. On the contrary, a small k increases the risk of matching wrong keypoints into one, but some slightly smaller values perform better at loose criteria. We empirically set k to 4. We also test different values of ϕ and set ϕ to 80°.

D. Qualitative Results

We present our prediction result as shown in Fig. 9 with oriented bounding boxes and vehicle axes. The points are the encoded pixels from 2-D point clouds, the boxes with lines in the upper row are ground truth and the boxes with lines in the middle and lower row mean prediction bounding box. It is apparent that most prediction bounding boxes fit the targets well, and our model can classify other confusing objects.

As shown in Fig. 9, compared to our model, the main disadvantage of the anchor-based model is that it tries to find a

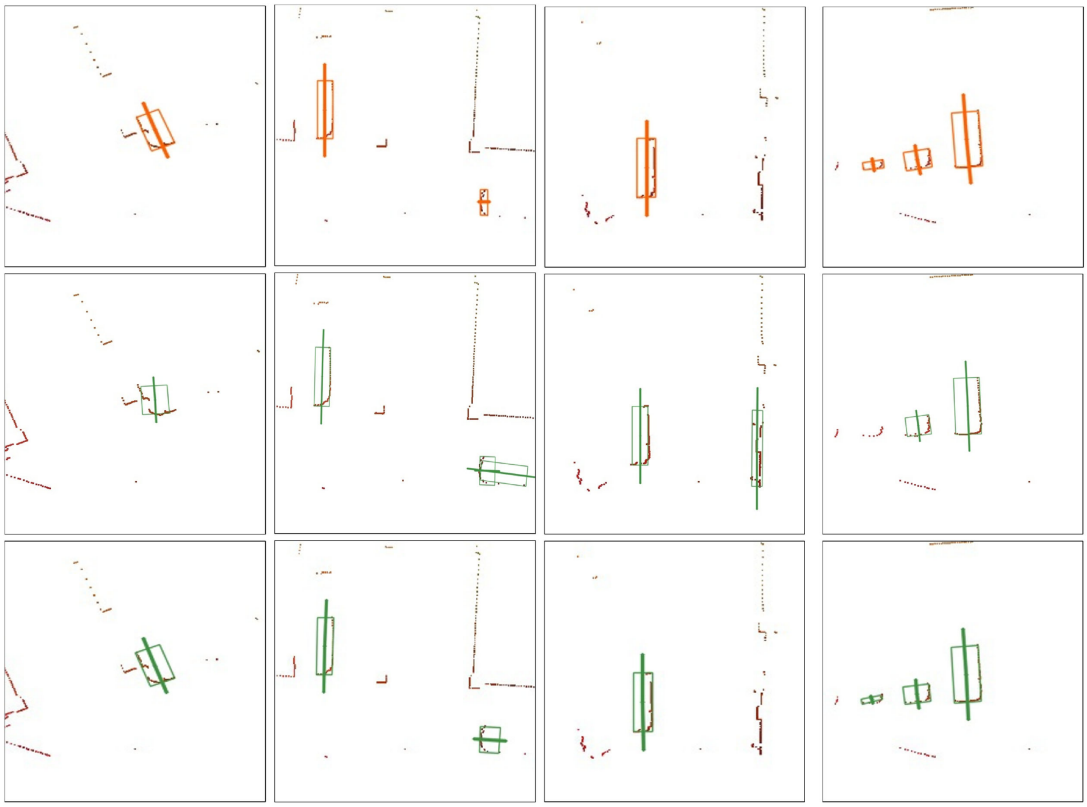


Fig. 9. Detection results. Upper row: Ground truth boxes in yellow. Middle row: Predicted bounding boxes generated by cascade pyramid RCNN in green. Lower row: Predicted bounding boxes generated by our model in green. Compared with cascade pyramid RCNN, our model predict are more accurate.

