DOI 10.1007/s10946-023-10173-3
Journal of Russian Laser Research, Volume 44, Number 6, November, 2023

A SMALL-SIZE 3D OBJECT DETECTION NETWORK FOR ANALYZING THE SPARSITY OF RAW LIDAR POINT CLOUD

Bo Zhang,[1] Haosen Wang,[1] Suilian You,[1] Chengfeng Bao,[1] Shifeng Wang,[1,2*] Bo Lu,[2] and Jonghyuk Kim[2]

[1] *School of Optoelectronic Engineering, Changchun University of Science and Technology Changchun 130022, China*

[2] *Zhongshan Institute of Changchun University of Science and Technology Zhongshan 528400, China*

Corresponding author e-mail: sf.wang@cust.edu.cn

Abstract

LiDAR is one of the main sensors for 3D object detection in autonomous driving. LiDAR has the advantages of high precision and high resolution, but as the distance increases, the points it acquires become sparse, resulting in uneven sampling points and hindering the feature extraction of discrete objects. Current 3D object detection methods using LiDAR ignore the sparse features of the original LiDAR point cloud, resulting in low classification accuracy over small object detection, hindering the development of autonomous driving technology. To address this problem, we propose point cloud Sparse Detection Network (PCSD), an end-to-end two-stage 3D object detection framework. First, PCSD uses a data augmentation algorithm to preprocess the KITTI dataset, then uses voxel point centroids to locate voxel features, and then uses a point sparsity-aware RoI grid pooling module to aggregate local voxel features. Finally, we improve the confidence of the final bounding box by using voxel features with the original point cloud sparse features. Experimental evaluation on the challenging KITTI object detection benchmark shows significant improvements, especially in pedestrian and cyclist classification accuracy improved by 13.22% and 9.33%, respectively, demonstrating the feasibility and applicability of our work.

Keywords: LiDAR Point Cloud, 3D object detection, data augmentation, point cloud sparsity.

1. Introduction

LiDAR for 3D object detection has become one of the most popular sensors, 3D object detection, and the task of localizing and classifying objects in LiDAR point cloud. However, the point cloud data obtained by LiDAR is proportional to the distance, and the LiDAR point cloud data of the complete object cannot be scanned due to the occlusion of the object. Therefore, for objects closer to the LiDAR, the scanned point cloud data is denser, while for objects farther away from the LiDAR, the scanned point cloud data is sparser.

Voxel-based methods rely on the quantitative representation of LiDAR point cloud and ignore the sparsity of point cloud when processing LiDAR point cloud data. Other methods employ farthest point sampling (FPS) to attenuate changes in point cloud sparsity. While effective at sampling locations when the LiDAR point cloud distribution is uneven, it performs poorly on large-scale LiDAR point cloud, increasing processing time, and reduce the number of key points in the LiDAR point cloud used in the next stage of proposal refinement.

Manuscript submitted by the authors in English on October 18, 2023.

At the same time, the sparsity of the original LiDAR point cloud will also affect the detection accuracy of smaller objects. Since they have a smaller contact area with the LiDAR beam, they are less efficient at locating objects. Current methods largely ignore the sparsity of original LiDAR point cloud and mainly focus on the detection accuracy of vehicles, resulting in low accuracy in detecting small objects. However, improving the accuracy of pedestrians and cyclists is critical to enhancing road safety, as autonomous driving technology develops.

Therefore, we propose a network for small object detection to address the low accuracy of current 3D detectors for pedestrians and cyclists by exploiting centroid localization and feature encoding of voxel points. These encoded features directly reflect the LiDAR point cloud sparsity of multiple classes of detected objects. In summary, our contributions include:

1. We propose an occlusion-resistant data augmentation algorithm (OA) developed specifically for 3D LiDAR point cloud, which simulates occluded objects, reconstructs intact objects, and flexibly locates and rotates objects, thereby increasing the diversity of full-body target construction.

2. We find that original LiDAR point cloud sparsity affects the detection of smaller objects such as pedestrians and cyclists, so we propose a point cloud sparsity analysis module (PA) to sense the sparsity of original point cloud and use sparse point features encoding additional features to refine the next stage of bounding box regression. PA includes voxel point centroid positioning and point sparse analysis RoI grid pooling.

