Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 5603123, 10 pages
https://doi.org/10.1155/2023/5603123

Research Article

Pillar-Based 3D Object Detection from Point Cloud with Multiattention Mechanism

Xin Li[1], **Bifa Liang**[1], **Jinhao Huang**[1], **Yuyang Peng**[2], **Yier Yan**[1], **Jun Li**[1], **Wenli Shang**[1] and **Wei Wei**[1]

[1]*Research Center of Intelligent Communication Engineering, School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China*
[2]*School of Computer Science and Engineering, Macau University of Science and Technology, Taipa, 999078 Macau SAR, China*

Correspondence should be addressed to Jun Li; lijun52018@gzhu.edu.cn

Received 6 September 2022; Revised 20 October 2022; Accepted 27 January 2023; Published 9 February 2023

Academic Editor: Xiaojie Wang

Object detection in point clouds is a critical component in most autonomous driving systems. In this paper, in order to improve the effectiveness of image feature extraction and the accuracy of detection of point clouds, a pillar-based 3D point cloud object detection algorithm with multiattention mechanism is proposed, which includes three attention mechanisms SOCA, SOPA, and SAPI. The results show that the recognition accuracy of the optimized algorithm for cars, pedestrians, and cyclists on KITTI dataset is significantly improved on the detection benchmarks of BEV and 3D. Despite using only LiDAR, our algorithm outperforms PointPillars, which is one of the state-of-the-art algorithms for 3D object detection, with respect to both 3D and BEV view KITTI benchmarks while maintaining a relatively competitive speed.

1. Introduction

Machine learning (ML) and the Internet of Things (IoT) can be applied in almost every industry [1, 2], from implementing AI digital assistants to supply chain automation. With the development of machine learning technologies, autonomous driving [3] has become more and more popular. For autonomous driving, the significance of IoT and autonomous driving technology cannot be overstated. Information is the bridge between IoT sensors and self-driving cars. Relying on IoT sensors for information collection and analysis, self-driving cars are one step closer to large-scale applications. In self-driving cars, there are many sensors such as LiDAR, radar, cameras, and IoT devices that communicate with each other. Using various deep learning models based on convolutional neural networks (CNN), based on the data received, enables the car to learn automatically and continuously improve detection performance over time and experience.

At present, 3D point cloud data got from LiDAR is mainly used by autonomous vehicles for object detection. Compared with two-dimensional images, LiDAR can provide more reliable depth and shape information and locate objects with higher accuracy. However, due to nonuniform sampling, occlusion, and reflection in 3D space, the LiDAR point clouds are sparse and have highly variable density. The accuracy of traditional 3D object detection algorithms based on manual features often suffers as a result. In recent years, 3D point cloud object detection algorithms [4–6] based on deep neural networks have been improved to some extent in terms of accuracy as deep neural networks have shown excellent capability of feature extraction and can handle high-dimensional data. Nevertheless, there is still enough potential for improvement in the accuracy of detection results for some categories, due to the highly sparse and inherently irregular nature of point clouds. Some of earlier works employ 3D convolution approaches or adopt the methods that project point clouds into a perspective view. Perspective view is another extensively used representation of LiDAR. Following this line of research, the VeloFCN [7] and LaserNet [8] are some of the representative works. The new research favors BEV of the LiDAR point cloud, the

advantage of which is almost no occlusion. In 2018, Simony et al. [6] introduced Complex-yolo, a model that projects point clouds onto a 2D plane and employs a 2D image approach for object detection, thus speeding up inference of the network. However, the projection is limited by the sparsity of the point cloud, which prevents convolution from extracting features better. To cope with this issue, a common method transforms the point cloud raster into 3D voxel grids and encodes each voxel one by one using handcrafted features. However, manual design not only cannot make full and effective use of the object's 3D information but also is not conducive to the application of other radars. Based on PointNet [9], an end-to-end deep neural network was proposed by Qi et al., in which point features are learned directly from point clouds. In 2018, Zhou and Tuzel [4] first proposed an end-to-end trainable network VoxelNet, a universal 3D detection framework. Different from most previous work, VoxelNet starts to learn information-rich feature representation and can simultaneously learn different feature representation from point clouds. However, the disadvantage of 3D convolution is that it is too time-consuming and is confronted with high computation complexity, leading to a slow inference speed of the network. Later on, Yan et al. [10] proposed SECOND, which reduces memory consumption and speeds up computation through sparse convolution operation. In order to use the standard 2D convolution detection pipeline to improve the inference speed, Lang et al. [11] proposed PointPillars, which encode point clouds into vertical pillars, a special division of voxels. To further improve the performance of point clouds object detection in challenging situations, TANet is presented by Liu et al. [12] in 2019, which utilizes a combination of attention mechanisms.

