




Article

Anti-Noise 3D Object Detection of Multimodal Feature Attention Fusion Based on PV-RCNN

Yuan Zhu ¹, Ruidong Xu ¹ , Hao An ¹ , Chongben Tao ² and Ke Lu ^{1,*} ¹ School of Automotive Studies, Tongji University, Shanghai 201800, China² Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215200, China

* Correspondence: luke@tongji.edu.cn; Tel.: +86-136-6197-8920

Abstract: 3D object detection methods based on camera and LiDAR fusion are susceptible to environmental noise. Due to the mismatch of physical characteristics of the two sensors, the feature vectors encoded by the feature layer are in different feature spaces. This leads to the problem of feature information deviation, which has an impact on detection performance. To address this problem, a point-guided feature abstract method is presented to fuse the camera and LiDAR at first. The extracted image features and point cloud features are aggregated to keypoints for enhancing information redundancy. Second, the proposed multimodal feature attention (MFA) mechanism is used to achieve adaptive fusion of point cloud features and image features with information from multiple feature spaces. Finally, a projection-based farthest point sampling (P-FPS) is proposed to downsample the raw point cloud, which can project more keypoints onto the close object and improve the sampling rate of the point-guided image features. The 3D bounding boxes of the object is obtained by the region of interest (ROI) pooling layer and the fully connected layer. The proposed 3D object detection algorithm is evaluated on three different datasets, and the proposed algorithm achieved better detection performance and robustness when the image and point cloud data contain rain noise. The test results on a physical test platform further validate the effectiveness of the algorithm.

Keywords: 3D object detection; multimodal feature; anti-noise; attention mechanism

Citation: Zhu, Y.; Xu, R.; An, H.; Tao, C.; Lu, K. Anti-Noise 3D Object Detection of Multimodal Feature Attention Fusion Based on PV-RCNN. *Sensors* **2023**, *23*, 233. <https://doi.org/10.3390/s23010233>

Academic Editors: Shiyang Tang, Yan Huang, Zhanye Chen and Ping Guo

Received: 9 November 2022

Revised: 14 December 2022

Accepted: 22 December 2022

Published: 26 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reliable 3D object detection is the primary step to ensure the proper working of autonomous driving systems. Current autonomous driving systems commonly use camera and LiDAR for 3D object detection in order to get more accurate detection results and more applicable scenarios. The camera can output high-resolution pictures, providing rich color and texture information, but it is not easy to extract the spatial position of the object. LiDAR can provide accurate spatial point cloud information, therefore most 3D object detection algorithms [1–5] are based on LiDAR and achieve better detection performance. However, both sensors are susceptible to weather noise: the camera is susceptible to image texture noise and the LiDAR is susceptible to spatial point coordinate noise [6–9]. This paper focuses on rainy scenarios, as it is the most common dynamic challenging weather condition that effects vision sensors. These effects can directly blur object colors and contours in images, and produce multiple reflections to affect the observability of spatial boundaries in point clouds, which seriously degrades the performance of the object detector. Therefore, effectively fusing the information provided by both sensors is an important approach to improve the robustness of environment perception. In this paper, we propose a camera and LiDAR fusion 3D object detection algorithm based on point-guided sampling of point cloud spatial features and image semantic features, which shows significant performance improvement in rain noise datasets.

In recent years, there are some studies on 3D object detection based on camera and LiDAR fusion, which can be classified into serial type [10–16] and parallel type [17–38]

according to the stage of fusion. The serial type method is represented by F-PointNet [13] which usually takes the image of the camera as input first and uses image object detection or semantic segmentation algorithm to get the spatial location of the object, then projects it to the LiDAR point cloud to extract the point cloud of the frustum region around the object, and finally uses the normal point cloud 3D object detection algorithm to get 3D bounding boxes. F-PointPillars [10] replaced the point cloud detection network with PointPillars on this basis and improved the detection speed. F-ConvNet [14] encoded the grouping within the frustum region and achieved better detection accuracy. This type of method can effectively reduce the number of point clouds and improve the computational efficiency, but requires high image detection performance of the front. The parallel type method, on the other hand, starts with MV3D [30], which extracts the image features of the LiDAR point cloud and camera separately, and then fuses them at different stages, and finally obtains the 3D object detection results through the fully connected layer. Another approach of parallel type methods is to introduce the correspondence between spatial points and pixel points into the feature fusion, which can improve the feature correlation. Both MVP [31] and EPNet [20] take advantage of this and achieve better detection performance. Most of the subsequent parallel type methods use this fusion method, which requires the establishment of a projection relationship between spatial and pixel points by sensor calibration, and then fusing the point or voxel features with the image features of the corresponding pixel points. However, since the point cloud is sparse while the image has higher pixel point density, it will cause the problems of resolution mismatch and uneven sampling of image features. Therefore, this paper proposes a P-FPS algorithm to project the point cloud onto the image plane and then downsample the point cloud by farthest point sampling (FPS). This method can improve the sampling rate of image features for close objects.

Attention-based methods [39] can aggregate information from different feature spaces, and DETR [40] introduced self-attention into the field of image-based 2D object detection. Some studies [41–43] since then have attempted to apply attention mechanisms to image-based perception, and others [44–47] have used attention mechanisms in point cloud-based applications. For image and point cloud feature fusion, the heterogeneity of the two sensors can lead to the features obtained by the feature extraction network not to be in the same feature space. In order to enable the point cloud features to fuse the most associated image features, an MFA module is proposed in this paper. This module allows point cloud features to be adaptively fused with image features from different feature spaces.

The contributions of this paper are as follows:

1. Aiming at the mismatch resolution between point cloud and image, a point-guided 3D object detection algorithm based on point cloud and image fusion is proposed. The algorithm is based on the point-guided strategy of PV-RCNN, which fuses the multi-stage features of two sensors and enriches the raw point cloud features. Compared with other 3D object detection algorithms, this algorithm achieves significant robustness when point clouds and images contain rain noise, and can utilize image and point cloud information more sufficiently.
2. An MFA module is proposed for the problem of fused information interference caused by different feature spaces of images and point clouds. This module uses an attention mechanism to calculate the attention weights of multiple image features and point cloud features corresponding to each point. Based on the correlation between point cloud features and spatial features, different image features are fused adaptively.
3. To solve the problem of insufficient image features of close objects extracted from point clouds obtained by FPS, a P-FPS algorithm is proposed. By projecting the LiDAR point cloud onto the image plane before implementing FPS, uniform sampling of the point cloud in the image plane is achieved and the image feature sampling rate of the object is improved.

The remainder of this paper is as follows: In Section 2, the related background research work is discussed. The proposed algorithmic framework and key modules are presented

in detail in Section 3. Section 4 describes the experimental setup and experimental results. Finally, Section 5 presents an ablation study of the method proposed in this paper.

2. Related Work

Image-driven fusion. Currently, camera-based 2D object detection has been able to achieve an average accuracy of over 90% in publicly available datasets. However, in 3D object detection, better results can be achieved by using LiDAR. Some studies have attempted to utilize the advantage of both sensors and used a serial form of 2D detection followed by 3D detection. Qi et al. [13] proposed F-PointNets for RGB-D data, and used Fast R-CNN to generate 2D bounding boxes and delineate a frustum region in 3D space, and then performed 3D detection of point clouds in this region based on PointNet. This approach reduces the computational requirements for point cloud search on the one hand, and reduces the interference of background point clouds on detection on the other hand. However, the frustum region obtained by this method will still contain background points behind the object; Wang et al. [14] made further improvements to address this problem by using sliding regions for point cloud feature extraction separately. To solve the heterogeneity problem in camera-LiDAR fusion, some works inspired by Pseudo-LiDAR [48] fuse the pseudo point cloud generated from the camera with the point cloud acquired by LiDAR. Pseudo-LiDAR++ [49], based on Pseudo-LiDAR [48], uses the binocular camera disparity estimation principle to generate pseudo point clouds and improve the quality of pseudo point clouds by matching them with LiDAR point clouds through K-Nearest Neighbor (KNN).