3. Our proposed PCSD network achieves better performance for pedestrians and cyclists on the KITTI data set.

2. Related Works

Voxel-based LiDAR 3D object detection involves partitioning the LiDAR point cloud into voxel grids and applying 3D and 2D convolutions directly to generate predictions. VoxelNet [1] introduced a VFE voxel feature encoding network, which efficiently extracts voxel features. Second [2] introduced 3D sparse convolution as a replacement for the original 3D convolution and proposed a novel angle regression method. CenterPoint [3] utilizes VoxelNet or PointPillars as backbone networks to characterize the input LiDAR point cloud. Voxel-rcnn [4] introduces voxel RoI pooling, which aggregates voxel features to generate RoI features. On the other hand, the Voxel Transformer [5] is the first to use Voxel-based Transformer Backbone, which can be flexibly inserted into other networks to realize functions. In 3D Cascade RCNN [6], in order to pursue higher object detection accuracy, multiple detection heads are connected in series to form a network. However, the voxel-based method offers significantly faster processing speeds compared to the point-based 3D object detection scheme. Therefore, voxel-based 3D object detection methods are more conducive to deploying LiDAR in real car.

Point-based LiDAR 3D object detection methods generally detect sampling point by point. A widely adopted technique in point-based detectors is the farthest point sampling (FPS) introduced in PointNet++. FPS selects points sequentially from the original point set, ensuring comprehensive coverage. In PointRCNN [7], FPS is employed to progressively down-sample the input LiDAR point cloud and generate 3D proposals from the down-sampled points. Furthermore, STD [8] introduces PointsPool as a method for extracting Region of Interest (RoI) features, while 3DSSD [9] employs a novel sampling stra-

tegy on the original LiDAR point cloud. Although point-based methods require a lot of time for LiDAR point cloud sampling and grouping, they exhibit higher accuracy compared to voxel-based methods.

3. Methodology

Our PCSD network uses the same voxel backbone as Second, the input raw LiDAR point cloud data is first subjected to Occlusion-resistant Data Augmentation and voxelised, using 3D sparse convolution, and then passed through the Region Proposal Network (RPN) to generate initial bounding box proposals. Then, the second stage further refines the bounding box by voxel features and point cloud sparse features. In Fig. 1, we show an overview of the PCSD framework.

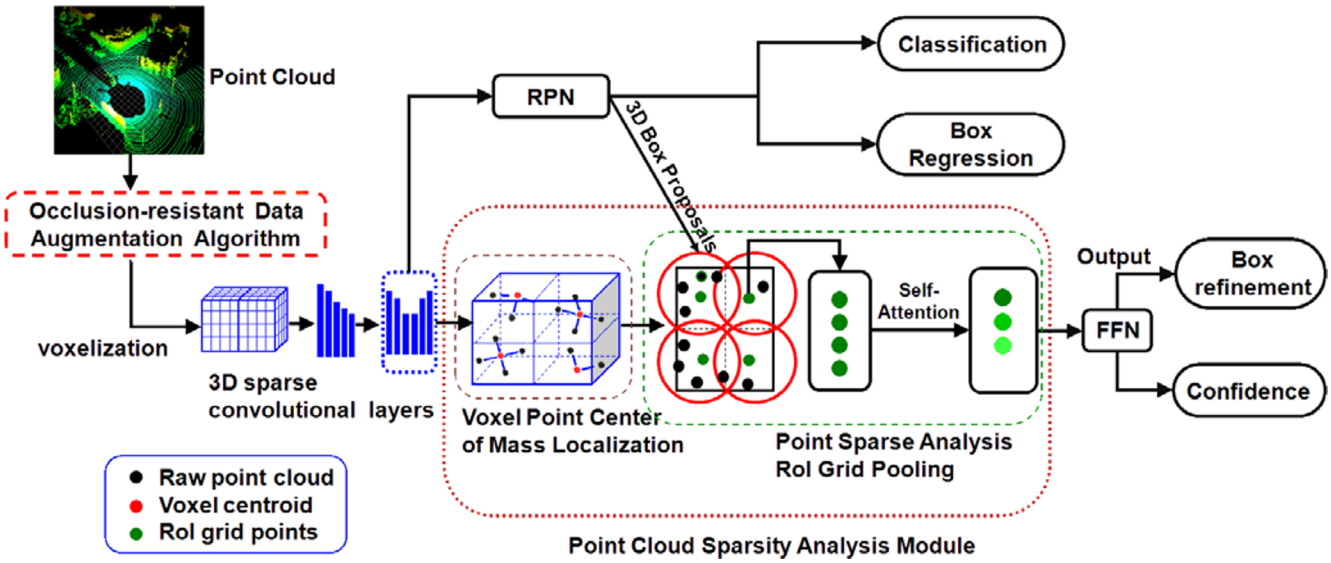


Fig. 1. PCSD framework. Here, the orange dashed box represents the voxel point center-of-mass localization section, and the green dashed box represents point sparse analysis RoI Grid Pooling. Also, FFN is a feed-forward network.