The so-called "attention mechanism" is a way of perception that mimics the human brain and human vision, a mechanism for focusing local information [13, 14]. The attention mechanism can dynamically select areas of attention as the task changes, which is achieved by adaptively assigning weights based on the degree of significance of the inputs. The point-wise attention mechanism proposed by TANet assigns weights on the basis of the importance of the points, but it has not considered the correlations between points, resulting in the loss of a fraction of valuable geometric information. Considering that each point within a pillar is semantically linked, we propose a new method called second order of point attention (SOPA), which links points with each other within a pillar. The experimental results show that the detection accuracy of pillar-based 3D object detection method with SOPA is better than that with point-wise attention. Particularly, for the two categories of pedestrians and cyclists, which are currently detected with relatively low accuracy and are more challenging to improve, there is an improvement for accuracy of each category by using SOPA. Similarly, in consideration of the channel-to-channel relationship, we propose second order of channel attention mechanism (SOCA) based on pillars between the backbone network and the feature extraction network stage, which can extract more effective information. In addition, taking into account that not all features in the pseudoimage have the same contribution to the detection task, we propose the spatial attention of pseudoimage mechanism (SAPI), which assigns different weights to each point in the pseudoimage with regard to the importance of the region in the pseudoimage to the task in the pseudoimage generation stage, which could lead to more accurate detection results. Compared with existing 3D point cloud object detection algorithms, a proposal that integrates these three second order attention mechanisms can achieve higher performance detection with relatively competitive speed.

2. Related Work

2.1. PointPillars Network. PointPillars [11] comprises three main phases: (1) a feature encoder network that transforms point clouds to sparse pseudo-images, (2) a 2D convolutional backbone network that converts pseudo-images into high-level representations, and (3) a detection head that detects and regresses 3D boxes.

2.1.1. Pointcloud to Pseudoimage. The space is partitioned into pillars [11], and at the same time, raw point clouds are assigned to the pillars and then converted into a form of sparse pseudoimage. Given l_0 to represent a point in the raw point cloud, which has coordinates x, y, z, and reflection intensity r. First, the input point cloud is partitioned into multiple pillar cells. And each pillar is a 3D grid obtained by dividing the point cloud in the X and Y plane in certain steps [0.16, 0.16]; then, a set of pillars A can be obtained. Each point in a pillar is encoded as a nine-dimensional vector E, which could be parameterized as $(x, y, z, r, x_0, y_0, z_0, x_d, y_d)$. Here, (x_0, y_0, z_0) represents the geometric centers of all points in pillars where the point cloud is located, and (x_d, y_d) indicates the offset of each point from the geometric center.

Random sampling is performed if there are more than $P = 32$ points in each pillar, and zero filling is employed if less than P, to ensure that the number of points in each pillar remains at 32. In this way, a feature tensor of (E, A, P) is obtained, and then the feature extraction is performed on the tensor. The original dimension E of the point cloud is 9, while the dimension F of point cloud is expanded to 64. Then, a feature tensor of (F, A, P) is acquired. The 2D feature map (F, A) is gained by performing the max pooling operation according to the third dimension. The last step is to generate a pseudoimage by a scatter calculus. Specifically, the original pillar is replaced by the (F, A) feature tensor generated in the previous step based on the pillar index value of each pillar to create a pseudoimage ($F = 64$, $H = 440$, $W = 500$). Here, W and H denote the width as well as the height of the canvas. In the process of constructing stacked pillars from the point cloud, the coordinates corresponding to each pillar are recorded. When the pseudoimage is constructed by the learned features, the pillars are filled with the corresponding learned features according to the pillar index.

2.1.2. Backbone. The backbone network is similar to that of VoxelNet [4], covering two subnetworks: the first one is a top-down structure with successively decreasing resolution, where the low resolution is responsible for extracting the

high-dimensional features, and the second network carries out the upsampling operations, in order to stitch together the features of the corresponding size. The first network is composed of three blocks, which consist of 3×3 conv layers behind followed by a Batch-Norm [15] layer and ReLU [16] layer. Here, the stride size is two, and the resolution decreases by half in the (H, W) direction after each convolutional layer. As it passed through three blocks, the resolution dropped three times, down to one-eighth. At the same time, the channel dimension expands from F to $4F$.

For the second network, after upsampling operation, the three blocks yield the same size features. Then, the features with the same resolution obtained after deconvolution are concatenated together to acquire an integrated feature.

2.1.3. Detection Head. SSD detection head [17] is employed for the 3D object detection in the final stage. Two anchors of 0 and 90 angles are put in the center of each pillar. For the calculation of IOU [18], rotating IOU [19] is the best one in terms of accuracy.

2.2. Attention Mechanism. Attention mechanism was first proposed in the field of image in the 1990s. The development of attention mechanism goes through four main phases. First, it utilizes RNN [20] and reinforcement learning to implement spatial attention. After that, Jaderberg et al. [21] proposed the STN, learning affine transformations to select important regions. In the third phase, CBAM [22] and Eca-net [23] are representative, of which the novel attention mechanism can adaptively predict underlying kernel features. In the fourth phase, self-attention is highly motivated [24]. Wang et al. were the first to introduce self-attention into computer vision and achieved great success in video understanding and object detection [25]. In recent years, as it has remarkable performance, an increasing number of studies based on attention mechanisms have emerged in the field of computer vision [26–28].

