# 3D Object Detection Algorithm Based on Point Cloud Multi-View Fusion

Yuan Liu[1], Zhanlei Fang[1], Yanqiang Li[1, 2, *], Kang Wang[1], Yong Wang[1], Chao Zhang[1]

[1]*Institute of Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China*
[2]*School of Control Sciences and Engineering, Shandong University, Jinan, China*

**Abstract**

The 3D object detection algorithm based on a single view of point cloud has limitations and cannot meet the requirements of complex scenes such as autonomous driving. And most of the existing point cloud multi-view fusion algorithms only focus on two views, and the fusion method is inefficient and simple. In order to coordinate the multiple view representations of point clouds, make full use of the advantages of different views, and alleviate their respective shortcomings in the 3D object detection task, we propose a multi-view fusion detection algorithm, namely PVR-SSD (Point-Voxel-Range Single Stage object Detector). PVR-SSD takes the point-based anchor-free center assignment algorithm as the backbone, and performs point cloud multi-view fusion in two parts. In the downsampling part, a point-range segmentation network is used for selective downsampling to increase the proportion of foreground points, especially small object points. In the feature fusion part, a point-voxel-range feature fusion module is designed, and an attention mechanism is introduced to adaptively aggregate multi-view features with the point-based view as an intermediate carrier. Finally, all-round evaluations on the highly competitive KITTI dataset demonstrate the effectiveness of the proposed algorithm.

**Keywords**

3D Object Detection; Deep Learning; Multi-View Fusion; Attention Mechanism

## 1. Introduction

3D object detection is receiving increasing attention due to its wide range of applications, such as intelligent robotics and autonomous driving. Currently, the most widely used 3D data detector is the Lidar sensor. The raw data obtained by Lidar is represented as a 3D point cloud, which is a collection of points with spatial location information. With some preprocessing, it can be converted to other common view representations, such as Voxel, Range.

The point-based algorithm directly extracts features from the original point cloud, which can retain accurate location information, but the disorder of the point cloud makes its neighborhood search inefficient and computationally expensive. The starting point of both voxel-based and range-based algorithms is to regularize irregular point clouds. The voxel-based algorithm rasterizes the point cloud in 3D space, and then extracts features through 3D sparse convolution. The voxel-based views can effectively preserve physical dimension information, but voxelization inevitably brings information loss, which reduces the fine-grained localization accuracy of  the algorithm. Compared to the voxel-based algorithm, the range-based view has a more compact representation and has no quantization errors, which helps alleviate the sparse problem of point clouds. However, the dimensional compression caused by the 2D projection will inevitably bring about the distortion of the geometric structure and the loss of spatial information[1].

The 3D object detection algorithm based on a single view has different degrees of problems, and it is difficult to achieve a balance between detection accuracy and speed. Using complementary infor-

ICCEIC2022@3rd International Conference on Computer Engineering and Intelligent Control
EMAIL: *Liyq@sdas.org (Yanqiang Li)

mation to preserve strengths and reduce weaknesses is an intuitive solution by combining different views together. Initially, researchers hoped to improve the performance of 3D object detection algorithms by fusing projected views from different perspectives. A certain effect has been achieved, but it still fails to break through the performance limitation of 2D space. The fusion scheme based on point-voxel integration is currently the most widely used. However, these schemes simply perform single-level feature interaction or result-level fusion, do not make full use of the potential relationship between multi-view features, and cannot measure the importance of the respective features of different views. There is still a lot of room for improvement in the 3D object detection algorithm based on multi-view fusion[2].

To sum up, how to effectively coordinate multiple view representations of point clouds, make full use of the advantages of different views, and introduce more effective views to deal with different scenarios is an issue to be studied at present. Therefore, this paper proposes a single-stage anchor-free algorithm PVR-SSD for 3D object detection by fusing different view features of point clouds.

Main contributions in this paper are listed as follows:

·We propose a point cloud segmentation sampling strategy that introduces range-based view features to achieve selective sampling of point clouds and increase the proportion of foreground points.