3.1. Point Cloud Data Augmentation

We propose an Occlusion-resistant Data Augmentation Algorithm (OA) to improve 3D object detectors. It is notoriously difficult for conventional 3D detectors to detect objects when they are occluded. Therefore, in order to reduce the negative impact of object occlusion, we use the OA point cloud data enhancement method to simulate the occluded object, and then apply the HPR-based method to flexibly construct complete objects and randomly position and rotate the object to increase the diversity of the object.

Hidden Point Removal (HPR): M_i is a group of random points, and we determine whether each of these points $M(i,t)$ is observable at LiDAR viewpoint C or not. The HPR method is to set C as the sphere at the center of the sphere, with radius R, and large enough. Using a spherical flip, each point $M(i,t)$ inside the sphere is reflected to the image outside the sphere along the ray from C to $M(i,t)$ by

the following equation:

$$\widehat{M}_{i,t} = M_{i,t} + 2\left(R - \|M_{i,t}\|\right)\frac{M_{i,t}}{\|M_{i,t}\|}, \tag{1}$$

where $|M(i,t)|$ is the distance from the circle center C to the point $M(i,t)$. If we define $\hat{M}_i = \{\hat{M}_{i,t}\}_t^T = 1$, then the visibility of $M(i,t)$ depends on whether it lies on a convex hull of $\hat{M}_i \cup \{\mathrm{C}\}$. Strictly speaking, we select the visible point C by adjusting each class of R. Due to the efficient convex hull operation, HPR is particularly fast in its computational speed, while meeting the accuracy requirements of LiDAR data enhancement.

3.2. Voxel Point Center of Mass Localization

PCSD inputs are LiDAR point cloud data, which are defined as being points cloud $\{\mathbf{p}_i = \{\mathbf{x}_{\mathbf{p}_i}, \mathbf{f}_{\mathbf{p}_i}\} \mid i = 1, \ldots, N_{\mathbf{p}}\}$, where $\mathbf{x}_{\mathbf{p_i}} \in \mathbb{R}^{\mathbf{3}}$ are spatial coordinates, $\mathbf{f}_{\mathbf{p}_i} \in \mathbb{R}^F$ are additional features, and $N_{\mathfrak{p}}$ is the number of points in the point cloud. Assume that $\mathbf{U}^l = \{\mathbf{U}_k^l = \{\mathbf{h}_{\mathbf{U}_k^l}, \mathbf{f}_{\mathbf{U}_k^l}\} \mid k = 1, \ldots, N_l\}$ is nonempty in the lth voxel layer collection of voxels, where $\mathrm{h}_{\mathrm{U}_k^l}$ is the voxel index, $\mathbf{f}_{\mathbf{U}_k^l}$ is the feature vector, and N_l is the number of nonempty voxels. First, by computing the voxel index $\mathrm{h}_{\mathrm{U}_k^l}$ from the spatial coordinates $\mathbf{x}_i$ and the voxel grid dimension, the points within the same voxel are grouped into a set $\mathcal{N}(\mathbf{U}_k^l)$. Then calculate the center of mass $C_{\mathrm{U}_k^l}$ of the voxel point as

$$C_{\mathbf{U}_k^l} = \frac{1}{|\mathcal{N}(\mathbf{U}_k^l)|}\sum_{\mathbf{X}_{\mathbf{p}_i} \in \mathcal{N}(\mathbf{U}_k^l)} \mathbf{X}_{\mathbf{p}_i}. \tag{2}$$

In Fig. 2, we illustrate the coexistence of voxel point primes and sparse voxel features, both of which are linked to a shared voxel index. To achieve this association, the intermediate hash table utilizes the voxel index $\mathrm{h}_{\mathrm{U}_k^l}$ to connect the center of mass $C_{\mathrm{U}_k^l}$ with U_k^l.

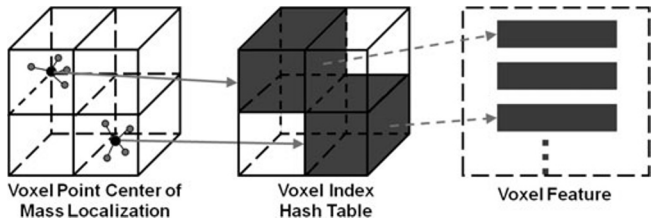


Fig. 2. Centroids (black) are mapped into their respective voxel grid indices.