The existent attention methods can be divided into channel attention [23, 29], spatial attention [30, 31], temporal attention, and branch channel [32].

2.2.1. Point-Wise Attention. In the pillar-based 3D point cloud object detection algorithm, for the K^{th} pillar in the space, the global features of the points in the pillar are retained after the max-pooling process, and then the vector $I \in R^{N*1}$ can be obtained. Here, N represents the number of points. To limit the complexity of the network, only two fully connected layers are employed. The point-wise attention mechanism can be expressed by the following formula:

$$\mathrm{PA} = W_2\delta(W_1 I). \tag{1}$$

$\delta(\bullet)$ is the ReLU activation function between two fully connected layers, where $W_1 \in R^{t\times N}$, $W_2 \in R^{N\times t}$ denote weights of the two fully connected layers. Here, t is the output length of the first fully connected layer as well as the output length of the second fully connected layer. $\mathrm{PA} \in R^{N*1}$ represents the point-wise attention of the points in the K^{th} pillar.

2.3. Contributions

(1) We propose three effective methods, including the second order of point attention mechanism (SOPA) based on pillars, the second order of channel attention mechanism (SOCA), and the spatial attention of pseudoimage mechanism (SAPI) after the stage of generating pseudoimage, to implement high-precision real-time object detection, respectively

(2) We conducted experiments on the KITTI dataset [33] and presented the latest detection results of cars, pedestrians, and cyclists on BEV, 3D, and AOS benchmarks. Our model runs at 34 Hz, while the detection accuracy of the category of cyclists and pedestrians, which is slightly low, is substantially improved by about 6% mAP (mAP on both BEV and AOS) over the other methods

(3) We performed several ablation experiments to examine the key influencing factors for achieving performance improvement

3. Multiattention Mechanism Network

The architecture of the multiattention mechanism network is demonstrated in Figure 1. The network accepts the raw point cloud as input and predicts 3D bounding boxes to identify cars, pedestrians, and cyclists. It is composed of the following stage: (1) first, the point cloud is voxelized, and then SOPA operation is performed on the point cloud, followed by a pillar feature network to convert it into the form of sparse pseudo-image. (2) Then, the generated sparse pseudoimage is subjected to SOCA operation. (3) After the backbone network, the SAPI operation is performed on the features of the output sparse pseudoimage, and finally, the 3D bounding box of the object is predicted by the detection head.

3.1. Second Order of Point Attention. As presented in Figure 2, when the point features are fed into the second order of attention module, the SOPA weight would be obtained as the output, and at this moment, the module is called SOPA module.

In the K^{th} pillar, all points $X^K \in R^{N\times C}$, where N represents the max number of points and C refers to the number of channels. Through the operation of a max pooling layer, a vector of the maximum values along the dimension C is obtained as $E^K \in R^{N\times 1}$, where $N \times 1$ represents a vector with N rows and 1 column. To maintain a large model capacity and further ensure the migration of representation capabilities, it is fed into a fully connected layer. Then, new vector is obtained as $Q^K \in R^{t\times 1}$, where t is the number of points after reduction through the fully connected layer W_1. Then, an activation function ReLU is to prevent the network from gradient disappearance. And then, the covariance matrix between two points in the same pillar is computed to get their correlation. This covariance matrix is expressed as $I^k \in R^{t\times t}$, where t represents the number of points in the SOPA while t refers to the number of channels in the

