Advanced Engineering Informatics 56 (2023) 101971

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Full length article

An object detection algorithm combining semantic and geometric information of the 3D point cloud[☆],[☆☆]

Zhe Huang [a], Yongcai Wang [a,*], Jie Wen [b], Peng Wang [a,b], Xudong Cai [a]

[a] School of Information, Renmin University of China, Beijing, 100872, China
[b] China Waterborne Transport Research Institute, Beijing, China

ARTICLE INFO

Keywords:
Deep learning
3D object detection
Point cloud
Geometric information
Autonomous vehicle

ABSTRACT

Accurately detect vehicles or pedestrians from 3D point clouds (3D object detection) is a fast developing research topic in autonomous driving and other domains. The fundamental component for feature extraction in 3D object detection is Set Abstraction (SA), which can downsample points while aggregating points to extract features. However, the current SA ignores the geometric and semantic properties of point clouds and may miss to detect remote small objects. In this paper, FocusSA is proposed, which consists two modules for enhancing useful feature extraction in the SA layer to improve 3D object detection accuracy. At first, Focused FPS (FocFPS) is proposed to evaluate the foreground and boundary scores of the points and reweighs the Furthest Point Sampling (FPS) using the evaluated scores to retain more contextual points in downsampling. Then a Geometry-aware Feature Extraction (GeoFE) module is proposed to add geometric information to enrich the awareness of geometric structure in feature aggregation. To evaluate the performances of the proposed methods, we conduct extensive experiments on three difficulty levels of Car class in KITTI dataset. The experimental results show that on "moderate" instances, our results outperform the state-of-the-art method by 1.08%. Moreover, FocusSA is easy to be plugged in popular architectures.

1. Introduction

Autonomous driving technologies are inspiring new applications in transportation engineering, autonomous logistics distribution, unmanned retail and shared travel [1–3] etc. In autonomous driving, accurately identifying and locating pedestrians and cars in 3D scenes is the most fundamental problem [4,5]. Point clouds are common 3D data format that can be captured by cars using Lidar sensor or by Visual Odometry methods [6]. Since the point cloud provides accurate (x, y, z) coordinates of the points, an object's 3D position can be easily inferred if the object is correctly detected [7]. So object detection from 3D point clouds attracts great research attentions. However, due to the intricate geometric feature and the discrete point structure, object detection from point cloud is not easy.

User interested objects are often represented as 3D models in early engineering applications. Targets are therefore detected by using segmentation, rendering, matching, and other approaches. These approaches, however, are computationally difficult, inefficient, and expensive, which are hard to be used in autonomous driving. With the great success of deep learning for autonomous and effective feature extraction, researchers have exploited deep learning [8] for 3D object detection.

The goal of 3D object detection is to extract the closest 3D bounding boxes around the user-interested objects, such as the vehicles or pedestrians. There are mainly three categories of approaches. The first category, known as multi-view based method, which converts point clouds from sparse formation to compact representation by projecting them to Bird's Eye View (BEV) or Front View (FV) [9]. However, in most cases, this strategy results in the loss of geometric information and local structures in the 3D point cloud. To solve this problem, The second category of method called voxel-based, which borrows the idea from image processing. By transforming point clouds into regular voxel grids, they change the sparse formation to compact representations by subdividing them to distributed voxels [10]. The third category, i.e., point-based methods, directly perform feature learning from the 3D points [11]. The latent features in the unbalanced point clouds are extracted via point cloud down-sampling and feature aggregation

[☆] This document is the results of the research project funded by the National Science Foundation.

[☆☆] This work was supported in part by the National Key R&D Program of China 2020YFB1710801, National Natural Science Foundation of China Grant No. 61972404, 12071478, Public Computing Cloud, Renmin University of China.

[*] Corresponding author.
E-mail address: ycw@ruc.edu.cn (Y. Wang).

https://doi.org/10.1016/j.aei.2023.101971
Received 3 February 2023; Received in revised form 29 March 2023; Accepted 8 April 2023
Available online 19 April 2023

1474-0346/© 2023 Published by Elsevier Ltd.

from the raw input point clouds. The point-based down-sampling and aggregation make point-based methods have flexible perception field and are more suitable for processing naturally unbalanced points.