·We designed a multi-view feature fusion module and introduced a self-attention mechanism, which can adaptively fuse point-based, voxel-based, and range-based view features, effectively improving the accuracy of 3D object detection algorithms.

·We conduct massive experiments on the KITTI da taset to evaluate the effectiveness and efficiency of our proposed method.

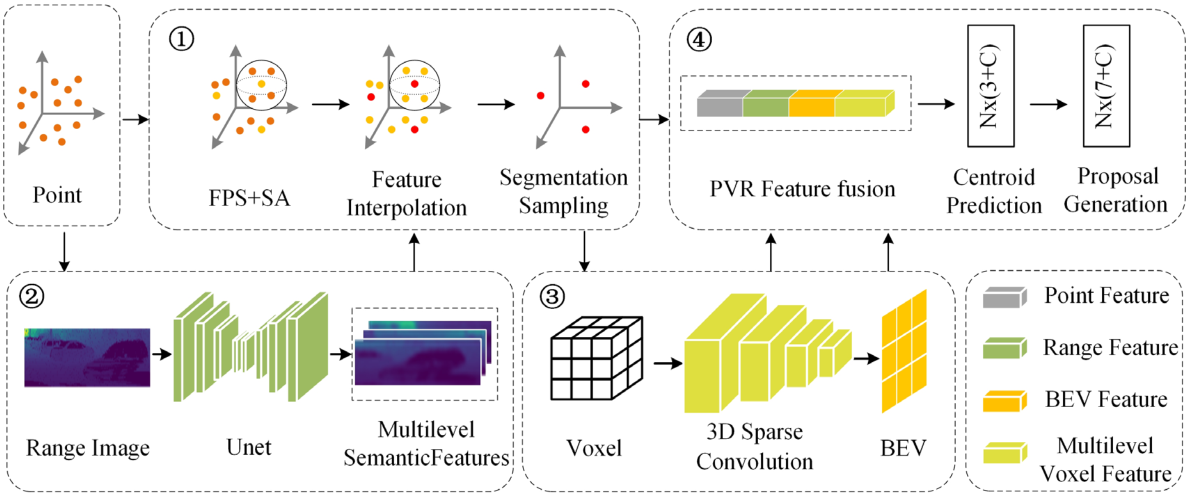

**Figure 1** PVR-SSD Framework Diagram

## 2. Our Framework

### 2.1. Overview

The whole framework of PVR-SSD is shown in Figure 1. It can be divided into four parts, and there is information interaction in multiple parts. The first three parts are the feature extraction networks for different views. The fourth part is the deep fusion of the three view features and the generation of the subsequent 3D bounding box. The point-based feature extraction network is the backbone network of the algorithm. After the original point cloud is input into the network, the FPS is used to perform preliminary preprocessing of the point cloud to reduce the data scale, and the SA module[3] is used for feature extraction. The range-based view branch adopts the lightweight Unet network to extract multi-level semantic features, and then feeds into the segmentation sampling layer in the point branch[4]. The segmentation sampling layer aggregates point-based and range-based features, and uses a point-range foreground point segmentation network for selective downsampling to obtain candidate center points. Then according to the preset confidence threshold, a certain number of original points are reserved and sent to the voxel-based branch of the third part. After voxelization, they are input into the 3D sparse convolutional network[5] to extract multi-scale voxel features, and then highly compressed to obtain BEV feature. The features extracted by the three branches and the candidate center points are sent to the PVR feature fusion module to obtain the candidate point set with the best feature combination. The candidate point set is input into the centeroid prediction layer, and the context clues around the bounding box are merged to obtain the predicted centroid point set. Finally, the predicted centroid points and aggregated features are input into the region proposal generation layer, which predicts the class and regresses to obtain a 3D bounding box.

## 2.2. Point-Range Segmentation Sampling

Common point cloud downsampling algorithms include Random sampling, FPS, Classs-aware sampling, and Centroid-aware sampling. The first two algorithms are based on traditional geometric reasoning ideas, and have good scene coverage, but the computational cost is high, and they treat all points equally, which cannot reflect the importance of foreground points. Classs-aware sampling focuses on sampling to get more foreground points. It is based on point-based views and introduces a separate training branch to learn the latent semantics of each point, enabling selective downsampling. Centroid-aware sampling uses the principle of centrality for weighting based on Classs-aware sampling, aiming to obtain points closer to the center of the instance[2].