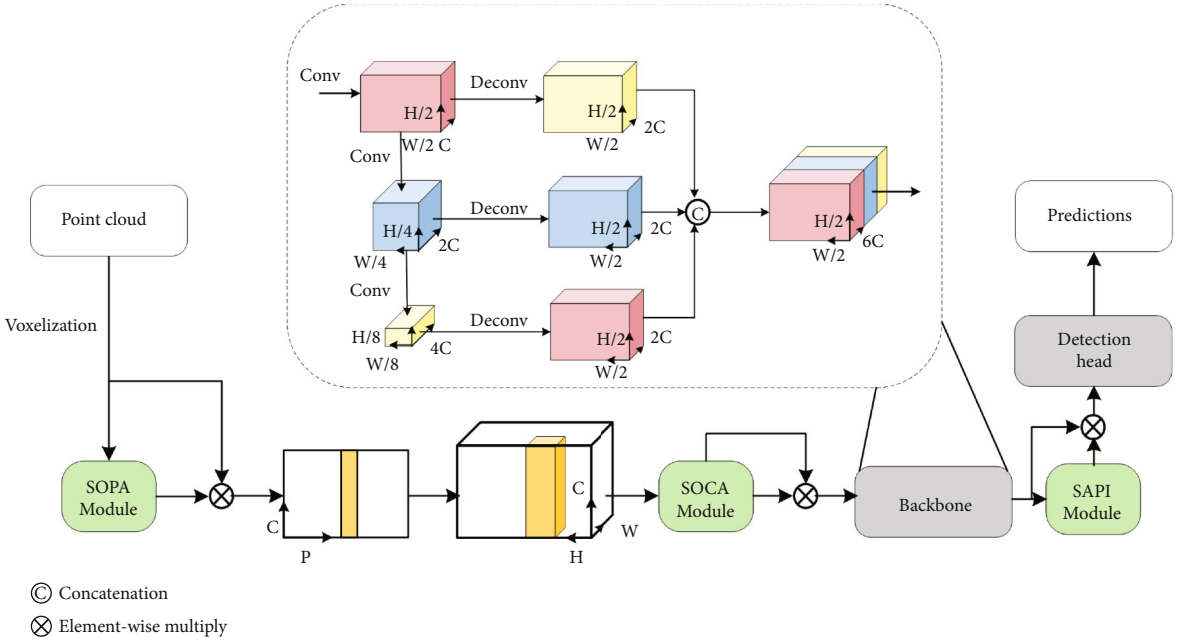

Figure 1: Network overview. The network mainly consists of SOPA, a pillar feature network, SOCA, backbone network, SAPI, and SSD detection head.

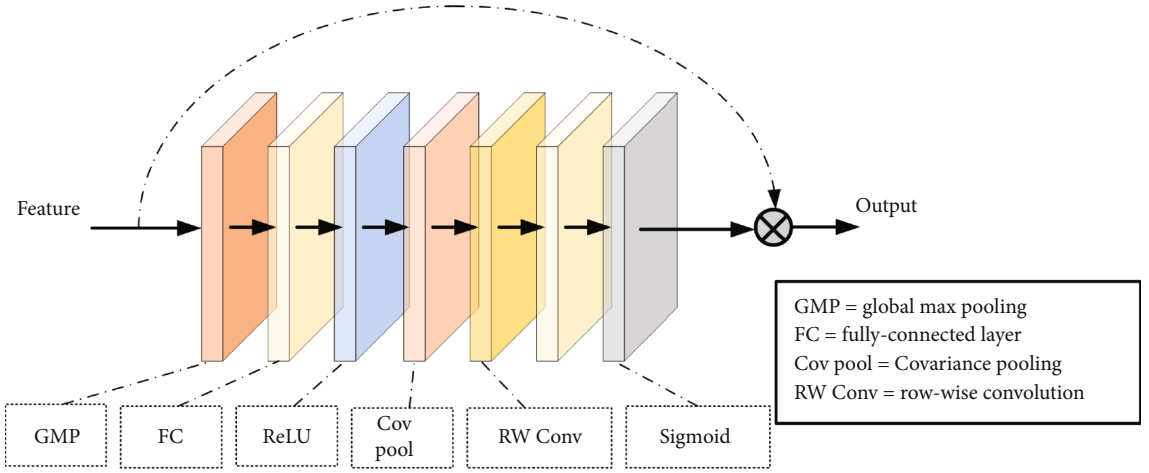

Figure 2: Second order of attention module.

SOCA, and $t \times t$ means dimension. Next, row-by-row convolution operation is applied to the covariance matrix; then, the vector is obtained as $J \in R^{t\times 1}$. The vector $J \in R^{t\times 1}$ is then fed into the fully connected layer W_2, and the N-dimensional attention vector is then gotten using a sigmoid function, denoted as $T \in R^{N\times 1}$. The SOPA can be presented as the expression as follows:

$$T = \sigma(W_2 \mathrm{RC}(\mathrm{Cov}(\delta(W_1(\mathrm{GMP}(X)))))). \quad (2)$$

Here, $\mathrm{Cov}(\bullet)$ calculates the covariance matrix of points, and $\mathrm{RC}(\bullet)$ denotes row convolution. $\delta(\bullet)$ is the ReLU activation function while $\sigma(\bullet)$ is the sigmoid function. With $W_1 \in R^{t\times N}$, $W_2 \in R^{N\times t}$ represents two different fully connected layers, respectively, and $X^K \in R^{N\times C}$ is the point in the given K^{th} pillar.

3.2. Second Order of Channel Attention. SOCA is similar to SOPA, as presented in Figure 2, when channel features are fed as inputs to the second order of attention module, the output is obtained as the weights of SOCA. For the pseudoimage features $Y \in R^{C\times H\times W}$ generated through a pillar feature network, SOCA can be expressed by the following equation:

$$M = \sigma(W_2 \mathrm{RC}(\mathrm{Cov}(\delta(W_1(\mathrm{GMP}(Y)))))), \quad (3)$$

where the superscripts H and W are the height and width of the pseudoimage.

3.3. Spatial Attention of Pseudoimage. Not all features of regions in the space can contribute equally to the task, and only the regions which are relevant to the task are of interest. Pixel points in each layer of the spatial feature are assigned with different weights. In this way, task-relevant parts of the space are chosen and then processed. Here, the spatial attention operation is performed on the pseudoimage, and that is why we refer to it as spatial attention of pseudoimage. If the feature map P and the signal H are given as input, the final output yields the spatial attention weights S. SAPI can be formulated as

$$S = \sigma\left(\varphi\left(\delta\left(\varphi_p(P)\right) + \varphi_h(H)\right)\right). \tag{4}$$

The correlation between P and H can be expressed as

$$H = \varphi_0(P). \tag{5}$$

Here, φ, φ_0, φ_p, and φ_h are computed as linear transformations with 1×1 convolution.