Despite of these advantages, the object detection accuracy of point-based methods are still lower than that of voxel-based approaches. By further investigating the Set Abstraction (SA) layers of the point-based object detection networks, we found that there are still three limitations in these methods.

(1) Existing methods have not effectively extracted the boundary context. Because the key of the 3D object detection is to extract the features from their surrounding points, pinpointing the objects' boundaries is crucial for object detection [12]. Actually, the boundary points can be effectively evaluated by comparing neighboring points' features. Then the point downsampling process can focus more on the boundary points.

(2) The relative geometry relationships among the points are not encoded [13] in existing feature aggregation step. Since the points on the objects (cars or pedestrians) and the points on the backgrounds (buildings etc.) may have quite large inter-distances, their relative geometric information contains latent information regarding the objects' boundaries. So we should add the relative geometric information into the feature aggregation step to utilize the geometric information.

(3) The last gap is that the far away small objects in 3D scenes are difficult to detect. These points are few in amount and occupy small space in the whole scene [14]. They are hard to be retained after uniform downsampling [15]. As a result, during the down sampling process, we should pay more attention to retain the points on the small objects.

To tackle the above gaps, we proposes a new lightweight and effective method named *Focused Set Abstraction (FocusSA)*. The goal is to concentrate more on the contextual foreground and boundary points in down-sampling and feature aggregation steps in the SA layer. This can help to improve the accuracy of object detection, particularly for the small objects. We in particular introduce two new modules. The first is *Focused Furthest Point Sampling (FocFPS)*, which exploits efficient methods to evaluate the foreground scores and boundary scores for points, and reweighs the FPS to concentrate more on the valuable foreground and boundary points. The second module is *Geometry-aware Feature Extraction (GeoFE)*, which adds relative geometric information into the feature aggregation step. It enriches the awareness to the geometric structure in aggregating the neighborhood features. These two modules are easy-to-use, which can be flexibly plugged into the point-based object detection frameworks. To balance between the widest coverage purpose, we selectively combine the traditional SA and FocusSA in the cascaded SA layers in both the single-stage and two-stage object detection networks. The popular KITTI dataset is used to assess our methods. Experiments demonstrate that our technique greatly outperforms state-of-the-art methods in terms of detection accuracy. The key contributions of this work are as follows:

- A lightweight and effective Focused Furthest Point Sampling (FocFPS) method is proposed to refrain from including too many possibly irrelevant points and to concentrate more on the boundary and foreground ones in the SA stage.
- A novel feature extraction module called GeoFE is proposed. It can incorporate geometric relationships into original point features for improving shape awareness and robustness.
- The proposed modules are lightweight to be integrated into various point-based detection models.
- Experimental results show that our proposed method significantly outperforms other methods on KITTI official dataset, and generates competitive results in engineering applications.

2. Related works

Our method is motivated by recent advances in LiDAR-based 3D object detection. When utilizing LiDAR data, there are primarily two streams.

Point-based 3D Object Detection. The first category is point-based object detection method. This type of methods takes raw point cloud as input and conducts down-sampling and feature aggregation on the point cloud. Pointnet [11] firstly invents the point based network, which gradually downsamples points and generates predictions from the kept points. PointNet++ [16] uses grouping operations (SA & FP layers) to extend the PointNet to retrieve features at multiple levels. To save memory and to cut computing costs, PointRCNN [15] directly segments 3D point clouds and manufactures high-quality 3D boxes. They fuse semantic and local spatial features together. Inspired by Hough voting, VoteNet [17] immediately votes for virtual object pointers and aggregates vote characteristics to build a set of high-quality proposals.