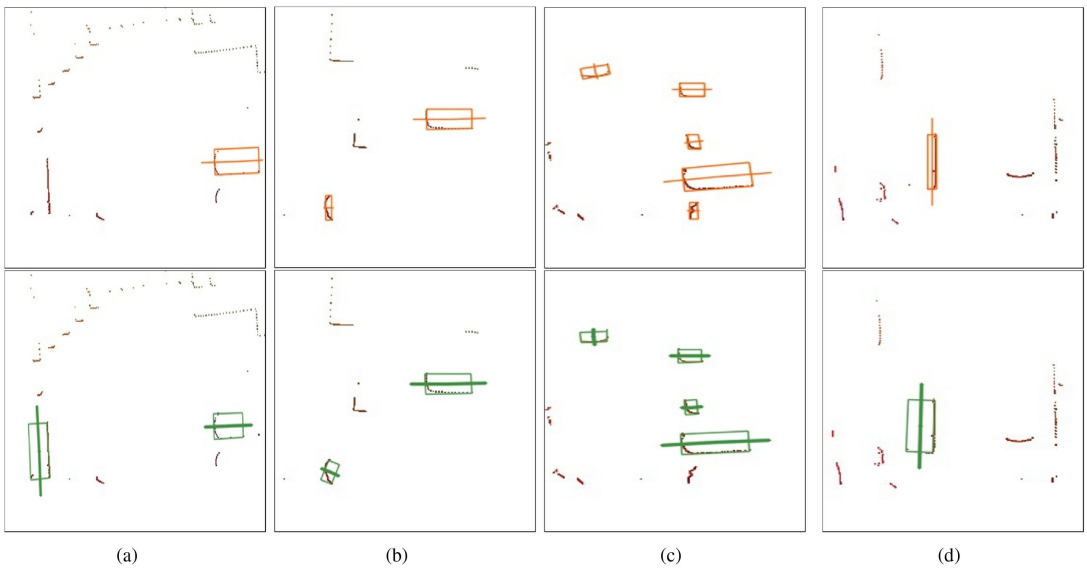


Fig. 10. Detection failure cases demonstrated. Upper row: Ground truth boxes in yellow. Lower row: Predicted bounding boxes generated by our model in green. (a) False positive. (b) Missing detection. (c) Missing detection. (d) Wrong detection.

center which is hard to get useful information. Our experiment also evaluates that the more challenging the evaluation standard becomes, the more remarkable improvement we will get. It is apparent from this Table I that our model obtains a more accurate location of center. The prediction process is more "visual," reasonable, and credible. Because we apply the inflection shift and rational inference process instead of embedding vector, we can achieve oriented bounding box prediction which is less achieved

in anchor-free approaches . It also overcomes the problems that anchor-free textcolorredapproaches, such as CornerNet [28], do not perform well when the similar objects of the same category are concentrated. The messy regression problem in the oriented bounding box detectors is also be avoided. Besides, our architecture allows the model to adjust the number of bounding boxes that can make full use of the anchor-free approach.

However, in some minor cases, the model predicts false positives on some L-shape confusing objects and false negatives on some blurry edges. As shown in Fig. 10, the missing detection may be the biggest problem due to our theory design. When the speed of vehicle reaches a level or the LiDAR scanning rate is not enough, the L-shaped would be distorted, which may cause our model to miss some detections. Nevertheless, as we can see, most of the predictions are still right and tight.

VI. Conclusion

In this article, we propose the *KAM-Net*, an anchor-free based method for vehicle detection from 2-D LiDAR point clouds pseudoimages. Our KAM-Net can explicitly capture the keypoint and edge information from hollow 2-D pseudoimages and achieve more interpretive and robust detection results compared to existing anchor-based methods. To avoid messy and challenging oriented bounding boxes regression existed in current methods, we integrate an adaptive candidate keypoints selection strategy to our KAM-Net. Extensive experiments on a recently released public benchmark demonstrate our KAM-Net superiority over current state-of-the-art methods. This method can theoretically be popularized to other similar detection works of oriented bounding box, such as scene text detection which has a clear direction. For future work, we will also explore the possibility of utilizing the multiview-based parameter free approach proposed in [53] for vehicle detection from groups of feature points.

References

[1] A. Balachandran, M. Brown, S. M. Erlien, and J. C. Gerdes, "Predictive haptic feedback for obstacle avoidance based on model predictive control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 26–31, Jan. 2015.

[2] S. Ma, H. Jiang, M. Han, J. Xie, and C. Li, "Research on automatic parking systems based on parking scene recognition," *IEEE Access*, vol. 5, pp. 21901–21917, 2017, doi: 10.1109/ACCESS.2017.2760201.