Feature fusion. Unlike the image-driven approach, MV3D [30] and AVOD [26] use the projection map of the LiDAR point cloud and the camera image as inputs, extract image and point cloud features separately, and fuse the proposal regions to obtain the final 3D detection results, avoiding the dependence on the camera detection performance. 3D-CVF [17] first transforms image features from multiple perspective to the Bird Eye View (BEV) plane, and then selects the fusion degree of the two types features by an adaptive gate fusion network. ContFuse [50] uses continuous convolution to aggregate images and BEV features at different resolutions based on geometric position relationships. This method uses dense image features to enrich the sparse LiDAR point cloud to enhance the 3D detection performance. These methods ignore the matching between features, although they use multi-layer fusion.

Project-based fusion. Fusing complementary information from point clouds and images would benefit 3D detection. However, the inherent heterogeneous nature between sensors created difficulties in information fusion, and many studies projected point clouds onto the images for data sampling. If these data are processed directly, some problems are encountered: (1) sparse point clouds do not correspond to high-resolution image features one by one, resulting in incomplete data fusion; (2) point cloud data are missing depth or height information after projection. To address these problems, many attempts have been made in recent studies. Zhu et al. [37] proposed VPFNet, a network that first generates 3D proposal regions from point clouds and divides them into multiple small grids again, with the corner points of each grid as virtual points, and next projects these points onto the image for image feature sampling and point cloud feature extraction, which reduces the computation and compensates the lack of sparse point clouds. Huang et al. [20] proposed a LI-Fusion module for fusing point cloud and image information at different scales in the feature extraction stage, and then associating the image features to the corresponding point cloud locations, while adding a consistency forcing loss for balancing the impact of classification loss and location loss on the detection results. Wen et al. [36] combined the fusion framework of MV3D [30] and AVOD [26], and proposed to project the height information of the point cloud into the image to obtain RGB-D data, and then fused it with the bird's-eye view to construct a single-stage detector while using an adaptive attention mechanism in feature extraction. Projection-based fusion methods usually capture image features based on keypoints after FPS downsampling. Due to the affine transformation, the

point cloud is no longer uniform in the image plane, and the feature-rich close objects cannot be extracted completely. Therefore, this paper proposes to project the point cloud onto the image plane followed by FPS in order to sample uniform image features. Meanwhile, unlike the concat fusion of previous methods, an MFA module is proposed to make the point cloud adaptively fuse image features with different weights.

3. Framework of Proposed 3D Detection Algorithm

For the above problems, this paper proposes a 3D object detection method that fuses LiDAR and camera. The method can adaptively fuse image and point cloud features based on the attention mechanism to cope with the performance degradation when the point cloud or image contain rain noise. As shown in Figure 1, the network is based on PV-RCNN [1], and image features are fused into the raw LiDAR features. When using cameras and LiDAR for 3D object detection, the difference in the spatial location of the two sensors can cause the results to be in different reference coordinate systems. To associate the point cloud and image features, a projection of the point cloud to the image plane is used to map each point to the corresponding pixel location. The projection of the point $P_L(x, y, z)$ of the LiDAR to the pixel point $P_I(u, v)$ is expressed as:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KM} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

where f is the camera focal length, dx and dy are the length and width of a single pixel at the image plane, u_0 and v_0 are offset of the pixel coordinate system, R is rotation matrix, T is translation matrix, z_c is the depth of the point to the image plane, K is the camera internal reference matrix, and M is the transformation matrix from the LiDAR coordinate system to the camera coordinate system.

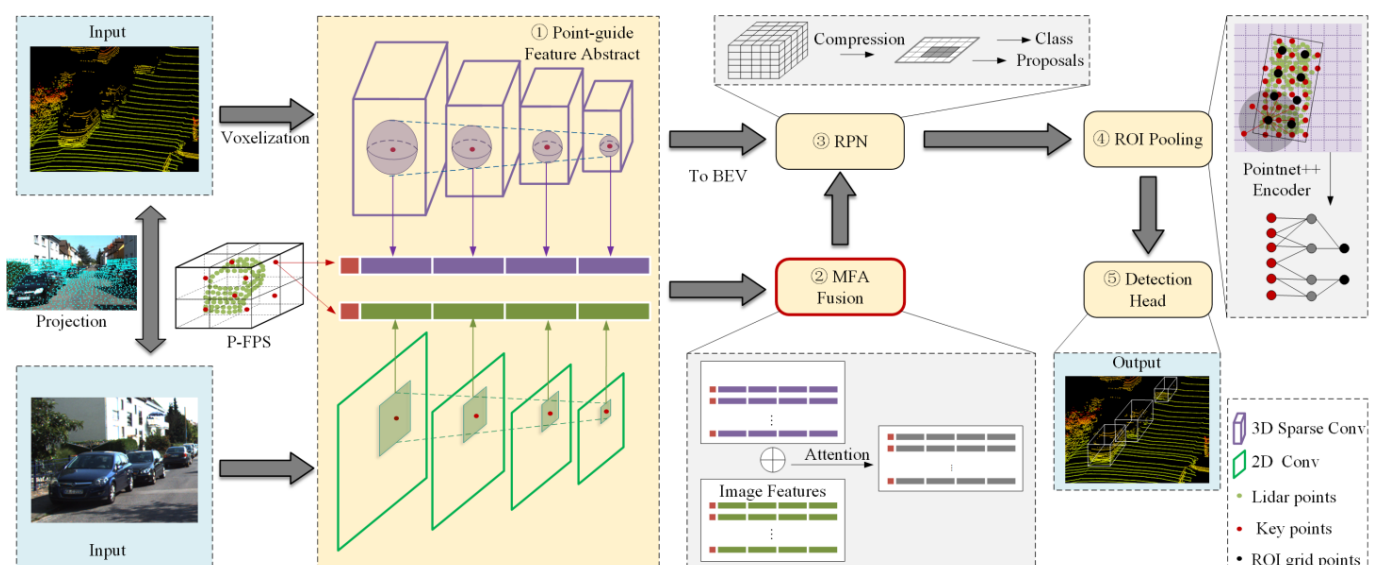


Figure 1. The architecture of the proposed 3D object detection.

By keypoints guidance, point cloud features and image features of different scales can be merged into the same point at the same time. Due to the heterogeneity of images and

point clouds, the two types of features are not in the same feature space. To address this problem, the proposed MFA module can effectively fuse the two types of features to get fused features. Subsequently, the fusion features guided by keypoints are processed by the region proposal network (RPN) module to get ROIs. Finally, ROIs are further reduced by ROI pooling, and the location and class information of the object is obtained by the refinement network.

3.1. P-FPS

Many point-based LiDAR 3D object detection algorithms [51,52] downsample the raw point cloud by FPS to get the keypoints to reduce the computational effort. Projection-based fusion algorithms [20,53] will project the LiDAR point cloud onto the image and often use this downsampling method to process the raw point cloud. However, FPS uniformly samples point clouds in 3D space, which results in uneven distribution of point clouds when projected onto an image. As shown in Figure 2, distant objects contain little feature information because they contain relatively few pixels. In contrast, the close object contains more pixels and can provide rich features. The key points obtained by FPS are mostly distributed in the distance. The image features sampled in this way lose considerable detail.