In these point-based methods, down-sampling and feature extraction from raw points are generally applied in the SA layers. Point-based detectors widely adopt the Furthest Point Sampling (FPS) technique [11], where the furthest points are consecutively chosen from the initial point set. In traditional FPS, all points are treated equally without differentiating the points' importance. The widest coverage is the only criterion in point down-sampling. As a result, the important feature points maybe missed. How to incorporate the point context information into the down-sampling scheme is recently noted. 3DSSD [18] introduces a fusion sampling strategy that applies both spatial distance and semantic feature distance as the sampling criteria in FPS to pursue informative sampling with good diversity. Recent work SASA [19] proposes to assign higher weights to the foreground points to improve the attention to the potentially contextual valuable points. *But in these existing works, the contexts about the most valuable boundary points and the geometrical relations among the boundary points and the neighbors are not well captured. This paper presents FocusSA to capture these information in the down-sampling and feature aggregation steps, while keeping efficiency of these steps.*

Voxel-based 3D Object Detection. The second category is Voxel-based 3D object detection. This type of algorithms firstly rasterize point clouds into discrete grid representations (voxels & pillars), so that convolutional neural networks (CNN) can be applied. A pioneering effort called Voxelnet [20] suggests to use 3D CNN and voxelization to represent 3D scenes. SECOND [21] applies sparse convolution layers for parsing the compact representation compared to the Voxelnet. The voxel size in pillars is unique in that it is limitless in the vertical direction. Pointpillars [22] is a seminal work in which pseudo-images are used as the representation following voxelization. A number of following works [23–25] have employed the similar encoding method. *However, the perception field of voxel-based methods is not as flexible as the point-based method. We use the geometric relationship between the points to provide shape awareness for point-based methods.*

3D Object Detection with Multi-model Sensors. There are numerous techniques that combine data from several sensors to detect objects. To produce 3D rotated boxes, these approaches combine feature proposals from maps acquired from several perspectives (BEV, FV, and image). MV3D [9] uses the BEV map to create a collection of extremely accurate 3D candidate boxes that are then projected onto the feature maps. AVOD [26] enhances MV3D by using image features during the proposal generation step. MMF [27] presents a multi-task (ground and depth estimation,2D object detection) multi-sensor (LiDAR, camera) 3D object detection network for end-to-end training. However, many multi-modal techniques merely add camera features to raw lidar point clouds before feeding them to 3D detection models that are already in place. DeepFusion [28] proposes "inverseAug", which provides precise geometric alignment between lidar points and picture pixels by inverting geometric-related augmentations, and cross-attention is used by "LearnableAlign" to dynamically record the relationships between lidar

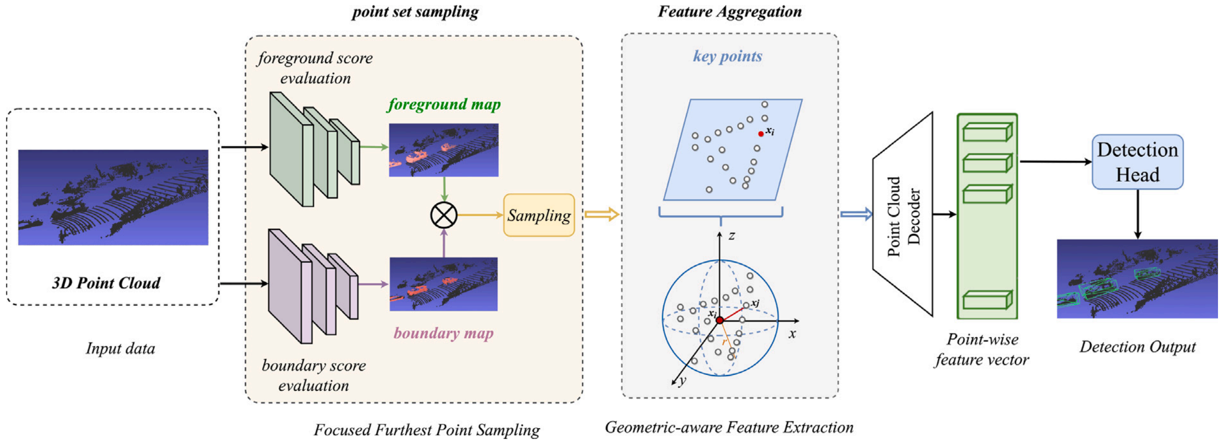


Fig. 1. Overall architecture of Focused Set Abstraction method (FocusSA). We add two segmentation modules to evaluate the point context score and use Focused Furthest Point Sampling (FocFPS) to update the point sampling method. Then, we add Geometry-aware Feature Extraction module (GeoFE) to encode the geometric features in feature aggregation.