[3] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transp. Res. Part C:, Emerg. Technol.*, vol. 40, pp. 1–13, 2014.

[4] G. Chen, H. Cao, J. Conradt, H. Tang, F. Röhrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[5] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[6] S. Shi *et al.*, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10529–10538.

[7] S. Shi, X. Wang, and H. Li, "PointECNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[8] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1951–1960.

[9] T. Mimuro, N. Taniguchi, and H. Takanashi, "Concept study of a self-localization system for snow-covered roads using a four-layer laser scanner," *Automot. Innov.*, vol. 2, no. 2, pp. 110–120, 2019.

[10] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7074–7082.

[11] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8445–8453.

[12] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1019–1028.

[13] G. Chen *et al.*, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Trans. Syst., Man, Cybern. Syst.*, pp. 1–18, 2020, doi: 10.1109/TSMC.2020.3005231.

[14] G. Chen, C. Kai, Z. Lijun, Z. Liming, and A. Knoll, "Vcanet: Vanishing-point-guided context-aware network for small road object detection," *Automot. Innov.*, pp. 1–13, 2021.

[15] J. Schlichenmaier, F. Roos, M. Kunert, and C. Waldschmidt, "Adaptive clustering for contour estimation of vehicles for high-resolution radar," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2016, pp. 1–4.

[16] J. Schlichenmaier, N. Selvaraj, M. Stolz, and C. Waldschmidt, "Template matching for radar-based orientation and position estimation in automotive scenarios," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2017, pp. 95–98.

[17] X. Shen, S. Pendleton, and M. H. Ang, "Efficient L-shape fitting of laser scanner data for vehicle pose estimation," in *Proc. IEEE 7th Int. Conf. Cybern. Intell. Syst., IEEE Conf. Robot., Autom. Mechatronics*, 2015, pp. 173–178.

[18] R. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 301–306.

[19] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient L-shape fitting for vehicle detection using laser scanners," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 54–59.

[20] S. Qu *et al.*, "An efficient L-shape fitting method for vehicle pose detection with 2D LiDAR," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2018, pp. 1159–1164.

[21] L. Beyer, A. Hermans, and B. Leibe, "DROW: Real-time deep learning-based wheelchair detection in 2-D range data," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 585–592, Apr. 2017.

[22] L. Beyer, A. Hermans, T. Linder, K. O. Arras, and B. Leibe, "Deep person detection in two-dimensional range data," *IEEE Robo. Auto. Lett.*, vol. 3, no. 3, pp. 2726–2733, 2018.

[23] Á. M. Guerrero-Higueras *et al.*, "Tracking people in a mobile robot from 2D LIDAR scans using full convolutional neural networks for security in cluttered environments," *Front. Neurorobot.*, vol. 12, Jan. 2019.

[24] G. Chen *et al.*, "Pseudo-image and sparse points: Vehicle detection with 2dlidar revisited by deep learning-based methods," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, 2020, doi: 10.1109/TITS.2020.3007631.

[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.

[28] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1808.01244

[29] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2D range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3402–3407.

[30] L. Spinello and R. Siegwart, "Human detection using multimodal and multidimensional features," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3264–3269.

[31] C. Weinrich, T. Wengefeld, C. Schroeter, and H.-M. Gross, "People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features," in *Proc. 23rd IEEE Int. Symp. Robot Hum. Interactive Commun.*, Edinburgh, U.K., 2014, pp. 767–773.

[32] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2D laser scanners," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 726–733.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[34] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2014, pp. 580–587.
[37] J. R. Uijlings, K. E. V. D. Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[38] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497
[39] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870
[40] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242
[41] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640
[42] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[44] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*, [Online]. Available: http://arxiv.org/abs/1509.04874.
[45] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 516–520.
[46] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.
[47] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1904.08189
[48] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1901.08043
[49] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," CoRR, vol. abs/1904.11490, 2019. [Online]. Available: http://arxiv.org/abs/1904.11490
[50] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
[51] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.
[52] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9657–9666.
[53] X. Li, M. Chen, F. Nie, and Q. Wang, "A multiview-based parameter free framework for group detection," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4147–4153.