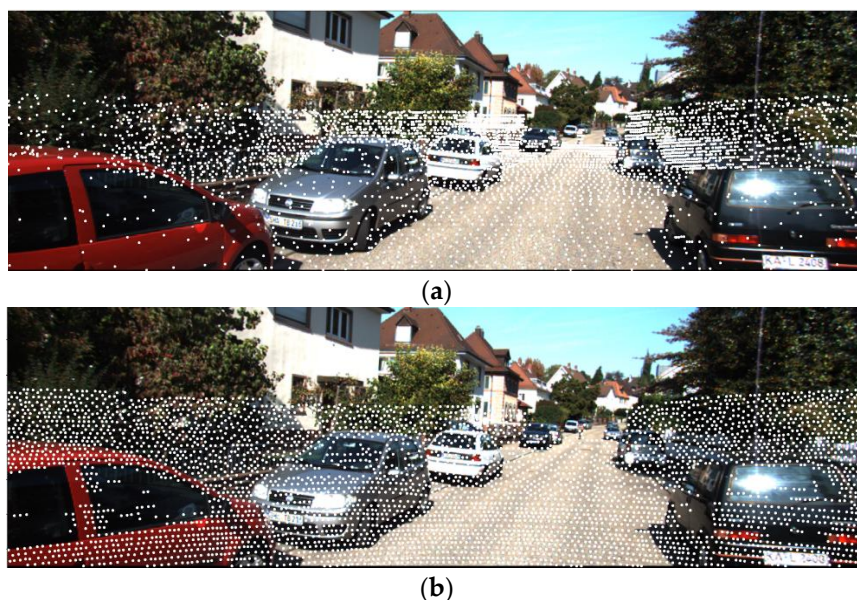


Figure 2. Comparison of (a) FPS and (b) P-FPS.

To solve the above problems, a P-FPS is proposed for point cloud sampling. As shown in Figure 3, suppose there are M 3D spatial coordinates $P_{Li}(x_i, y_i, z_i) \in R^L, i = 1, 2 \dots M$ in the raw point cloud, which can be projected onto the image plane by Equation (2) to obtain the corresponding 2D pixel coordinates $P_{Ii}(u_i, v_i) \in R^I, i = 1, 2 \dots M$. In 3D space, the FPS requires the calculation of the 3D Euclidean distance between the coordinates of two points, P_{Li} and P_{Lj} . Correspondingly, in the image plane, the 2D Euclidean distance D_I between the pixel coordinates P_{Ii} and P_{Ij} is required to be calculated, which can be expressed as:

$$D_I = \|F_p(x_i, y_i, z_i) - F_p(x_j, y_j, z_j)\|_2, i, j = 1, 2 \dots M \quad (3)$$

where $F_p(\cdot)$ is the projection transformation relationship from the point cloud to the image plane, which can be obtained from Equation (2). After the image plane is processed by 2D FPS, the close object can contain more points. With the index information of these points, the keypoints in 3D space can be obtained. The specific content of the algorithm is shown in Algorithm 1.

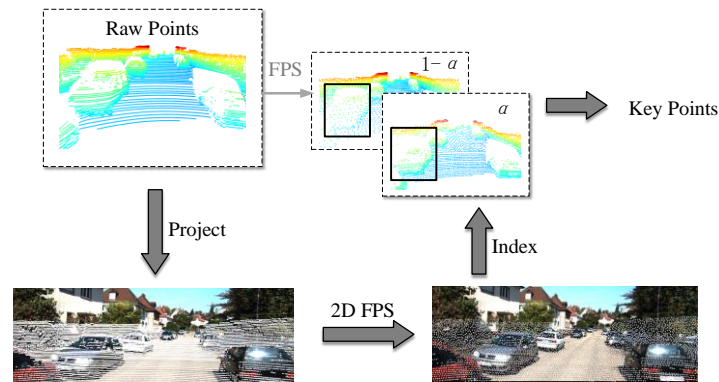


Figure 3. Illustration of the P-FPS.

The sampling method above allows closer points to be obtained, but results in a slight reduction of distant points. To balance this loss, the number of points obtained by FPS and 2D-FPS is divided by the following equation:

$$Num_{P-FPS} = [\alpha \cdot Num_{2DFPS} + (1 - \alpha) \cdot Num_{FPS}] \quad (4)$$

Algorithm 1 2D FPS

Input : Raw point set $P_{Li} \in R^L, i = 1, 2 \dots M$

Output : Keypoints set $P_{Kj} \in R^L, j = 1, 2 \dots T$

```

1  Suppose  $t$  keypoints has been sampled and  $(t + 1)$ th keypoint is sampled as follows
2  for  $i = 0$  to  $(M - t)$  by 1 do
3    for  $j = 0$  to  $t$  by 1 do
4      Calculate the distance between point  $P_{Li}$  and point  $P_{Kj}$ :  $D_I = \|F_p(P_{Li}) - F_p(P_{Kj})\|_2$ 
5      if  $D_I <$  last minimum dist then
6        dist =  $D_I$ 
7      end if
8    end for
9  end for
10 Select the point with minimum dist as the keypoint
11 return Keypoints set

```

3.2. Point-Guide Feature Abstract

3D Feature Abstraction. The raw point cloud is first divided into many voxels of size $d \times w \times h$. The feature vector of each non-empty voxel is encoded as:

$$V_i = \left\{ \frac{\sum x_i}{Point_num}, \frac{\sum y_i}{Point_num}, \frac{\sum z_i}{Point_num}, \frac{\sum r_i}{Point_num} \right\} \quad (5)$$

where $Point_num$ is the number of points contained within each voxel and r_i is the reflection intensity of that point. Then, the 3D sparse convolution of $3 \times 3 \times 3$ is used to obtain spatial features at four different downsampling scales ($1 \times, 2 \times, 4 \times, 8 \times$).

2D Feature Abstraction. Image features contain rich texture and color information. In order to be consistent with 3D feature extraction, ResNet-18 [54] is utilized to extract image features. The network uses a series of 3×3 2D convolutional kernels to downsample the feature maps at four different scales. The four scales of feature maps focus on different feature details, and the same downsample scale as the 3D backbone is adopted.

Point Feature Sample. After obtaining T keypoints $P_K = \{p_{K1}, \dots, p_{Kj}\}, j = 1, 2 \dots T$ by P-FPS, each point will aggregate 3D and 2D features, respectively. For 3D features,

each scale feature is first sampled using ball query. Suppose that the k th layer Voxel-wise features are expressed as:

$$V^k = \{v_1^k, \dots, v_N^k\} \quad (6)$$

The 3D coordinates of the k th layer are expressed as:

$$V^c = \{v_1^c, \dots, v_N^c\} \quad (7)$$

The feature vectors of all voxels within the radius r are encoded as:

$$F_i^r = \left\{ \left[v_i^k; v_i^c - p_{K_i} \right]^T \middle| \begin{array}{l} \|v_i^c - p_{K_i}\|^2 < r \\ \forall v_i^c \in V^c, \forall v_i^k \in V^k \end{array} \right\} \quad (8)$$

where $v_i^c - p_{K_i}$ denotes the local space features. Subsequently, these voxel features are aggregated to the keypoint p_{K_i} as follows:

$$F_i^{3Dk} = \max\{MLP(G(F_i^r))\} \quad (9)$$

where $G(\cdot)$ denotes the random sampling of t voxels features. The feature vectors of all 4 scales are concatenated to the keypoint p_{K_i} as follows:

$$F_i^{3D} = [F_i^{3D1}, F_i^{3D2}, F_i^{3D3}, F_i^{3D4}] \quad (10)$$

For 2D features, the keypoints cannot be accurately corresponded to individual pixel points, although they have been projected onto the pixel plane. The reason is that the projection point coordinates are continuous, while the pixel point coordinates are discrete. Therefore, bilinear interpolation is employed to sample the four pixels around the projection point, and this process can be expressed as:

$$F_i^{2Dk} = \left\{ B \left(F_{i1}^{lk}, F_{i2}^{lk}, F_{i3}^{lk}, F_{i4}^{lk} \right) \right\} \quad (11)$$

where B denotes the bilinear interpolation function and $F_{i1}^{lk}, F_{i2}^{lk}, F_{i3}^{lk}, F_{i4}^{lk}$ denote the feature vectors of the four neighboring pixel points around the projection point. Finally, as with the 3D features, the 2D features are stitched to the same keypoint p_{K_i}

$$F_i^{2D} = [F_i^{2D1}, F_i^{2D2}, F_i^{2D3}, F_i^{2D4}] \quad (12)$$

3.3. MFA Fusion

Since DETR [40] introduced the self-attention transformer [39] into the field of object detection, many studies [36,41,42,45] have attempted to use the attention mechanism for 2D or 3D object detection. Most of these approaches focus on spatial attention, and in this paper, we propose an MFA module based on the attention of feature channels. The purpose of this module is to allow point cloud features to adaptively fuse image features based on feature correlations. This module improves the robustness of the detection algorithm when the image or point cloud contain noise.