and image features. A number of following works [29,30] have used a similar technique. *The point-based method studied in this paper can be incorporated with these methods to further enhance the final performances on 3D object detection.*

Geometric Features Extraction. Following CNNs' ground-breaking performance, there has been a lot of interest in extending CNN to handle geometric data. [31] improves the model's ability to detect objects by using a multi-scale neural network, because the multi-scale features can provide geometric information between different sections of 2D image. In contrast to 2D images, point clouds are geometric data, but they are sparse, and the objects captured by the point clouds generally have incomplete shapes. The lack of implicit grids necessitates new building blocks to accommodate the point cloud structure. A few studies have concentrated on the geometric learning in 3D point clouds. KCNet [32] represents the geometric pattern using a learnable point-set kernel. DGCNN [33] uses point relationships learned in a high-dimensional feature space to capture comparable local forms. ShapeNet [34] offers the normal vector and xyz for each point along with the coordinates. In [35], a geometric deep neural network incorporates a differentiable functional map layer that enables inherent structured prediction of correspondence between nonrigid forms. RS-CNN [36] proposes a relation-shape convolutional neural network, which can learn from the geometric topological constraint among points. *These existing works exploit geometric information mainly in the CNN backbone. The proposed GeoFE module instead adds relative geometric information in the PointNet feature aggregation step, which is more suitable to process discrete feature points after downsampling.*

3. Approach

Our work focuses on the fundamental feature extraction modules in the SA layer. We firstly evaluate the significance of the points. The points that are useful to object detection are more likely to be retained. We then employ geometric relations to aggregate local and global features. These operations are embedded in SA layer without the need for complicated network construction. The overall structure of our proposed method is shown in Fig. 1. The feature extraction component is composed by a series of Focused Set Abstraction (FocusSA) layers. Based on the original SA layer design, we embed two new modules, i.e., Focused Furthest Point Sampling (FocFPS) and Geometry-aware Feature Extraction (GeoFE). FocFPS updates the point sampling method by mapping input point features to two binary segmentation masks. GeoFE encodes the geometric features in the feature aggregation process. Then, we use the Point Cloud Decoder to generate the point-wise feature vector. Finally, the object detection results are output through the detection head.

3.1. Focused furthest point sampling

Each FocusSA layer is composed by a FocFPS component and a GeoFE component. In each FocFPS component, we embed two point segmentation modules, which can evaluate *the boundary score* b_i and *the foreground scores* o_i for each point, Then both scores are applied to reweigh the Furthest Point Sampling to make the FocFPS be aware of the object foreground and boundary features.

3.1.1. Point context score evaluation

We use a light-weight supervised method to predict the foreground segmentation scores. This method is similar to [37,38]. However, it differs in some aspects. For example, [37,38] focus on 2D image, training a binary classifier (head/background) using training strategy with annotated heads. Our method focuses on 3D point cloud, which maps the input point features to the binary segmentation masks (foreground/background). Whether a point is in the ground-truth 3D bounding boxes or not is used as the supervision label to train a segmentation model. This end-to-end architecture for supervised learning does not require separate training.

In order to predict the segmentation annotation o_i for each point, we use several Multi-Layer Perceptions (MLPs) for process the input point cloud. As shown in upper part of Fig. 2, we define $f_1^{(S_k)}, f_2^{(S_k)}, \dots, f_{N_k}^{(S_k)}$ as the S_k-dimension point features, which can calculate the foreground score $o_i \in [0,1]$ through a simple MLPs:

$$o_i = \sigma\left[\mathcal{M}\left(f_i{}^{(S_k)}\right)\right], \tag{1}$$

Where $\mathcal{M}$ denotes the MLP layers within the kth FocusSA layer, which can mapp input features f_i to foreground scores o_i. $\sigma(\cdot)$ is the sigmoid function. N_k is the total number of input points.