4. Implementation Details

4.1. Loss Function. We use the same loss functions as presented in SECOND and PointPillars. We parameterize a 3D ground truth box as $(x, y, z, w, l, h, \theta)$, where (x, y, z) represent the center location, and (w, h, l) and θ are the size and the heading angle of the bounding box, respectively. The regression residuals between ground truth and anchor boxes are computed as follows:

$$\Delta x = \frac{x^{gt} - x^a}{d^a}, \Delta y = \frac{y^{gt-y^a}}{d^a}, \Delta z = \frac{z^{gt} - z^a}{h^a}, \tag{6}$$

$$\Delta w = \log \frac{w^{gt}}{w^a}, \Delta l = \frac{l^{gt}}{l^a}, \Delta h = \log \frac{h^{gt}}{h^a}, \tag{7}$$

$$\Delta\theta = \sin\left(\theta^{gt} - \theta^a\right), \tag{8}$$

where x^{gt} denoting ground truth, and x^a is the bounding box, with $d^a = \sqrt{(l^a)^2 + (w^a)^2}$. All of the losses are summed up as the total loss of the overall network model, with the overall loss function defined as

$$L = \frac{1}{N_{\text{pos}}}\left(\beta_{\text{loc}} L_{\text{loc}} + \beta_{\text{cls}} L_{\text{cls}} + \beta_{\text{dir}} L_{\text{dir}}\right), \tag{9}$$

where N_{pos} represents the number of positive anchors. We set $\beta_{\text{loc}} = 2$, $\beta_{\text{cls}} = 1$, and $\beta_{\text{dir}} = 0.2$.

The regression loss is denoted by the following equation:

$$L_{\text{loc}} = \sum_{b \in (x,y,z,w,l,h,\theta)} \text{Smooth}L1(\Delta b). \tag{10}$$

For the classification loss, we adopt focal loss [34]:

$$L_{\text{cls}} = -a(1-p)^r \log p, \tag{11}$$

where p denotes the probability of being a positive anchor. We adopt the settings of $r = 2$ and $a = 0.25$.

5. Experiment

5.1. Dataset. All test results are evaluated using KITTI's official evaluation test metrics, including aerial view (BEV), 3D, 2D, and average orientation similarity (AOS), where AOS evaluates the average orientation similarity of two-dimensional detection (BEV). KITTI dataset are available in easy, moderate, and hard difficulties, and the official KITTI leaderboard ranked by performance on moderate. Performance is measured as mean average precision (mAP) on KITTI validation.

The experiments all employ the KITTI 3D object detection benchmark dataset, which is composed of 7,481 training samples and 7,518 test samples. And the KITTI benchmark requires detection category [33], which include cars, pedestrians, and cyclists. We also follow the generally used training-validation split, which contains 3,712 training samples and 3,769 validation samples.

5.2. Settings. Here, we use xy resolution: 0.16 m, maximum number of points per pillar (V): 100, maximum number of pillars (Z): 12000. Our approach is based on the PyTorch framework, with all networks trained on the NVIDIA 2080Ti computing platform.

We train it for 160 epochs with an initial learning rate of $2 * 10^{-4}$ and decrease the learning rate by 0.8 every 15 epochs with Adam [35] optimizer.

5.3. Results. In this section, we will introduce the results of our object detection algorithm using three types of attention mechanisms. The tables below present the effect of adding three kinds of attention mechanisms to the network. Besides, combining any of the two attention mechanisms (SOPA, SOCA) separately results in 2–3% mAP boost overall.

As shown in results Tables 1–3, the network of object detection using our second order of multiattention mechanism exceeds most of the published networks (mAP on both AOS and BEV benchmarks). As listed in results Tables 4 and 5, we also find our method combining three attention mechanisms achieved BEV mAP (88.37%, 54.13%, and 67.38%) and 3D mAP (76.22%, 49.45%, and 63.58%) in the moderate difficulty categories of car, pedestrian, and cyclist, respectively. Moreover, in most of methods using only LiDAR, better results are achieved in all categories in three difficulty cases.

We show several qualitative results in Figure 3. And while we trained only on LiDAR point clouds, the 3D bounding boxes have been projected into the camera coordinate system for the sake of clarity of interpretation. Overall, our model provides highly accurate 3D bounding boxes in all categories.

6. Ablation Experiments

In this section, we provide the results of ablation experiments to evaluate the key factors that affect the accuracy of the experiments.