As shown in Figure 4, for a keypoint p_{K_i} , the image features F_i^{2Di} at the k th layer scale are encoded by two sets of N 1D convolution kernels to obtain N Image value V_{in}^k and Image key K_{in}^k , respectively, while the point cloud features F_{in}^{3Dk} are encoded by 1D convolution kernels to obtain Point query Q_i^k . Each feature point is sampled to obtain a 1D feature vector, therefore a 1D convolution kernel is used to encode the image feature vector and the point cloud feature vector to obtain the same feature space. The vector length of each layer is set to [32,32,64,128] after the 1D convolution kernel encoding.

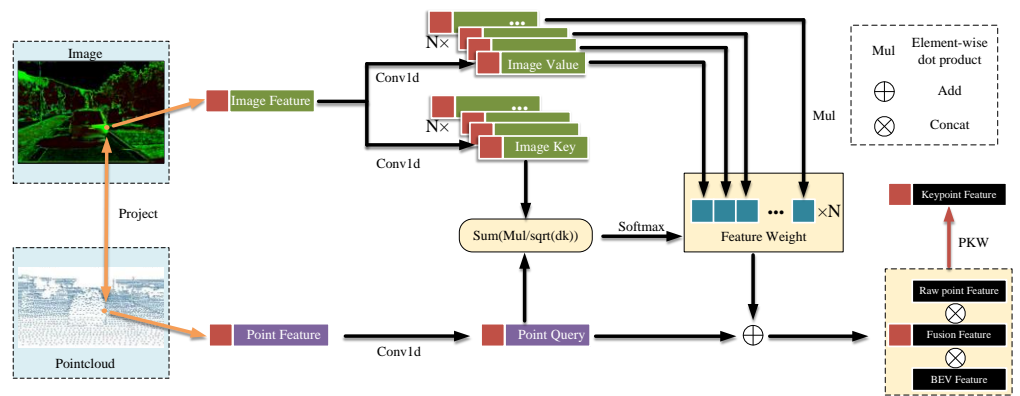


Figure 4. Multimodal Feature Attention (MFA) fusion module.

In the spatial self-attention mechanism of DETR [40], a positional encoding is added to the feature vector to enable the network to learn positional features. In the MFA module, the feature vectors used for attention calculation come from the same location, thus there is no need to add position encoding. Image key and Point query are used to calculate the feature weights. The correlation vector RE^k and feature weight W^k are obtained by summation of the dot product, and the formula is as follows:

$$RE^k = \begin{bmatrix} re_1^k \\ re_2^k \\ \dots \\ re_j^k \end{bmatrix} = \begin{bmatrix} \text{SUM}(Q_i^k \cdot K_{i1}^k) \\ \text{SUM}(Q_i^k \cdot K_{i2}^k) \\ \dots \\ \text{SUM}(Q_i^k \cdot K_{ij}^k) \end{bmatrix} \frac{1}{\sqrt{d_k}}, j = 1, \dots, N \quad (13)$$

$$W^k = \text{Softmax}(RE^k) = \begin{bmatrix} w_1^k \\ w_2^k \\ \dots \\ w_j^k \end{bmatrix}, j = 1, \dots, N \quad (14)$$

where d_k is the dimension of Image key and Point query, which is used as a scale factor to prevent extreme values after the summation of the dot product. Then the softmax function is used to score the N elements in the correlation vector RE_j^k to get the feature weight W^k , which represents the correlation between different image features and point cloud features. Feature weight is weighted and summed with N Image values V_{in}^k to obtain the final image feature vector, which is summed with Point query Q_i^k to obtain the final fusion feature vector F_i^{Fk} .

$$F_i^{Fk} = J(w_j^k \cdot V_{ij}^k) + Q_i^k \quad (15)$$

where $J(\cdot)$ denotes the summation of the dot product.

The raw point-cloud feature is aggregated as in Equation (8). For the BEV, we project the keypoint to the 2D bird-view coordinate system, and utilize bilinear interpolation to obtain the features from the bird-view feature. Finally, to enrich the features, each keypoint-guided feature vector is joined with feature vectors representing the raw point cloud and BEV. It can be expressed as:

$$F_i^k = [F_i^{F1}; F_i^{F2}; F_i^{F3}; F_i^{F4}; F_i^{raw}; F_i^{BEV}] \quad (16)$$

After downsampling by P-FPS, although more points fall within the object, there are still many background points that do not contribute much to the detection task. Therefore, following the design of PV-RCNN, the predicted keypoint weighting (PKW) module is used to predict the probability of each keypoint being a foreground point. The difference is

that the keypoint features processed by PKW contain information from images and point clouds. The process of predicting keypoints can be expressed as:

$$\tilde{F}_i^p = \mathcal{A}(F_i^k) \cdot F_i^k \quad (17)$$

where $\mathcal{A}(\cdot)$ denotes a three-layer MLP network with a sigmoid function to predict foreground confidence between [0, 1].

3.4. ROI Pooling and Detection Head

The RPN structure is the same as PV-RCNN, where many ROIs are predicted based on 3D spatial features. As shown in Figure 5, the keypoints feature vectors contained within each ROI are aggregated to a fixed $6 \times 6 \times 6$ grid point in order to obtain a more accurate prediction of the bounding boxes. The process is similar to the point feature abstraction process, where the keypoints around the grid points are sampled by using ball queries at different scales. The final ROI contains 216 grid point feature vectors within each ROI, and then the class and 3D bounding boxes of the object are predicted by two layers of MLP.

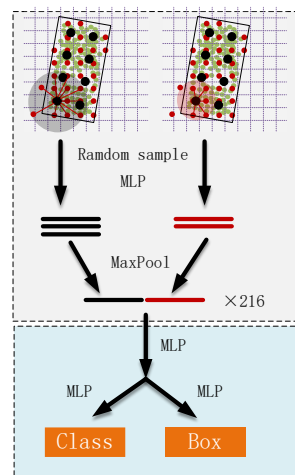


Figure 5. Region of Interest (ROI) pooling module and detection head.

4. Experiments

4.1. Datasets

KITTI [55] is a common benchmark for outdoor 3D object detection. There are 7481 training samples and 7518 test samples, where the training samples are generally divided into the train split (3712 samples) and the val split (3769 samples). In the 3D object detection task, Car, Pedestrian and Cyclist were evaluated separately. KITTI presents three object detection tasks (Easy, Moderate and Hard) of different difficulty according to the size of the object, the degree of occlusion and the degree of truncation. In the experimental results, we adopted “Easy”, “Mod.” and “Hard” to represent these three tasks.

Most of the KITTI data are collected in good weather, however there are many real scenarios with bad weather (e.g., rain) that can affect the accuracy of LiDAR and camera outputs. To simulate the rainy scenarios, two noises are added separately. The camera is often influenced by rain which causes blurring and occlusion of the output image, resulting in the loss of detailed features of the object [7]. In this paper, this blurring is simulated using Gaussian blurring as follows:

$$\begin{cases} F(u) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u^2}{2\sigma_u^2}} \\ F(v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{v^2}{2\sigma_v^2}} \end{cases} \quad (18)$$

where u, v denote the local coordinates of the Gaussian kernel and σ_u^2, σ_v^2 denote the variance in both directions. The Gaussian kernel is a two-dimensional normally distributed matrix containing a set of pixel weights that can smooth the pixels inside the Gaussian kernel. For the occlusion effect of raindrops, we randomly generated different raindrop traces through image processing.