Let $\mathrm{C}_k^{l+1} = \{\mathrm{C}_{\mathbf{U}_j^l} \mid K_{l+1}(\mathrm{h}_{\mathbf{U}_j^l}) = \mathrm{h}_{\mathbf{U}_k^{l+1}}\}$ represent the collection of voxel point primes, where K_{l+1} denotes the convolution block responsible for mapping the voxel index $\mathrm{h}_{\mathrm{U}_j^l}$ to $\mathrm{h}_{\mathrm{U}_k^{l+1}}$. By employing weighted averaging of the grouped voxel point primes, we can effectively calculate the subsequent layer of primes,

$$\mathbf{C}_{\mathbf{U}_k^{l+1}} = \frac{1}{|\mathcal{N}(\mathbf{U}_k^{l+1})|}\sum_{\mathbf{C}_{\mathbf{U}_j^l} \in \mathbf{C}_k^{l+1}} |\mathcal{N}(\mathbf{U}_j^l)|\mathbf{C}_{\mathbf{U}_j^l}, \tag{3}$$

where

$$|\mathcal{N}(\mathrm{U}_k^{l+1})| = \sum_{C_{\mathbf{U}_j^l} \in C_k^{l+1}} |\mathcal{N}(\mathrm{U}_j^l)|. \tag{4}$$

From this point of view, the voxel-point-centroid positioning module can be applied to large-scale point cloud processing.

3.3. Point Sparse Analysis RoI Grid Pooling

Point Sparse Analysis RoI Grid Pooling is an enhancement of RoI Grid Pooling. Initially, a random set of grid points denoted as $G^{\mathrm{b}} = \{\mathrm{g}_1, \ldots, \mathrm{g}_{U^3}\}$ is selected to sample each bounding box proposal b. Here, $\mathcal{N}(\mathrm{g}_j)$ denotes the assembly of voxel point centroids enclosed within a spherical region centered at g_j, with radius R. This sparsity analysis RoI grid pooling enables the encoding of the estimated probability densities as additional features within the sphere query, thereby facilitating more implicit feature ($\Psi^l_{\mathbf{g}_j}$) representation,

$$\Psi^l_{\mathbf{g}_j} = \left\{ \begin{bmatrix} \mathbf{f}_{\mathbf{U}^l_k} \\ \mathbf{C}_{\mathbf{U}^l_k} - \mathbf{g}_j \\ p(\mathbf{C}_{\mathbf{U}^l_k} | \mathbf{g}_j) \end{bmatrix}^{\top}, \quad \forall \mathbf{C}_{\mathbf{U}^l_k} \in \mathcal{N}(\mathbf{g}_j) \right\}. \tag{5}$$

As shown in Fig. 3, $\mathrm{C}_{\mathrm{U}^l_k} - \mathrm{g}_j$ is the relative offset between the center of mass and the voxel point and the sparsity probability $p(\mathrm{C}_{\mathbf{U}^l_k} \mid \mathfrak{g}_j)$; they are all coded as additional features.

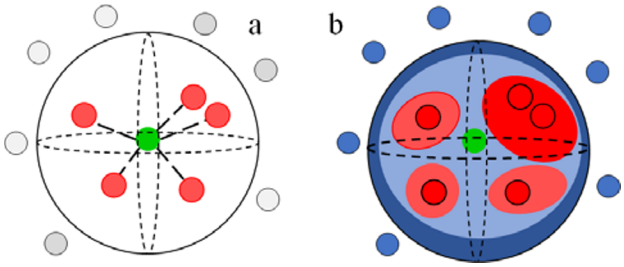


Fig. 3. Relative offset (a) and probability sparsity (b) of points. Here, high-sparsity probabilities are shown in blue, and low-sparsity ones are shown in red.

The sparsity probability of grid points reads

$$p\left(\mathbf{C}_{\mathbf{U}^l_k} \middle| \mathbf{g}_{\mathrm{j}}\right) \approx \frac{1}{|\mathcal{N}(\mathbf{g}_{\mathrm{i}})|\sigma^3} \Sigma_{\mathbf{C}_{\mathbf{U}^l_i} \in \mathcal{N}(\mathbf{g}_{\mathrm{i}})} w\left(\mathbf{C}_{\mathbf{U}^l_k}, \mathbf{C}_{\mathbf{U}^l_i}\right), \tag{6}$$

where σ represents the window width, and ω is

$$\mathcal{W}(\mathbf{C}_{\mathbf{U}^l_k}, \mathbf{C}_{\mathbf{U}^l_i}) = \prod_{d=1}^{3} w\left(\frac{\mathbf{C}_{\mathbf{U}^l_k} - \mathbf{C}_{\mathbf{U}^{l,d}_i}}{\sigma}\right). \tag{7}$$