Table 1: Results on the KITTI test BEV detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	88.35	86.10	79.83	84.76	58.66	50.23	47.19	52.02	79.14	62.25	56.00	65.79
PointPillars + point-wise attention	89.63	86.72	83.72	86.69	58.62	53.29	48.69	53.53	83.69	63.04	61.62	69.45
PointPillars + SOPA	91.23	87.27	84.90	87.80	58.11	52.24	49.12	53.16	81.72	66.96	62.60	70.43

Table 2: Results on the KITTI test 3D detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	79.05	74.99	68.30	74.11	52.08	43.53	41.49	45.70	75.78	59.07	52.92	62.59
PointPillars + point-wise attention	86.82	75.80	72.94	78.52	53.01	46.74	42.96	47.57	78.71	59.84	56.56	65.03
PointPillars + SOPA	86.13	75.27	72.86	78.08	53.44	47.61	43.45	48.10	79.34	61.63	59.18	66.72

Table 3: Results on the KITTI test average orientation similarity (AOS) detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	90.19	88.76	86.38	88.44	32.39	31.41	29.84	31.21	82.43	68.16	61.96	70.85
PointPillars + point-wise attention	90.53	88.75	87.25	88.84	50.18	46.40	43.62	46.73	84.49	69.75	65.53	73.25
PointPillars + SOPA	90.56	88.77	87.26	88.86	54.67	50.85	47.43	50.97	85.09	73.38	69.71	76.06

Table 4: Results on the KITTI test BEV detection benchmark.

Model	Speed (Hz)	mAP	Car			Pedestrian			Cyclist		
		Mod.	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)
MV3D [36]	2.8	N/A	86.02	76.90	68.49	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [37]	16.7	N/A	88.81	85.83	77.33	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [38]	10	N/A	88.20	79.41	70.02	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [39]	10	64.11	88.53	83.79	77.90	58.75	51.05	47.54	68.09	57.48	50.77
F-PointNet [40]	5.9	65.39	88.70	84.00	75.33	58.09	50.22	47.20	75.38	61.96	54.68
HDNET [41]	20	N/A	89.14	86.57	78.32	N/A	N/A	N/A	N/A	N/A	N/A
PIXOR++ [41]	35	N/A	89.38	83.70	77.97	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet	4.4	58.25	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
SECOND	20	60.56	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78
PointPillars	62	66.19	88.35	86.10	79.83	58.66	50.23	47.19	79.14	62.25	56.00
Ours	34	69.96	92.76	88.37	85.31	60.67	54.13	48.22	82.22	67.38	59.86

As shown in Tables 1–3, the ablation experimental results show that the accuracy of SOPA is overall improved by 1–2% mAP compared to adding point-wise attention. The points in the pillars are correlated with each other, thus processing the points in the point cloud individually will inevitably drop part of the useful geometric information, further affecting the detection accuracy. SOPA relates points within the same pillar to retain more meaningful information.

As indicated by our results in Tables 6–8, from the ablation experimental results, we can observe that the accuracy of adding the SOCA is superior to that of the only-fused channel attention mechanism, and the accuracy of adding the SOCA has overall improvement of 4–5% mAP compared with the existing method PointPillars. In particular, from Tables 5 and 9, we can see that the detection results of cyclist categories with slightly low detection accuracy are improved

Table 5: Results on the KITTI test 3D detection benchmark.

Model	Speed (Hz)	mAP	Car			Pedestrian			Cyclist		
		Mod.	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)
MV3D [36]	2.8	N/A	71.09	62.35	55.12	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [37]	16.7	N/A	82.54	66.22	64.04	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [38]	10	N/A	83.71	73.04	59.16	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [39]	10	55.62	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61
F-PointNet [40]	5.9	57.35	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39
VoxelNet	4.4	49.05	77.47	65.11	57.73	39.48	33.69	31.50	61.22	48.36	44.37
SECOND	20	56.69	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
PointPillars	62	59.20	79.05	74.99	68.30	52.08	43.53	41.49	75.78	59.07	52.92
Ours	34	63.08	87.48	76.22	73.55	54.78	49.45	43.07	81.69	63.58	60.55

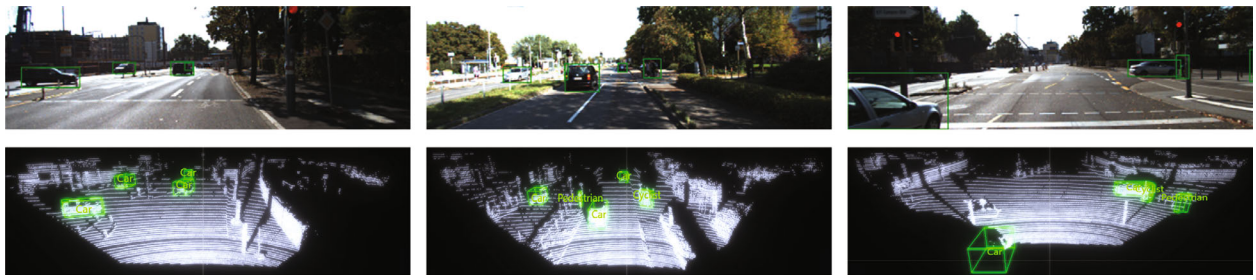

Figure 3: Visualization of our results. Qualitative results of the proposed method on the KITTI validation split. For each sample, the upper part is the image labeled with the predicted 3D bounding box, and the lower part is a representative view of the corresponding point clouds (best viewed when zoomed-in).