LiDAR is prone to noise in rainy environments [56,57], because raindrops can cause multiple reflections to LiDAR. The result of this is that the 3D contour boundary of the object is less obvious. Random noise is added to the 3D coordinates of the raw point cloud to simulate this effect.

Figure 6 shows the effect after adding the noise. It can be observed that the vehicle textures in the images become blurred and the vehicle contours in the point cloud are no longer regular. These changes increase the difficulty of object detection because the object specific features are weakened.

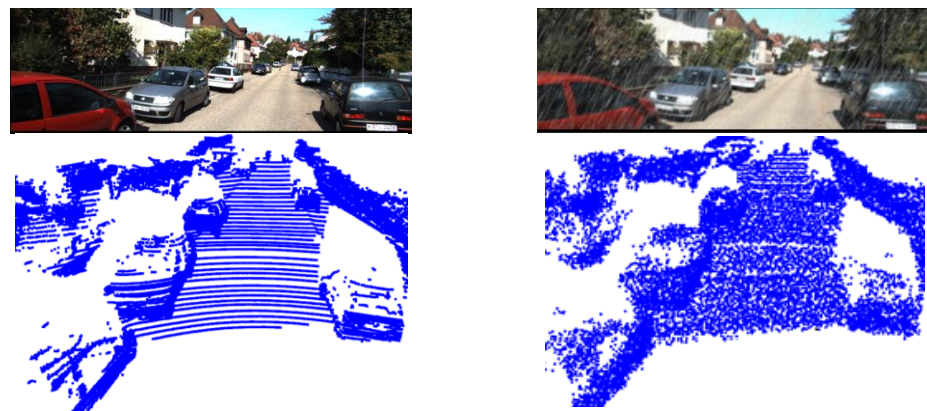


Figure 6. The effect of adding rain noise to the data.

4.2. Training and Inference Details

Since the camera FOV is in front of the vehicle, this paper takes the camera FOV as the boundary and removes the point clouds beyond the boundary. Meanwhile, a rectangular area is set to filter the remaining point clouds. The range of this region is ($0 \text{ m} < x < 70.4 \text{ m}$, $-40 \text{ m} < y < 40 \text{ m}$, $-1 \text{ m} < z < 3 \text{ m}$), which is divided into many voxels of size (0.05 m, 0.05 m, 0.1 m).

For training, both the 2D backbone and 3D backbone use pretrained parameters directly, and the parameters are not updated. In addition, since there is a fixed transformation relationship between LiDAR and camera, and the fusion data augmentation scheme is not mature, conventional data augmentation methods are not used in this paper. The network is built using OpenPCDet and trained on a GTX3090 with batchsize 4, epoch 80, and learning rate 0.01.

4.3. Results on the KITTI Dataset with Rain Noise

The mainstream object detection algorithms are tested on the generated KITTI with rain noise. Table 1 shows the test results of each algorithm in a KITTI validation split with rain noise. For better evaluation, all algorithms are retrained on data with rain noise, and the same training method is used. As can be seen in Table 1, the proposed algorithm shows the best results in all three detection tasks of Car, Pedestrian and Cyclist. Compared with the baseline algorithm PV-RCNN in this paper, the mAP is improved by 5.86%, 21.15% and 41.17% in the three categories of the 3D detection task, respectively. Meanwhile, Table 2 demonstrated that the proposed algorithm is still the optimal result in the BEV task. Compared with PV-RCNN, the mAPs of the three categories are improved by 3.76%, 16.80% and 40.00%, respectively. In the detection tasks of Pedestrian and Cyclist, the algorithm in this paper shows a relatively large lead. The reason is that Pedestrian and Cyclist contain few point clouds and their features are more affected by rain noise. It causes difficulties in

point cloud-based algorithms in both training and detection, and the proposed algorithm can aggregate image features as a complement to enrich the overall feature information.

Table 1. Results of 3D detection on the KITTI validation split with rain noise after training.

Modality	Car				Pedestrian				Cyclist			
	Easy	Mod.	Hard	mAp	Easy	Mod.	Hard	mAp	Easy	Mod.	Hard	mAp
PV-RCNN [1]	85.32	74.28	69.96	76.52	46.71	41.71	38.52	42.31	61.23	40.13	36.51	45.96
PointRCNN [4]	85.59	76.74	72.58	78.30	47.90	40.54	34.47	40.97	80.44	51.49	49.15	60.36
PartA2	88.91	77.31	74.57	80.26	56.79	49.38	43.17	49.78	76.31	51.72	47.44	58.49
SECOND [2]	80.24	67.93	64.45	70.87	41.82	35.37	31.68	36.29	61.32	42.46	40.43	48.07
PointPillars [5]	74.58	61.73	59.70	65.34	32.79	28.00	24.51	28.43	55.66	35.45	33.41	41.51
EPNet [20]	87.96	76.53	75.24	79.91	-	-	-	-	-	-	-	-
MFA fusion (Ours)	88.24	77.95	76.85	81.01	56.82	50.49	46.47	51.26	84.46	60.20	57.59	64.88

Table 2. Results of Bird’s Eye View (BEV) on the KITTI validation split with rain noise after training.

Modality	Car				Pedestrian				Cyclist			
	Easy	Mod.	Hard	mAp	Easy	Mod.	Hard	mAp	Easy	Mod.	Hard	mAp
PV-RCNN [1]	92.12	83.95	79.39	85.15	52.93	47.45	44.21	48.20	62.70	41.15	39.00	47.61
PointRCNN [4]	91.95	84.26	81.97	86.06	54.37	44.82	38.44	45.88	81.09	53.97	49.54	61.53
PartA2	92.87	84.13	83.63	86.88	63.44	55.51	48.96	55.97	77.37	54.59	50.23	60.73
SECOND [2]	89.93	82.01	79.95	83.96	51.59	44.25	40.22	45.35	64.20	45.96	43.78	51.31
PointPillars [5]	89.76	79.51	77.30	82.19	40.67	34.10	31.53	35.43	58.23	37.61	35.59	43.81
EPNet [20]	92.16	85.89	84.27	87.44	-	-	-	-	-	-	-	-
MFA fusion (Ours)	92.49	86.31	86.24	88.35	61.85	55.62	51.43	56.30	81.19	60.74	58.02	66.65

Figure 7 shows the comparison of PR curves for different detection algorithms on KITTI with rain noise. Recall is the proportion of the number of objects that are correctly detected among all ground truths. The precision in the Figure 7 indicates the percentage of detected objects that are correctly detected at different recall rates. From the figure, it can be observed that the proposed algorithm achieves the highest accuracy rate at the recall rate of [1–0.6]. The high accuracy is also maintained at the recall rate of [0.6–0].

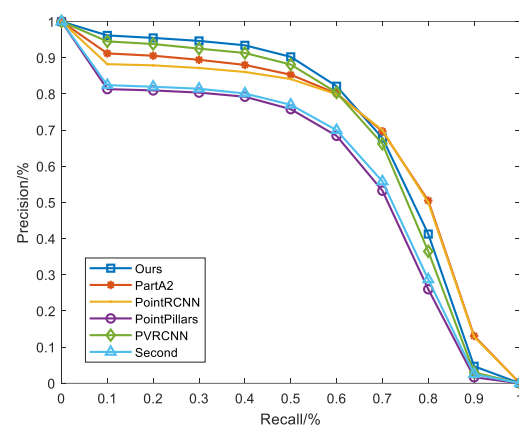


Figure 7. Precision–recall curve.

In order to compare the robustness of the algorithms to the effects of rain noise, pre-trained models are directly used for testing on noisy datasets. The proposed algorithm is trained on the raw KITTI without rain noise. As can be seen in Table 3, the proposed algorithm still achieves the best results even without training on noisy data.