In each XYZ dimension d, there exists an independent kernel ω. Following the incorporation of these features, the multi-scale grouping (MSG) module of PointNet is used to generate a specific feature vector $\mathrm{T}^l_{\mathfrak{g}_j}$, where maxpool indicates that the maximum value is taken to reduce the parameter effect, and FFN is a feed-forward neural network,

$$\mathrm{T}^l_{\mathrm{g}_j} = \mathrm{maxpool}\left(\mathrm{FFN}(\Psi^l_{\mathrm{g}_j})\right). \tag{8}$$

The final feature is expressed as the sum of features of different voxel layers,

$$\mathrm{T}_{\mathrm{g}_j} = [\mathrm{T}^1_{\mathrm{g}_j}, \ldots, \mathrm{T}^L_{\mathrm{g}_j}]. \tag{9}$$

We introduce a novel position encoding method (see Fig. 4) for point sparsity analysis within LiDAR point cloud $\mathbf{T}_{G^{\mathrm{b}}} = \{\mathcal{T}_{\mathrm{g}_i} \mid |\mathcal{N}(\mathrm{g}_i)| > 0, \forall \mathrm{g}_i \in \mathcal{G}^{\mathrm{b}}\}$ represented as performing a self-attention operation

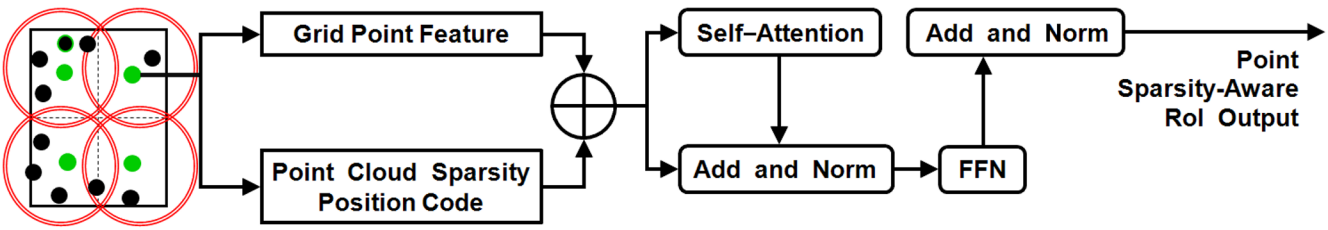


Fig. 4. The point cloud sparsity-aware RoI grid pooling for performing a self-attention operation on grid point features.

between features. This is accomplished, using standard encoder layers and residual connections, akin to the nonlocal neural network blocks. These operations enable the module to capture and incorporate spatial relationships and dependences between the grid points effectively,

$$\tilde{\mathbf{T}}_{\mathbf{g}_i} = \mathcal{T}_{\mathbf{g}_i}(\mathbf{T}_{\mathcal{G}^{\mathrm{b}}}) + \mathbf{T}_{\mathbf{g}_i}, \tag{10}$$

where $\mathcal{T}_{\mathbf{g}^i}$ is the encoder output of $\mathrm{T}_{\mathbf{g}_i}$, and $\tilde{\mathrm{T}}_{\mathbf{g}_i}$ is the feature of output grid points. The empty dot feature $|\mathcal{N}(\mathbf{g}_i)| = 0$ is not affected by the self-focus module and retains its original feature encoding.

Point Sparsity Position Coding involves utilizing the positions of local grid points and the point count within the bounding box to encode the position information within the self-attentive module.

Finally, we set the same confidence level and regression target as Second.

4. Experiment

4.1. Data Set and Training Details

We train the PCSD network on the KITTI dataset [10]; this data set is the most commonly used data set for 3D target detection algorithms, which contains 7481 training samples and 7518 test samples; we use the standard average precision (AP) on easy, moderate, and hard. The KITTI data set was produced from a Velodyne 64-line LiDAR acquisition with 64 laser lines, each channel being a separate laser beam with a vertical angular resolution of 0.4°, a horizontal angular resolution of 0.08° – 0.35°, a vertical field of view (FOV) of 26.8°, a horizontal FOV of 360°, a measurement range of 0.5 – 100 m, and a point cloud data volume of 220 w points/s. The optimum distance of LIDAR is 50 – 60 m; when it exceeds this distance, the number of scanned points will become smaller, which will affect the detection results. The PCSD uses the Adam optimizer for training. The initial learning rate is 0.01, and then updated using a single-period strategy combined with cosine. Our network model was trained on a single server, which consisted of Ubuntu 18.04 operating system, a single NVIDIA 1080 Ti GPU, training 100 epochs with a batch size of 2, and took about 6 days of time cost. Some detailed processing, we set the non-maximum suppression (NMS) threshold to 0.1 on the KITTI data set. This threshold is used to remove redundant frames, ensuring that the final bounding box prediction is accurate and precise.