After evaluating the point-wise foreground scores o_i, we evaluate the boundary scores for the points. As shown in Fig. 2, whether a key point p_i belongs to a boundary point can be determined by the relationship between point p_i and its neighbors. Different categories have different characteristics, such as color, texture, shape, or reflection intensity, especially at the boundaries. When the feature difference of a key point and its neighbors is notable, this point is more likely to be at the junction of different categories. We hope to preserve these valuable points. Therefore, we design a deep network to predict the boundary score $\hat{b}_i \in [0,1]$ for the input point cloud. Specifically, based on the semantic labels, the boundary labels are produced on the fly.

As shown in the group point of Fig. 2, in the training samples, according to the labels of each point, we define the boundary label by following method. Given a certain number of points that are neighbors of p_i, if there are more than a specified proportion of points that fall into various categories, then p_i is labeled a "boundary point", otherwise it is

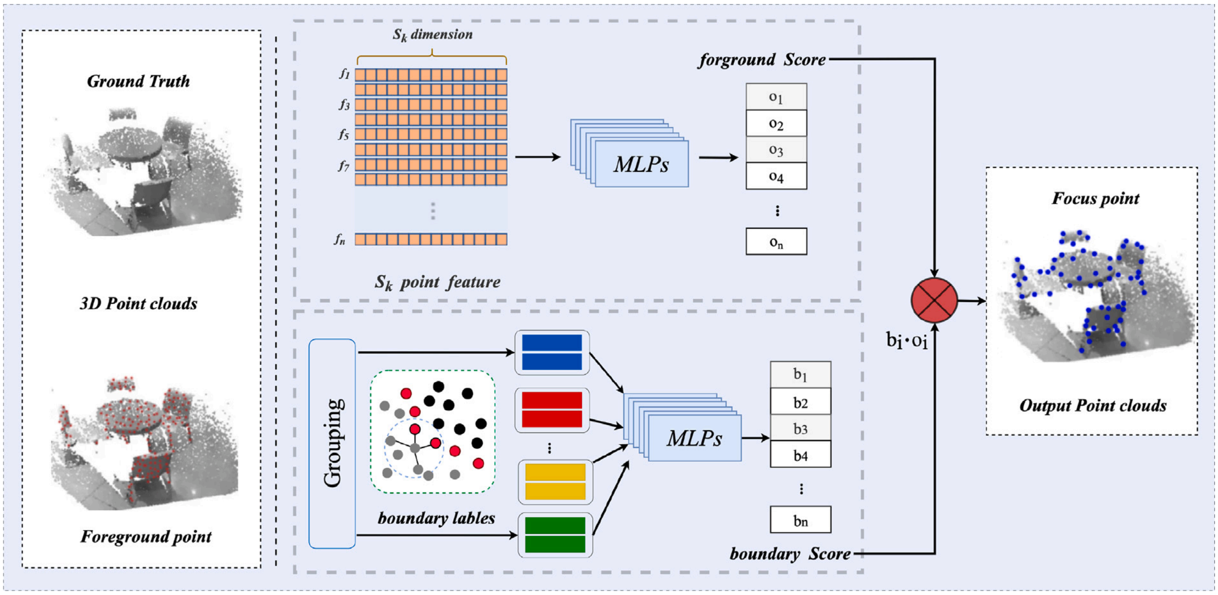


Fig. 2. Diagram of Focused Furthest Point Sampling (FocFPS). Gray and black points are belong to different categories. The red points are the transition area between two objects.