Table 6: Results on the KITTI test BEV detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	88.35	86.10	79.83	84.76	58.66	50.23	47.19	52.02	79.14	62.25	56.00	65.79
PointPillars + channel attention	89.38	85.39	83.43	86.06	57.38	51.84	48.12	52.44	79.62	63.59	60.38	67.86
PointPillars + SOCA	89.50	86.30	83.80	86.53	58.37	53.07	49.18	53.54	81.26	64.75	61.87	69.29

Table 7: Results on the KITTI test 3D detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	79.05	74.99	68.30	74.11	52.08	43.53	41.49	45.70	75.78	59.07	52.92	55.35
PointPillars + channel attention	82.68	75.37	72.10	76.71	52.19	45.82	42.48	46.83	75.44	59.03	55.61	63.36
PointPillars + SOCA	83.92	76.32	72.77	77.67	53.91	47.35	42.90	48.07	78.58	61.08	57.54	65.73

by a large margin (6% 3D mAP and 7% AOS mAP) on the detection benchmark of 3D and AOS. In the backbone network, each channel is processed separately in isolation. Ignoring the correlation between channels will lose some valuable information and decrease the detection precision. And SOCA associates channels with channels to retain more useful feature information.

The residual influencing factors might be the selection of various hyperparameters, including network design (convolution size, number of convolution layers, convolution type, and number of channels), projection using only a bird's eye view or incorporating a front view, whether to choose pitch angle as main parameter in the pillar feature net, choice of single or multiple detection heads, and lots more, which

Table 8: Results on the KITTI test average orientation similarity (AOS) detection benchmark.

Model	Car				Pedestrian				Cyclist			
	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)	Easy (%)	Mod. (%)	Hard (%)	mAP (%)
PointPillars	90.19	88.76	86.38	88.44	32.39	31.41	29.84	31.21	82.43	68.16	61.96	70.85
PointPillars + channel attention	90.55	88.75	87.09	88.79	54.61	50.27	47.38	50.75	84.24	70.09	65.78	73.37
PointPillars + SOCA	90.61	88.91	87.24	88.92	58.30	53.48	50.37	54.05	83.99	71.85	68.41	74.75

Table 9: Results on the KITTI test average orientation similarity (AOS) detection benchmark.

Model	Speed (Hz)	mAOS	Car			Pedestrian			Cyclist		
		Mod.	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)	Easy (%)	Mod. (%)	Hard (%)
SubCNN [42]	0.5	72.71	90.61	88.43	78.63	78.33	66.28	61.37	71.39	63.41	56.34
AVOD-FPN [39]	10	63.19	89.95	87.13	79.74	53.36	44.92	43.77	67.61	57.53	54.16
SECOND	20	54.53	87.84	81.31	71.95	51.56	43.51	38.78	80.97	57.20	55.14
PointPillars	62	68.86	90.19	88.76	86.38	32.39	31.41	29.84	82.43	68.16	61.96
Ours	34	72.14	94.89	91.09	88.47	56.52	50.99	47.57	86.80	74.34	70.67

requires more experimental studies to separate and evaluate each potential influence factor.

7. Conclusions

This paper presents an object detection algorithm based on PointPillars by combining multiple attention mechanisms. A novel deep network and encoder, which improves the traditional end-to-end algorithm and adds multiple attention mechanisms to the network structure in the stage of feature extraction, improves the effectiveness of image feature extraction. On KITTI dataset, the algorithm provides higher detection performance (BEV, 3D and AOS mAP) at a relatively competitive speed. Our experimental results show that our point cloud object detection algorithm using multiple attention mechanisms is an excellent network for LiDAR 3D object detection at present, and the comparison with PointPillars further demonstrates the effectiveness of the proposed method in this paper. It is worth noting that the proposed object detection algorithm can be extended into intelligent transportation systems [43, 44] and private transmission systems [45, 46].

Data Availability

We use xy resolution: 0.16 m, maximum number of points per pillar (V): 100, maximum number of pillars (Z): 12000. Our approach is based on the pytorch framework, with all networks trained on the NVIDIA 2080ti computing platform.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Guangzhou Science and Technology Project under Grant 202102021132, in part by National Nature Science Foundation of China under Grant 62173101, in part by Guangzhou Key Laboratory of Software-Defined Low Latency Network under Grant 202102100006, in part by the Open Research Project of Zhijiang Laboratory under Grant 2021KF0AB06, and in part by the International Collaborative Research Program of Guangdong Science and Technology Department under Grants 2020A0505100061.

References

[1] A. Tizghadam, H. Khazaei, M. H. Moghaddam, and Y. Hassan, "Machine learning in transportation," *Journal of Advanced Transportation*, vol. 2019, Article ID 4359785, 3 pages, 2019.