4.2. Analysis of the Detection Results

4.2.1. Data Set Results

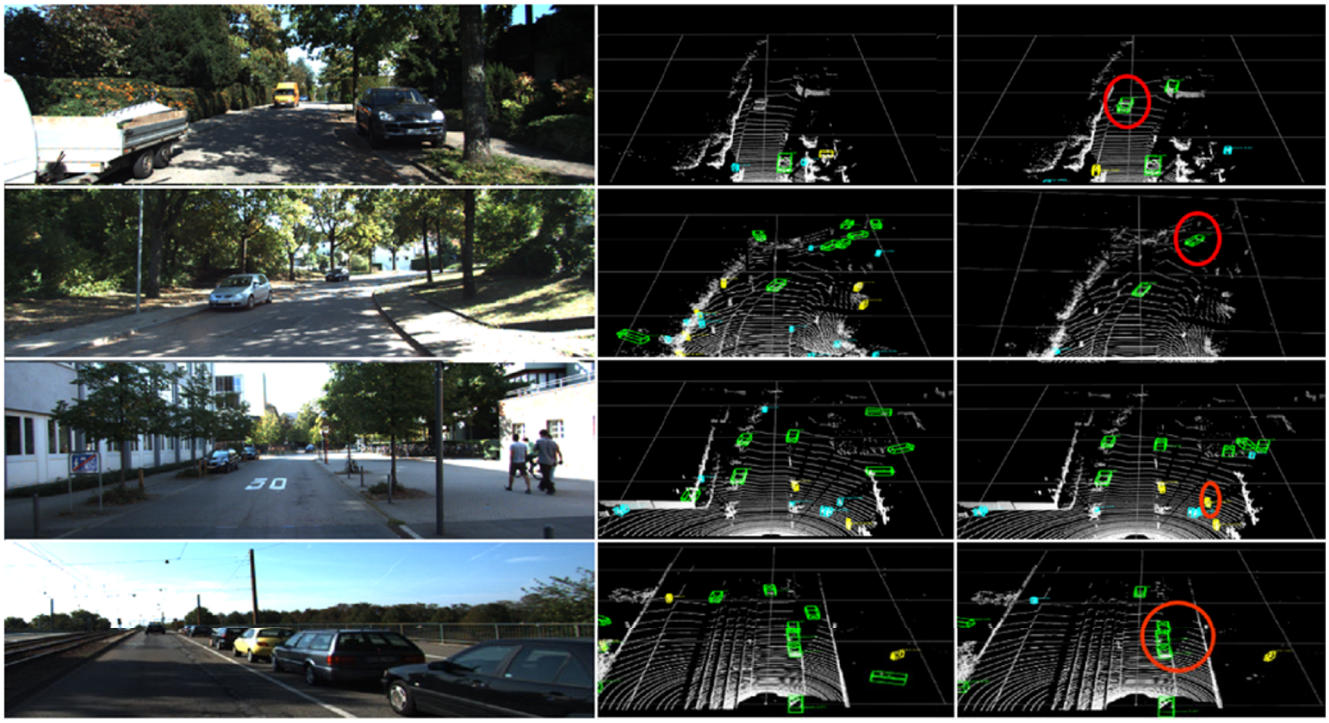
In Table 1, one can see that, in the 3D AP-R40 benchmark, our network is improved compared with PointPillars [11], PointRCNN [12], PV-RCNN [13], and CT3D networks in the categories of pedestrians and cyclists, at least + 4%. The results prove the superiority and applicability of our network, which improves the accuracy of small object detection, ensures road safety, and promotes the use of LiDAR in car.

In Fig. 5, we show the Second network visual detection results of our network along with the base network. We can see that there are missed detections due to the low car classification detection accuracy of the Second network and due to the low classification accuracy in the cyclist classification, which leads to detecting the cyclist as a pedestrian. However, our detection accuracy is ahead of it and the network is more stable, so the PCSD network can accurately locate and detect objects without false detection.