Table 3. Results of 3D detection on the KITTI val split with rain noise before training for the Car category.

	Modality	Car			mAp
		Easy	Mod.	Hard	
PV-RCNN [1]		34.90	31.00	29.07	31.66
PointRCNN [4]		27.04	25.61	23.54	25.40
PartA2	L	20.02	21.64	20.24	20.63
SECOND [2]		36.76	33.30	31.91	33.99
PointPillars [5]		30.82	27.87	27.26	28.65
MFA fusion (Ours)	L + C	37.07	33.73	31.43	34.08

4.4. Results on the NUSCENES Dataset

To validate the effectiveness of the proposed algorithm, it was evaluated and compared on the NUSCENES dataset. The dataset uses six cameras, one 32-line LiDAR, five millimeter wave radars, IMU and GPS with 1000 different urban scenes, including scenes with visual impact such as night and rain. As shown in Table 4, the proposed algorithm in the NUSCENES dataset achieves a competitive performance. The proposed algorithm achieves the highest results in the mAP metric. The rainy and night scenes contained in the dataset affect the camera more, and the proposed algorithm adaptively learns the association of image features with point cloud features in different feature spaces, thus achieving a high robustness.

Table 4. Results of detection on the NUSCENES dataset.

	mATE	mASE	mAOE	mAVE	mAAE	mAP	NDS
PointPillars [5]	33.87	26.00	32.07	28.74	20.15	44.63	58.23
SECOND [2]	31.15	25.51	26.64	26.26	20.46	50.59	62.29
CenterPoint [58]	30.11	25.55	38.28	21.94	18.87	56.03	64.54
MFA fusion (Ours)	35.56	28.33	35.29	29.75	22.46	57.54	60.51

4.5. Results on Physical Test Platform

As shown in Figure 8, a physical test platform was built to verify the algorithm proposed in this paper. The platform is installed with a 32-line LiDAR, an IMU, a monocular camera, and a 77 GHz millimeter wave radar. Data from all sensors were collected by VECTOR VN5640 and aligned to the same timestamp with same time source. The proposed algorithm was tested on LiDAR and camera data, which were converted to the format of the KITTI dataset.

**Figure 8.** Physical test platform.

As shown in Figure 9, a slice of data collected by the actual physical test platform shows that the proposed algorithm obtains better 3D detection performance. Our algorithm and PV-RCNN are trained on KITTI data with noise, while noise is added on the actual test data. The results in the first row are output by the camera, the results in the second row are

obtained by the proposed algorithm, and the results in the third row are obtained by the PV-RCNN. It can be observed that the proposed algorithm is able to detect more objects. It is important to note that testing the algorithm on the collected data is a cross-domain problem. This is due to the fact that the KITTI sensor models and scenarios are not the same as the collected data, and random noise simulating the effects of weather is added during the testing. The proposed algorithm incorporates information from different stage features and different feature spaces to effectively alleviate this problem.

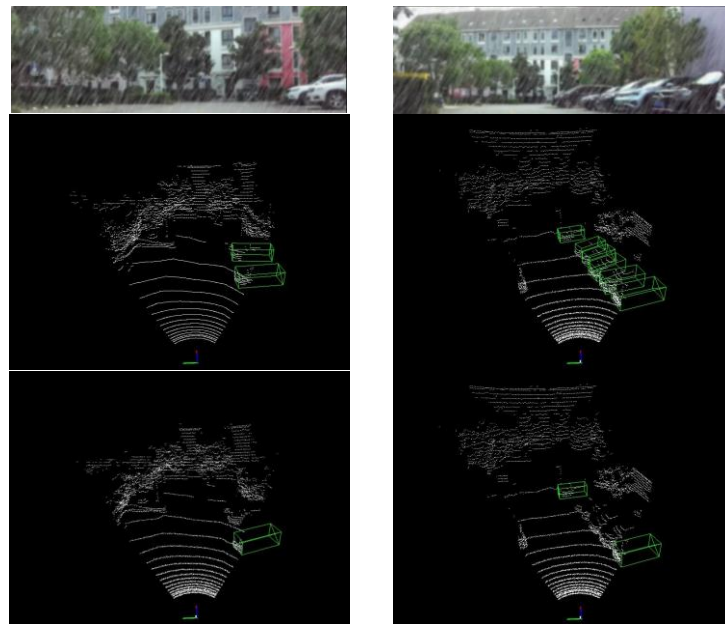


Figure 9. Visual results of physical test platform dataset.

5. Ablation Studies

5.1. On P-FPS

To study whether the points sampled by the proposed P-FPS contribute to the object detection task, the recall of ground truth points is used to evaluate the results. The object bounding box provided by the ground truth is used to determine whether the points sampled by P-FPS and FPS belong to the foreground points. As shown in Figure 10, the number of sampled points falling into the ground truth object box by P-FPS is approximately three times higher than that by FPS for the same setting of keypoints number. From the results, it is clear that the proposed P-FPS can sample more points belonging to the close objects, which is helpful for both the effectiveness of image feature sampling and the subsequent object bounding boxed prediction stage.

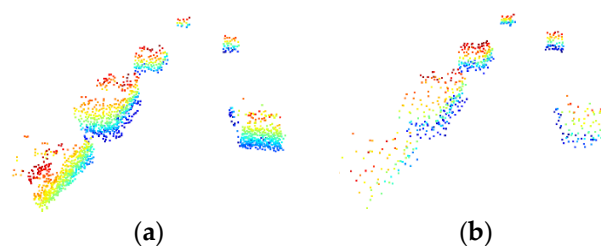


Figure 10. Comparison of P-FPS and FPS on recalling GT points: (a) P-FPS (recall 1507 points); (b) FPS (recall 575 points).

While increasing the number of close object points, the distant points should not be ignored. Therefore, different proportions are assigned for the sampling points number

of FPS and 2D-FPS by Formula (4). As shown in Figure 11, the effects of different α are compared on the detection performance, and three categories of mAP are used to evaluate the detection results. It can be seen that the proposed method achieves the optimal detection performance at $\alpha = 0.3$. When α decreases, the sampling results tend to be FPS, and the keypoints obtained do not contain closer object points. On the contrary, when α increases, the sampling results tend to be P-FPS, and the keypoints lose the features of distant objects. Table 5 shows the 3D detection results of the proposed sampling method compared with the traditional FPS method. The proposed sampling algorithm is substantially better than FPS in Car and Pedestrian, and obtains competitive results in Cyclist.

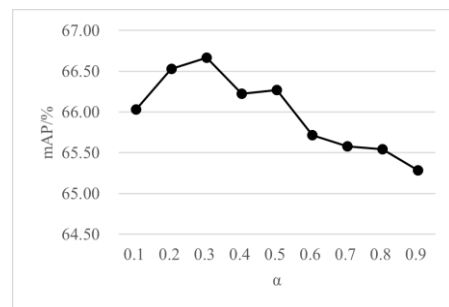


Figure 11. Performance comparisons on different α .

Table 5. Results of 3D detection on the rain noise data with different sample methods.

	Car				Pedestrian				Cyclist			
	Easy	Mod	Hard	mAp	Easy	Mod	Hard	mAp	Easy	Mod	Hard	mAp
FPS	88.14	76.58	76.23	80.32	56.15	50.32	44.82	50.43	83.61	61.89	57.38	67.63
P-FPS	88.24	77.95	76.85	81.01	56.82	50.49	46.47	51.26	84.46	60.20	57.59	67.42

5.2. On Fusion Feature

PKW module can predict the probability of keypoints being foreground points, and the accuracy of this result will directly affect the subsequent 3D bounding box prediction. Keypoints with predicted probability greater than 0.05 are marked as true positives. It can be observed from Figure 12a,b that by fusing the feature information of the image, the foreground points predicted by the PKW module are more accurate. Conversely, if only LiDAR point cloud features are used, the PKW module will predict more background points. The results confirm that the PKW module will assign more weights to the foreground points by using the fused features, which further reduces the influence of background points.