[2] P. Salva-Garcia, J. M. Alcaraz-Calero, Q. Wang, J. B. Bernabe, and A. Skarmeta, "5G NB-IoT: efficient network traffic filtering for multitenant IoT cellular networks," *Security and Communication Networks*, vol. 2018, Article ID 9291506, 21 pages, 2018.

[3] M. Rasib, M. A. Butt, S. Khalid et al., "Are self-driving vehicles ready to launch? An insight into steering control in autonomous self-driving vehicles," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6639169, 22 pages, 2021.

[4] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3D object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, Salt Lake City, UT, USA, 2018.

[5] B. Yang, W. Luo, and R. Urtasun, "Pixor: real-time 3D object detection from point clouds," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, Salt Lake City, UT, USA, 2018.

[6] M. Simony, S. Milzy, K. Amendey, and H. M. Gross, "Complex-yolo: an Euler-region-proposal for real-time 3D object

detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 197–209, Munich, Germany, 2018.

[7] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D LiDAR using fully convolutional network," in *Proceedings of the Robotics: Science and Systems*, pp. 1–8, MI, USA, 2016.

[8] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: an efficient probabilistic 3D object detector for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12677–12686, Long Beach, CA, USA, 2019.

[9] R. Qi, H. Su, K. C. Mo, and L. P. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 75–85, Honolulu, HI, USA, 2017.

[10] Y. Yan, Y. Mao, and B. Li, "SECOND: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[11] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12697–12705, Long Beach, CA, USA, 2019.

[12] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: robust 3D object detection from point clouds with triple attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11677–11684, New York, USA, 2020.

[13] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews Neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.

[14] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in Cognitive Sciences*, vol. 9, no. 4, pp. 188–194, 2005.

[15] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, Lille, France, 2015.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, Haifa, Israel, 2010.

[17] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference*, pp. 21–37, Amsterdam, The Netherlands, 2016.

[18] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Computer Vision – ECCV 2018*, pp. 784–799, Munich, Germany, 2018.

[19] D. Zhou, J. Fang, X. Song et al., "Iou loss for 2D/3D object detection," in *2019 International Conference on 3D Vision (3DV)*, pp. 85–94, Quebec City, QC, Canada, 2019.

[20] F. Wang and D. M. Tax, "Survey on the attention based RNN model and its applications in computer vision," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 1–42, Las Vegas, NV, USA, 2016.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in neural information processing systems*, p. 28, Curran Associates, Inc, 2015.

[22] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *Computer Vision – ECCV 2018*, pp. 3–19, Munich, Germany, 2018.

[23] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),*, pp. 11531–11539, Seattle, WA, USA, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Advances in neural information processing systems*, p. 30, Curran Associates, Inc, 2017.

[25] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, Salt Lake City, UT, USA, 2018.

[26] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9167–9176, Seoul, Republic of Korea, 2019.

[27] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Advances in neural information processing systems,*, p. 32, Curran Associates, Inc, 2019.

[28] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: criss-cross attention for semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV),*, pp. 603–612, Seoul, Republic of Korea, 2019.

[29] L. Chen, H. Zhang, J. Xiao et al., "SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6298–6306, Honolulu, HI, USA, 2017.

[30] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV),*, pp. 6688–6697, Seoul, Republic of Korea, 2019.

[31] H. Zhao, Y. Zhang, S. Liu et al., "PSANET: point-wise spatial attention network for scene parsing," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 267–283, Munich, Germany, 2018.

[32] H. Zhang, C. Wu, Z. Zhang et al., "Resnest: split-attention networks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2735–2745,, New Orleans, LA, USA, 2022.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, Providence, RI, USA, 2012.

[34] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, Italy, 2017.

[35] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations, pp. 1-15, ICLR*, p. 13, San Diego, 2015.

[36] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6526–6534, Honolulu, HI, USA, 2017.

[37] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 641–656, Munich, Germany, 2018.

[38] K. Shin, Y. P. Kwon, and M. Tomizuka, "Roarnet: a robust 3D object detection based on region approximation refinement," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2510–2515, Paris, France, 2019.

[39] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Madrid, Spain, 2018.

[40] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from rgb-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927, Salt Lake City, UT, USA, 2018.

[41] B. Yang, M. Liang, and R. Urtasun, "Hdnet: exploiting hd maps for 3D object detection," in *Conference on Robot Learning*, vol. 87, pp. 146–155, Zürich, Switzerland, 2018.

[42] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 924–933, Santa Rosa, CA, USA, 2017.

[43] Z. Ning, S. Sun, X. J. Wang et al., "Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4201–4217, 2022.

[44] X. Wang, Z. Ning, L. Guo, S. Guo, X. Gao, and G. Wang, "Mean-field learning for edge computing in mobile blockchain networks," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2022.

[45] M. Wen, Q. Li, K. J. Kim et al., "Private 5G networks: concepts, architectures, and research landscape," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 1, pp. 7–25, 2022.

[46] M. Wen, E. Basar, Q. Li, B. Zheng, and M. Zhang, "Multiple-mode orthogonal frequency division multiplexing with index modulation," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3892–3906, 2017.