Table 1. 3D Detection Results for Categories Using AP-R40 on the KITTI Val Set.[a]

Method	Car 3D (IoU = 0.7)	Pedestrian 3D (IoU = 0.5)	Cyclist 3D (IoU = 0.5)
Second	90.55 81.61 78.61	55.94 51.14 46.16	82.96 66.74 62.78
Second-iou	86.77 79.23 77.17	36.45 33.57 31.16	83.18 63.75 60.23
PointPillars	87.75 78.39 75.18	57.30 51.41 46.87	81.56 62.80 58.83
PointRCNN	91.73 80.67 78.16	63.12 55.32 48.36	89.21 71.17 66.94
PointRCNN-IoU	91.98 80.83 78.44	62.59 55.36 48.85	89.17 72.42 67.96
PV-RCNN	92.10 84.36 82.48	62.71 54.49 49.87	89.10 70.38 66.01
CT3D	**92.34 84.97 82.91**	61.05 55.57 51.10	89.01 71.88 67.91
Ours	91.58 81.92 82.02	**69.66 64.02 59.24**	**92.41 75.61 72.43**

[a]Bold represents the highest detection accuracy.



a) **b)** **c)**

Fig. 5. The visualization results on the KITTI test set. The left-hand side is the image from the KITTI data set (a), the middle is the detection result of the Second (b), and the right=hand side is the detection result of PCSD (c). The red circles indicate where our network is superior.

4.2.2. Ablation Experiments

We perform a detailed ablation study on PCSD, using the Second network as the basis for all models. Training for 100 epochs using the KITTI data set goes as follows:

1. **Remove OA.** Starting with no preprocessing operations on the original point cloud, the training

is directly performed.

2. **Remove the PA module.** This module can get the original point cloud sparse features and further refine the regression frame. When this module is removed, the regression frame is determined only by voxel characteristics.

Table 2. The 3D Results of Ablation Study Based on KITTI.[a]

Import OA	Import PA	Car 3D (IoU = 0.7)	Pedestrian 3D (IoU = 0.5)	Cyclist 3D (IoU = 0.5)
×	×	90.55 81.61 78.61	55.94 51.14 46.16	82.96 66.74 62.78
√	×	90.30 81.10 78.04	57.35 51.02 46.65	89.10 70.38 66.01
×	√	90.96 71.61 68.10	63.09 55.56 50.72	**92.97** 74.87 71.22
√	√	**91.58 81.92 82.02**	**69.66 64.02 59.24**	92.41 **75.61 72.43**

[a]Bold represents the highest detection accuracy.

Table 3. The BEV Results of Ablation Study Based on KITTI. [a]

Method	Car 3D (IoU = 0.7)	Pedestrian 3D (IoU = 0.5)	Cyclist 3D (IoU = 0.5)
Second	90.55 81.61 78.61	55.94 51.14 46.16	82.96 66.74 62.78
Second-iou	86.77 79.23 77.17	36.45 33.57 31.16	83.18 63.75 60.23
PointPillars	87.75 78.39 75.18	57.30 51.41 46.87	81.56 62.80 58.83
PointRCNN	91.73 80.67 78.16	63.12 55.32 48.36	89.21 71.17 66.94
PointRCNN-IoU	91.98 80.83 78.44	62.59 55.36 48.85	89.17 72.42 67.96
PV-RCNN	92.10 84.36 82.48	62.71 54.49 49.87	89.10 70.38 66.01
CT3D	**92.34 84.97 82.91**	61.05 55.57 51.10	89.01 71.88 67.91
Ours	91.58 81.92 82.02	**69.66 64.02 59.24**	**92.41 75.61 72.43**

[a]Bold represents the highest detection accuracy.

In Tables 2 and 3, one can see that (1) The PA module has the greatest impact on the network, because we encode point cloud sparsity features into initial box proposals and further refine the proposals to improve classification accuracy. Using PA improves the network by $+4.82\%$ and $+7.27\%$ in pedestrian and cyclist 3D classification accuracies, respectively. (2) The second one is the OA data enhancement algorithm, because it provides multi-view, real pictures for classified objects, increasing the diversity of objects, thereby improving the accuracy of object classification. Using OA, the network improves the 3D classification accuracy of pedestrians and cyclists by $+0.59\%$ and $+2.93\%$, respectively. (3) Both OA and PA achieved significant improvements in 3D and BEV detection of pedestrians and cyclists, proving the necessity of adding these two modules. From the ablation experiments, we can see that each module plays an important role. They complement each other to achieve the best detection performance of the network.

Different from the current 2D object detection, urban-oriented autonomous driving technology re-

quires higher 3D object detection accuracy. It needs to detect surrounding objects quickly and accurately in a real environment. Therefore, we should also pay more attention to the BEV indicators in the KITTI data set.

As shown in Fig. 6, PCSD achieves the highest MAP values in terms of pedestrian and bicycle classification accuracy, both on 3D and BEV benchmarks; therefore, we consider the proposed method useful and necessary.