Class-aware sampling and centroid-aware sampling achieve good results in foreground point preservation, but the features of only point-based views are exploited. And the sampling performance of the network tends to get more large target points and ignore small target points. The range-based view features are dense, suitable for processing large-scale outdoor scenes, and has low operational complexity. In order to explore the advantages of downsampling with range-based view features, we tested with point-based view, range-based view and point-range fusion view respectively. The test results on the KITTI validation set are shown in Table 1 and Table 2. In order to better compare the performance of several methods, data enhancement processing is performed on the KITTI validation set. It can be seen from the table that the point-range-based method has a similar instance recall rate compared with the point-based method, but the proportion of foreground points of pedestrian and cyclist is higher, indicating that more small target points are sampled on the basis of ensuring instance recall.

Combined with the above analysis, we propose to use a point-range segmentation network for downsampling. As shown in Figure 1, after the point-based and range-based features are extracted respectively, feature propagation and fusion between the two views are realized by establishing a point-range view mapping relationship. The mapping of point-based view to range-based view (P2R) adopts a spherical projection method based on scan expansion[6]. In this way, the data occlusion problem can be avoided and the resulting range-based view is smoother. The feature map can be described by Eq.1.

**Table 1** Instance recall rate (number of instances covered by sampling points/number of all instances)

| views | 1024 | | | 512 | | |
|---|---|---|---|---|---|---|
| | Car | Ped. | Cyc. | Car | Ped. | Cyc. |
| Point | 98.1% | 99.3% | 97.5% | 98.1% | 99.0% | 97.2% |
| Range | 97.5% | 99.1% | 97.2% | 97.9% | 98.3% | 95.6% |
| Point+Range | 98.5% | 99.4% | 97.6% | 98.2% | 99.3% | 97.9% |

**Table 2** Proportion of foreground points (the proportion of objects of different categories in the sampled foreground points)

| views | 1024 | | | 512 | | |
|---|---|---|---|---|---|---|
| | Car | Ped. | Cyc. | Car | Ped. | Cyc. |
| Point | 62.1% | 25.5% | 12.3% | 62.1% | 25.5% | 12.3% |
| Point+Range | 58.7% | 27.7% | 13.5% | 58.7% | 27.7% | 13.5% |

$$rangeimage[\theta, \varphi] = depth \tag{1}$$

where $\theta$ represents the inclination angle, $\varphi$ represents the azimuth angle. The feature map from range-based view to point-based view (R2P) adopts bilinear interpolation. In order to avoid the sampling network being too bloated, the feature fusion of point-range view adopts the method of splicing followed by MLP.

The segmentation sampling strategy uses the above fused features to predict the semantic category of each point through a two-layer scale-invariant MLP. The class loss of points adopts cross entropy loss. In order to get points closer to the center of the instance, we weight the loss function according to the principle of centrality. The weight function is defined as follows:

$$Weight = \sqrt[3]{\frac{\min(f,b)}{\max(f,b)} \times \frac{\min(l,r)}{\max(l,r)} \times \frac{\min(u,d)}{\max(u,d)}} \qquad (2)$$

where *(f,b,l,r,u,d)* represents the distance from the foreground point to the front, rear, left, right, upper and lower faces, respectively. Points within the bounding box are assigned different weights based on their spatial location by multiplying with the loss term for foreground points, thereby assigning higher probability to points closer to the center. The weighted cross-entropy loss is defined as follows:

$$L_{seg} = -\sum_{c=1}^{C} (Weight \cdot s\log(\hat{s}) + (1-s)\log(1-\hat{s})) \qquad (3)$$

where $C$ represents the number of classes, $s$ is the one-hot semantic label, and $\hat{s}$ represents the probability of predicting the semantic class.