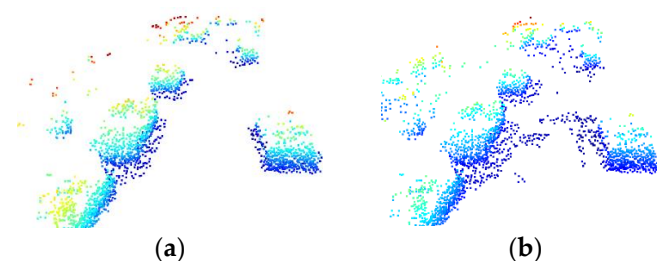


Figure 12. Effect of fusion feature on Predicted Keypoint Weighting (PKW). (a) Foreground point prediction with fusion feature. (b) Foreground point prediction without fusion feature.

The detection performance of fusing features at different stages is evaluated separately. As shown in Table 6, the first 4 rows are the result of using only one of the features in $\{F_i^{F1}, F_i^{F2}, F_i^{F3}, F_i^{F4}\}$. It can be observed that the F_i^{F1}, F_i^{F2} features obtain higher detection

accuracy, which indicates that low-dimensional features contribute more to the detection algorithm. The last row adopts the fused features of all stages and obtains the best detection accuracy.

Table 6. Effects of different feature components.

F_i^{F1}	F_i^{F2}	F_i^{F3}	F_i^{F4}	mAP (Car)
✓				80.54
	✓			79.91
		✓		79.25
			✓	77.36
✓	✓	✓	✓	81.01

5.3. On MFA

In the attention mechanism, multiple heads are usually used to fuse representations from different feature spaces. The detection performances of the proposed algorithm with different numbers of heads are evaluated. As shown in Figure 13, in the Car category, the best detection performance is achieved when the number of heads is 4. When the number of heads decreases or increases, the detection accuracy decreases.

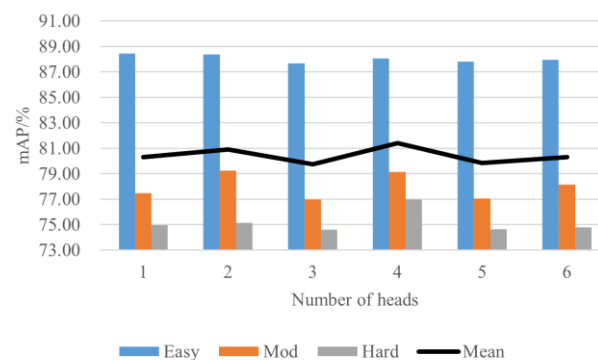


Figure 13. Performance comparison on the car category with different heads.

The image features contain rich texture features. In order to verify the effectiveness of these features in improving the detection performance in rain scenarios, the image features are removed and compared with the proposed algorithm. It can be seen from Table 7 that only using point cloud features will significantly reduce the detection performance of the algorithm for car category. This suggests that even if the image features become blurred, there is still a non-negligible contribution to the detection accuracy.

Table 7. The suppression effect of fusion features under rain noise.

Feature	Easy	Mod	Hard	mAP (Car)
Only LiDAR	88.05	74.21	70.49	77.58
MFA fusion	88.24	77.95	76.85	81.01

To visually demonstrate the attention effect of MFA, the attention results of each keypoint are visualized in Figure 14. The first row in the figure is the raw image output by the camera, the second row is the attention difference of each feature point, and the third row is the detection result. Each keypoint aggregates image features in four different feature spaces through the MFA module, and these features are assigned different weights by the attention mechanism. The different colors shown in the Figure 14 represent the image features with the largest weight among the four kinds of image features fused by the keypoint. It can be observed that the proposed method adaptively selects different

image feature fusion weights at different locations in the image. There is a clear boundary difference between most objects and backgrounds. This shows that the MFA module can selectively fuse image features according to the illumination changes and texture details of the image.

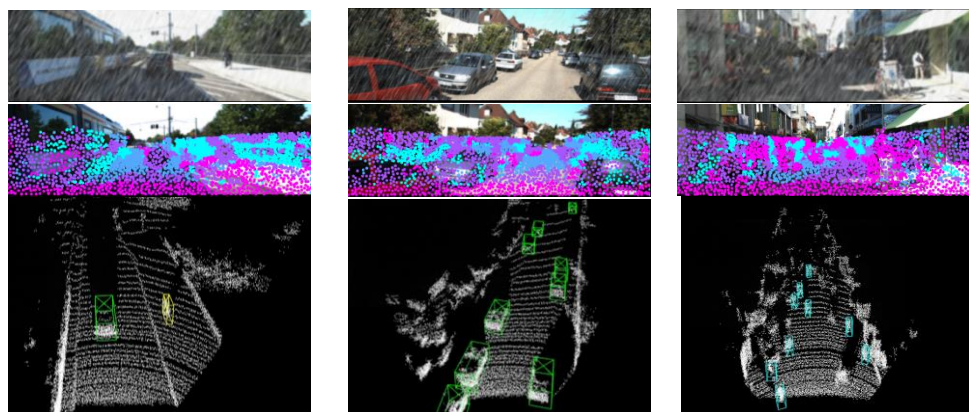


Figure 14. Visualization of MFA. Green/blue/yellow boxes indicate Car/Pedestrian/Cyclist.

6. Conclusions

In this paper, a new camera and LiDAR-based 3D object detection algorithm is proposed, which is based on point-guided sampling of point cloud features and image features. The multi-head attention mechanism is introduced into the feature fusion task, and the proposed MFA mechanism is used to realize the adaptive fusion of point cloud features and image features. A projection-based sampling method, P-FPS, is proposed, which can significantly improve the sampling rate of image features and obtains more abundant image information to the fused features. In some scenes, the proposed F-FPS can extract about three times the number of effective feature points, compared with traditional FPS. In order to simulate the scene where the sensor contain rain noise, the KITTI dataset with rain noise is established. The proposed method shows better detection performance and robustness when tested on KITTI and NUSCENES datasets. Compared with the baseline algorithm PV-RCNN in this paper, the mAP is improved by 5.86%, 21.15% and 41.17% in the three categories of the 3D detection task, respectively. Finally, the algorithm is tested on the physical test platform, which further verifies the effectiveness of the algorithm. In future work, more real extreme weather data will be collected, and weather simulation tests in virtual environments will be used to test the robustness of the algorithm. In addition, the domain adaptation problem caused by the different distributions of the training dataset and the testing dataset will also be studied to improve the generalization of the feature extraction network.

Author Contributions: In this article, the author’s contributions are shown below: conceptualization, Y.Z. and R.X.; methodology, Y.Z., H.A. and R.X.; formal analysis, H.A.; supervision, K.L.; writing—original draft, R.X.; writing—review and editing, C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: <https://www.cvlibs.net/datasets/kitti/> and <https://www.nuscenes.org/nuscenes>.