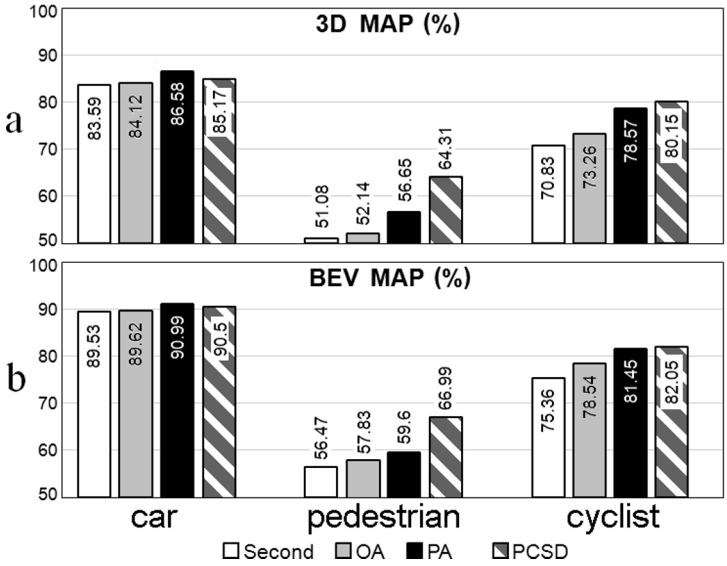


Fig. 6. 3D (a) and BEV (b) results of detections on the benchmark. Here, also the mean average precision (MAP) of the 3 classifications is used and presented in the form of a bar chart.

5. Conclusions

In this paper, we introduced a new two-stage network – PCSD. It is a structured network designed to improve small object detection accuracy. According to the experimental results, we proved that our method is novel, efficient, and useful. Compared with baseline Second network, car and pedestrian and cyclist detection results are significantly improved on the 3D benchmark, improving by 1.58%, 13.22%, and 9.33%, respectively. On the BEV benchmark, the car and pedestrian and cyclist detection results are significantly improved, improving by 0.97%, 10.52%, and 6.69%, respectively. According to the experimental results, our PCSD network has high detection accuracy, which greatly improves the detection accuracy of small objects; thus, ensuring the road safety. It also boosts the use of LiDAR in self-driving cars. For situations where car detection accuracy is less improved, in our future work, we will consider the use of a new method to refine the proposal of the initial bounding box to improve the detection accuracy. We hope that these novel perspectives can provide new ideas for the development of object detection.

Acknowledgments

This work is funded by the Jilin Province Science and Technology Plan Project and International Science and Technology Cooperation Project under Grant No. 20210402074GH. This work is also supported by the 111 Project of China under Grants Nos. D21009 and D17017, and the Future Vision Light Field and Intelligent City Laboratory.

References

1. Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), pp. 4490–4499.
2. Y. Yan, Y. Mao, and B. Li, Second: Sparsely embedded convolutional detection[J]. *Sensors*, **18**, 3337 (2018); DOI: 10.3390/s18103337

3. T. Yin and X. Zhou, "Center-Based 3D Object Detection and Tracking," in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), pp. 11784–11793;
4. J. Deng, S. Shi, P. Li, et al., "Voxel R-CNN: Towards High Performance Voxel-Based 3D Object Detection," arXiv:2012.15712 [cs.CV] (2020).
5. J. Mao, Y. Xue, M. Niu, et al., "Voxel Transformer for 3D Object Detection," in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), pp. 3164–3173; arXiv: 2109.02497v2 [cs.CV] (2021).
6. Q. Cai, Y. Pan, T. Yao, and T. Mei, "3D Cascade RCNN: High Quality Object Detection in Point Clouds," arXiv: 2211.08248 [cs.CV] (2022).
7. S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud," in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), pp. 770–779.
8. Z. Yang, Y. Sun, S. Liu, et al., "STD: Sparse-to-Dense 3D Object Detector for Point Cloud," in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), pp. 1951–1960; DOI: 10.1109/ICCV.2019.00204
9. Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), pp. 11040–11048.
10. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, *Int. J. Robotics Res.*, **32**, 1231 (2013); DOI: 10.1177/0278364913491297
11. A. H. Lang, S. Vora, H. Caesar, et al., "PointPillars: Fast Encoders for Object Detection From Point Clouds," in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), pp. 12697–12705.
12. S. Shi, X. Wang, H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), pp. 770–779.
13. S. Shi, C. Guo, L. Jiang, et al., "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," arXiv: 1912.13192 [cs.CV] (2019).