## 2.3. Multi-View Fusion

The voxel-based view is inferior to the original point-based view in detection accuracy and range-based view in detection efficiency, but it has the best comprehensive detection performance. The performance of voxel-based algorithm is heavily dependent on the voxel resolution. In order to apply the advantages of voxel-based view and suppress its disadvantages. As shown in part 3 of Fig. 1, we apply a voxel feature extraction network to the segmented foreground points, and then compensate for the sparse problem of voxel features by introducing range-based features.

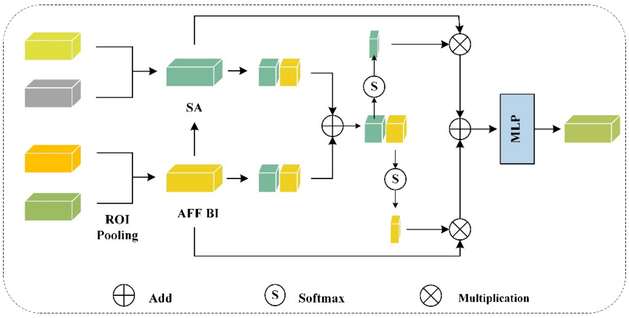

**Figure 2** Point-Voxel-Range Feature Fusion Module

The purpose of multi-view feature fusion is to optimize the combination of features from different views, aggregate key information, and eliminate redundant information. At present, the commonly used feature aggregation methods in the field of 3D object detection are feature addition and feature splicing. This average weighted fusion method cannot measure the importance of the respective features of different views, and inevitably introduces many invalid features. Therefore, we design a multi-view feature fusion module for different views of the same point cloud. The details of feature fusion are shown in Figure 2. Add in the figure represents feature additionand Multiplication represents element-by-element multiplication.

### 2.3.1. Attentional Feature Fusion

BEV features and range-based features are 2D pseudo-image forms. Point-based features and multi-scale voxel features are 3D forms.  Therefore, the modules are first fused pairwise according to the feature form. The multi-scale voxel features are distributed in 3D space, so they are directly gathered on the sampling points through the SA module. The BEV features extracted by the deep network pay more attention to the global information, and the features are relatively sparse due to the number of input points. Range-bsed features contain multi-level semantic information and are relatively dense. In order to better fuse the above two semantically and scale-inconsistent features, we introduce an attention mechanism feature fusion module AFF[7]. The network structure is shown in Figure 3.

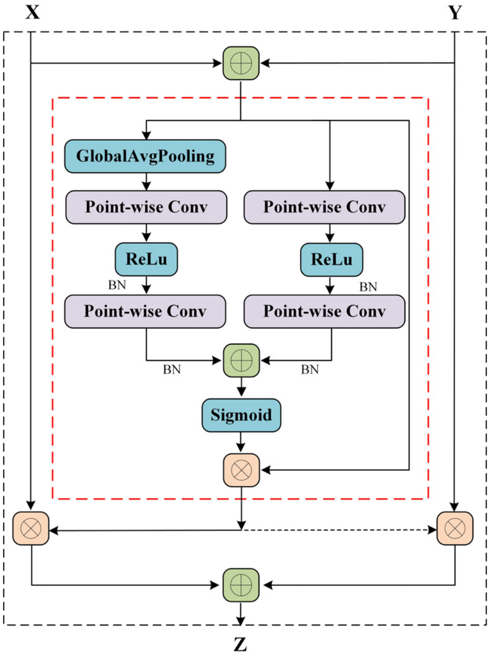

**Figure 3** AFF

It can be expressed by the formula as:

$$Z = M(X \oplus Y) \otimes X + (1 - M(X \oplus Y)) \otimes Y \tag{4}$$

where M(X) represents the network inside the red box, and the dashed arrow represents the 1-M(X) operation. The network uses two branches with different scales to extract the channel attention weights. One of the branches uses Global Avg Pooling to extract the attention of global features, and the other branch directly uses point-wise convolution to extract the channel attention of local features. To keep as lightweight as possible, Point-wise conv is chosen as the local channel context aggregator.