Acknowledgments: We appreciate the critical and constructive comments and suggestions from the reviewers that helped improve the quality of this manuscript. We also would like to offer our sincere

thanks to those who participated in the data processing and provided constructive comments for this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10526–10535.
2. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [[CrossRef](#)] [[PubMed](#)]
3. Luo, X.; Zhou, F.; Tao, C.; Yang, A.; Zhang, P.; Chen, Y. Dynamic Multitarget Detection Algorithm of Voxel Point Cloud Fusion Based on PointRCNN. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 20707–20720. [[CrossRef](#)]
4. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
5. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection From Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
6. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; Seattle, WA, USA, 13–19 June 2020; pp. 11682–11692.
7. Hnewa, M.; Radha, H. Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques. *IEEE Signal Process. Mag.* **2021**, *38*, 53–67. [[CrossRef](#)]
8. Aldibaja, M.; Kuramoto, A.; Yanase, R.; Kim, T.H.; Yonada, K.; Sukanuma, N. Lateral Road-Mark Reconstruction Using Neural Network for Safe Autonomous Driving in Snow-Wet Environments. In Proceedings of the 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Shenyang, China, 24–27 August 2018; pp. 486–493.
9. Wang, W.; You, X.; Chen, L.; Tian, J.; Tang, F.; Zhang, L. A Scalable and Accurate De-Snowing Algorithm for LiDAR Point Clouds in Winter. *Remote Sens.* **2022**, *14*, 1468. [[CrossRef](#)]
10. Paigwar, A.; Sierra-Gonzalez, D.; Erkent, Ö.; Laugier, C. Frustum-PointPillars: A Multi-Stage Approach for 3D Object Detection Using RGB Camera and LiDAR. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2926–2933.
11. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4604–4612.
12. Shin, K.; Kwon, Y.P.; Tomizuka, M. RoarNet: A Robust 3D Object Detection Based on RegiOn Approximation Refinement. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2510–2515.
13. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D Object Detection From RGB-D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 918–927.
14. Wang, Z.; Jia, K. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 1742–1749.
15. Tao, C.; He, H.; Xu, F.; Cao, J. Stereo Priori RCNN Based Car Detection on Point Level for Autonomous Driving. *Knowl. Based Syst.* **2021**, *229*, 107346. [[CrossRef](#)]
16. Kashevnik, A.; Ali, A. 3D Vehicle Detection and Segmentation Based on EfficientNetB3 and CenterNet Residual Blocks. *Sensors* **2022**, *22*, 7990. [[CrossRef](#)] [[PubMed](#)]
17. Yoo, J.H.; Kim, Y.; Kim, J.; Choi, J.W. 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-View Spatial Feature Fusion for 3D Object Detection. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 720–736.
18. Piergiovanni, A.J.; Casser, V.; Ryoo, M.S.; Angelova, A. 4D-Net for Learned Multi-Modal Alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15435–15445.
19. Qian, R.; Garg, D.; Wang, Y.; You, Y.; Belongie, S.; Hariharan, B.; Campbell, M.; Weinberger, K.Q.; Chao, W.-L. End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5881–5890.
20. Huang, T.; Liu, Z.; Chen, X.; Bai, X. EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 35–52.
21. Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; Jia, J. Focal Sparse Convolutional Networks for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5428–5437.
22. Wen, L.; Jo, K.-H. Fully Convolutional Neural Networks for 3D Vehicle Detection Based on Point Clouds. In *Proceedings of the Intelligent Computing Theories and Application*; Huang, D.-S., Jo, K.-H., Huang, Z.-K., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 592–601.

23. Rashed, H.; Ramzy, M.; Vaquero, V.; El Sallab, A.; Sistu, G.; Yogamani, S. FuseMODNet: Real-Time Camera and LiDAR Based Moving Object Detection for Robust Low-Light Autonomous Driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019; pp. 2393–2402.
24. Wang, Z.; Zhan, W.; Tomizuka, M. Fusing Bird’s Eye View LIDAR Point Cloud and Front View Camera Image for 3D Object Detection. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1–6.
25. Wang, L.; Chen, T.; Anklam, C.; Goldluecke, B. High Dimensional Frustum PointNet for 3D Object Detection from Camera, LiDAR, and Radar. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1621–1628.
26. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S. Joint 3D Proposal Generation and Object Detection from View Aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 12 July 2018; pp. 1–8.
27. Wen, L.; Jo, K.-H. LiDAR-Camera-Based Deep Dense Fusion for Robust 3D Object Detection. In *Proceedings of the Intelligent Computing Methodologies*; Huang, D.-S., Premaratne, P., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 133–144.
28. Wang, Z.; Zhao, Z.; Jin, Z.; Che, Z.; Tang, J.; Shen, C.; Peng, Y. Multi-Stage Fusion for Multi-Class 3D Lidar Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 3120–3128.
29. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-Task Multi-Sensor Fusion for 3D Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7337–7345.
30. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-View 3D Object Detection Network for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
31. Yin, T.; Zhou, X.; Krähenbühl, P. Multimodal Virtual Point 3D Detection. *Adv. Neural Inf. Processing Syst.* **2021**, *34*, 16494–16507.
32. Sindagi, V.A.; Zhou, Y.; Tuzel, O. MVX-Net: Multimodal VoxelNet for 3D Object Detection. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 7276–7282.
33. Wang, C.; Ma, C.; Zhu, M.; Yang, X. PointAugmenting: Cross-Modal Augmentation for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11794–11803.
34. Xu, D.; Anguelov, D.; Jain, A. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 244–253.
35. Li, P.; Chen, X.; Shen, S. Stereo R-CNN Based 3D Object Detection for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7644–7652.
36. Wen, L.-H.; Jo, K.-H. Three-Attention Mechanisms for One-Stage 3-D Object Detection Based on LiDAR and Camera. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6655–6663. [[CrossRef](#)]
37. Zhu, H.; Deng, J.; Zhang, Y.; Ji, J.; Mao, Q.; Li, H.; Zhang, Y. VPFNet: Improving 3D Object Detection with Virtual Point Based LiDAR and Stereo Data Fusion. *IEEE Trans. Multimed.* **2022**, 1–14. [[CrossRef](#)]
38. Geng, K.; Dong, G.; Yin, G.; Hu, J. Deep Dual-Modal Traffic Objects Instance Segmentation Method Using Camera and LIDAR Data for Autonomous Driving. *Remote Sens.* **2020**, *12*, 3274. [[CrossRef](#)]
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
40. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
41. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 10 February 2022.
42. Wang, Y.; Guizilini, V.C.; Zhang, T.; Wang, Y.; Zhao, H.; Solomon, J. DETR3D: 3D Object Detection from Multi-View Images via 3D-to-2D Queries. In Proceedings of the 5th Conference on Robot Learning, London, UK, 11 January 2022; pp. 180–191.
43. Cao, J.; Song, C.; Song, S.; Xiao, F.; Zhang, X.; Liu, Z.; Ang, M.H. Robust Object Tracking Algorithm for Autonomous Vehicles in Complex Scenes. *Remote Sens.* **2021**, *13*, 3234. [[CrossRef](#)]
44. Zhao, L.; Xu, S.; Liu, L.; Ming, D.; Tao, W. SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation. *Remote Sens.* **2022**, *14*, 4471. [[CrossRef](#)]
45. Misra, I.; Girdhar, R.; Joulin, A. An End-to-End Transformer Model for 3D Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 2906–2917.
46. Zhai, Z.; Wang, Q.; Pan, Z.; Gao, Z.; Hu, W. Multi-Frame Point Cloud Feature Fusion Based on Attention Mechanisms for 3D Object Detection. *Sensors* **2022**, *22*, 7473. [[CrossRef](#)] [[PubMed](#)]
47. Liu, M.; Ma, J.; Zheng, Q.; Liu, Y.; Shi, G. 3D Object Detection Based on Attention and Multi-Scale Feature Fusion. *Sensors* **2022**, *22*, 3935. [[CrossRef](#)] [[PubMed](#)]
48. Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8445–8453.

49. You, Y.; Wang, Y.; Chao, W.-L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. In Proceedings of the Eighth International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
50. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
51. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 7 June 2017; Volume 30.
52. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3D Object Detection With Pointformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
53. Zhang, Y.; Chen, J.; Huang, D. CAT-Det: Contrastively Augmented Transformer for Multi-Modal 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 908–917.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
55. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
56. Zhang, C.; Huang, Z.; Ang, M.H.; Rus, D. LiDAR Degradation Quantification for Autonomous Driving in Rain. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3458–3464.
57. Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W. Weather Influence and Classification with Automotive Lidar Sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1527–1534.
58. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-Based 3D Object Detection and Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.