### 2.3.2. Adaptive Fusion

Finally, the features converge into two branches with point views as carriers. In order to eliminate the interference of redundant information and measure the importance of different features, the module maps the features of the two branches respectively to obtain the initial weight vector of each branch. Then, the weights of each branch are voted and added, and the superimposed results are converted into probability weights through softmax. The features of different branches are weighted and then superimposed to obtain the final fusion feature.

It can be expressed as the following formula:

$$F_f = \sum_{i}^{n} split[softmax(\sum_{i}^{n} sigmoid(w_i * F_i))]_i \cdot F_i \tag{5}$$

where $F_i$ represents the branch feature, $w_i$represents the convolution kernel, is used to estimate the initial weight vector of each branch mapping, $\cdot$ represents element-wise multiplication, $n$ represents the number of fusion branches, and corresponds to the number of weight vectors.

## 3. experiment

## 3.1. Experimental Setup

KITTI Dataset is one of the most popular dataset of 3D detection for autonomous driving. We divide the kitti training set into training set and validation set, which contain 3712 frames of data and 3769 frames of data respectively.The IoU threshold settings for different classes of objects are set to 0.7 for cars and 0.5 for pedestrians and cyclists. Out PVR-SSD is trained in an end-to-end manner, using Adam with a single-loop learning strategy for optimization. For the KITTI dataset, we train the entire network with the batch size 16, learning rate 0.01 for 80 epochs on 2 GTX 1080 Ti 10 GB GPUs, which takes around 6 hours.

## 3.2. Experimental Results and Analysis

In the KITTI benchmark, the detection objects are divided into three categories: easy, medium, and difficult according to the difficulty level. We compare the performance and efficiency of PVR-SSD with existing algorithms according to the view representation category, which are recorded in Table 3.

As shown in the table, it can be seen that: 1). In terms of detection accuracy, PVR-SSD outperforms other existing methods in most categories; 2). In terms of detection efficiency, PVR-SSD has faster detection speed than multi-view algorithms; 3).PVR-SSD has a greater improvement in small target detection.

Because with the introduction of the range-based view branch, the sampling network can effectively preserve foreground points and increase the proportion of small object points. And multi-view feature fusion provides multi-dimensional and deep feature representation, eliminating the interference of redundant information, so as to achieve comprehensive and accurate perception of objects of different scales and categories. Despite using the multi-view features, our PVR-SSD still shows a very competitive running latency due to the anchor-free algorithm structure and efficient implementation.

## 3.3. Ablation Studies

### 3.3.1. Downsampling Ablation Experiment

To further verify the effectiveness of the proposed point-range segmentation downsampling method, we replace it with the D-FPS+F-FPS and the Centroid-aware sampling. As shown in Table 4, the proposed sampling method achieves the best detection performance in all three categories. Especially for small objects such as pedestrians and cyclists, the detection performance is significantly improved after adding the range-based view feature. This shows information provided by range-based view features can expand the receptive field, effectively preserve foreground information during downsampling, and improve the proportion of small objects, thereby achieving better detection performance.

**Table 3** Detection performance comparison on KITTI dataset.

| Method | 3D Car（IoU=0.7） | 3D Ped.（IoU=0.5） | 3D Cyc.（IoU=0.5） |
|---|---|---|---|
| D-FPS + F-FPS | 79.37 | 40.27 | 61.30 |
| Centroid-aware | 80.42 | 41.02 | 62.84 |
| Point + Range | 81.36 | 45.53 | 65.76 |

**Table 4** Ablation studies of PVR-SSD on different sampling strategies

| | Method | Reference | GPU | 3D Car (IoU=0.7）<br>Easy Mod. Hard | 3D Ped. (IoU=0.5)<br>Easy Mod. Hard | 3D Cyc. (IoU=0.5)<br>Easy Mod. Hard | Speed<br>（s） |
|---|---|---|---|---|---|---|---|
| Point | PointRCNN | CVPR 2019 | TITAN XP | 86.96 75.64 70.70 | 47.98 39.37 36.01 | 74.96 58.82 52.53 | 0.1 |
| | 3DSSD | CVPR 2020 | TITAN V | 88.36 79.57 74.55 | 54.64 44.27 40.23 | 82.48 64.10 56.90 | 0.04 |
| | IA-SSD | CVPR 2022 | RTX 2080Ti | 88.34 80.13 75.04 | 46.51 39.03 35.60 | 78.35 61.94 55.70 | **0.013** |
| Voxel | SECOND | Sensors 2018 | GTX 1080Ti | 84.65 75.96 68.71 | 45.31 35.52 33.14 | 75.83 60.82 53.67 | 0.04 |
| | PointPillars | CVPR 2019 | GTX 1080Ti | 82.58 74.31 68.99 | 51.45 41.92 38.89 | 77.10 58.65 51.92 | 0.016 |
| Range | RangeRCNN | CVPR 2021 | Tesla V100 | 88.47 81.33 77.09 | - - - | - - - | 0.06 |
| Multi - View | STD | ICCV 2019 | TITAN V | 87.95 79.71 75.09 | 53.29 42.47 38.35 | 78.69 61.59 55.30 | 0.08 |
| | PVRCNN | CVPR 2020 | GTX 1080Ti | 90.25 **81.43** 76.82 | 52.17 43.29 40.29 | 78.60 63.71 57.65 | 0.08 |
| | HVPR | CVPR 2021 | GTX 2080Ti | 86.38 77.92 73.04 | 53.47 43.96 40.64 | - - - | -- |
| | **ours** | - | GTX 1080Ti | **91.32** 81.36 **77.16** | **56.69 45.53 42.26** | **84.61 65.76 59.39** | 0.05 |
| | Improvement | - | - | +1.07 -0.07 +0.07 | +2.05 +1.26 +1.62 | +2.13 +1.66 +1.74 | +0.037 |

### 3.3.2. Fusion Ablation Experiment

To verify the effectiveness of the feature fusion approach in Figure 2, we investigate the effect of changing the fusion style on multi-view features, as shown in Table 5. It can be seen that there are considerable improvements compared to these two methods. And the addition of features does not improve the detection accuracy, indicating that redundant features are harmful to the network, which proves the necessity of the feature fusion strategy in this paper.

**Table 5** Ablation studies of PVR-SSD on different fusion strategies

| Method | 3D Car<br>（IoU=0.7） | 3D Ped.<br>（IoU=0.5） | 3D Cyc.<br>（IoU=0.5） |
|---|---|---|---|
| Add | 79.73 | 38.63 | 58.94 |
| Cat | 80.46 | 41.43 | 63.52 |
| ours | 81.36 | 45.53 | 65.76 |

## 4. Conclusion

This paper proposes a point-voxel-range view fusion detection algorithm, namely PVR-SSD. The algorithm proposes a segmentation sampling strategy and an adaptive multi-view fusion method, and conducts extensive experiments on the open source dataset KITTI, and achieves good detection results. But the algorithm still has room for further research. In addition to point, voxel, and range views, point clouds have some other views, and real scene data can not only be captured with point clouds, but other data sources such as images can also be regarded as samples of the real physical world. In the future, more in-depth research can be done from the perspective of other views or multi-source data.

This work was supported by the National Natural Science Foundation of China (52072212), Natural Science Foundation of Shandong Province (ZR2021MF103).

## 5. Reference

[1] Yang, Z., Sun, Y., Liu, S. and Jia, J., 2020. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11040-11048).

[2] Zhang Y , Hu Q , Xu G , et al. Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds[J]. 2022.

[3] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems. 2017;30.

[4] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InInternational Conference on Medical image computing and computer-assisted intervention 2015 Oct 5 (pp. 234-241). Springer, Cham.

[5] Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection. Sensors. 2018 Oct 6;18(10):3337.

[6] Fan L, Xiong X, Wang F, Wang N, Zhang Z. Rangedet: In defense of range view for lidar-based 3d object detection. InProceedings of the IEEE/CVF International Conference on Computer Vision 2021 (pp. 2918-2927).

[7] Dai Y, Gieseke F, Oehmcke S, Wu Y, Barnard K. Attentional feature fusion. InProceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2021 (pp. 3560-3569).