labeled "not a boundary point". Then, the boundary score is generated as following. Firstly, we utilize ball query method to find neighbor points that are within a radius. We gather features for the immediate area and use the difference in features between p_i and its neighbors $p_j \in \mathcal{N}(i)$ as input. In order to predict the boundary score $\hat{b}_i$ for p_i, we use a number of shared MLPs. For point $\hat{b}_i$, the boundary score evaluation may be written as follows:

$$\hat{b}_i = \sigma \left\{ \mathcal{M} \left[\mathcal{V} \left(f_{p_i}^{S_k}, f_{p_j}^{S_k} \right) \right] \right\} \ , \forall p_j \in \mathcal{N}\left(p_i\right), \tag{2}$$

where p_j is the neighborhood of p_i. $\mathcal{M}$ is the boundary detection network in the kth SA layer. $\mathcal{V}$ denotes the variance of the collected features. $\sigma(\cdot)$ is the sigmoid function.

3.1.2. Focused FPS by utilizing context scores

In order to preserve the positively scored points and erase those useless negative points, we then introduce a new point sampling algorithm, called Focused Furthest Point Sampling (FocFPS). The fundamental concept is to prioritize foreground and border points by a reweighing scheme using their early predicted foreground and boundary scores.

Let n be the total number of points, let $\mathbf{K}$ be the set of already selected points where $|\mathbf{K}| = k$; $\mathbf{K}$ is initialized by selecting the point with the largest $x-$coordinates. $\mathbf{U}$ denotes the remaining points which are not sampled yet, $|\mathbf{U}| = n - k$. For each point $p_i \in \mathbf{U}$, a distance array $\vec{\mathbf{d}_i} = \left\{d_i^1, d_i^2, \ldots, d_i^k\right\}$ that maintains the distances from p_i to already selected points is calculated. For points in $\mathbf{U}$, their 3D coordinates are $\left\{x_1, x_2, \ldots, x_{n-k}\right\}$; boundary scores and foreground scores are provided by two lightweight point segmentation modules, which are $\left\{b_1, b_2, \ldots, b_{n-k}\right\}$ and $\left\{o_1, o_2, \ldots, o_{n-k}\right\}$ respectively. Then for $p_i \in \mathbf{U}$, its distance vector $\vec{\mathbf{d}_i}$ is weighted by the α power of the product of the boundary and foreground scores. This makes the weighted distance $\vec{\mathbf{d}_i}'$ be context aware.

$$\vec{\mathbf{d}_i}' = (o_i \cdot b_i)^{\alpha} \cdot \vec{\mathbf{d}_i} \tag{3}$$

where α is a factor that controls the importance of the semantic and boundary information. FocFPS then selects in $\mathbf{U}$ the one with the largest weighted distance as the next sampled point, i.e., $s = \mathop{arg\max}\limits_{i} \left(\left\{ \vec{\mathbf{d}_i}', i \in \mathbf{U} \right\} \right)$. Then $\mathbf{K} = \mathbf{K} \cup s$ and $\mathbf{U} = \mathbf{U} \setminus s$, until enough points are sampled.

Algorithm 1 Algorithm of Focused Furthest Point Sampling. N is the total number of input points and K will be the set of selected points, M is the number of output points sampled by the algorithm. v_i is a label indicating whether the point i has been sampled.

```
Input: coordinates $X = \left\{x_1, x_2, ..., x_{n-k}\right\}$, distance array: $\vec{\mathbf{d}_i} = \left\{d_i^1, d_i^2, ..., d_i^k\right\}$,
    boundary scores: $B = \left\{b_1, b_2, ..., b_{n-k}\right\}$, foreground scores:$O = \left\{o_1, o_2, ..., o_{n-k}\right\}$
Output: sampled key point set $K = \left\{k_1, k_2, ..., k_m\right\}$
initialize an empty sampling point set $K$
for $i = 1 \rightarrow M$ do
    if $i = 1$ then
        $k_i = argmax(X)$
    else
        $\vec{\mathbf{d}_i}' = \{(o_i \cdot b_i)^{\alpha} \cdot \vec{\mathbf{d}_i} | v_i = 0\}$
        $k_i = argmax(\vec{\mathbf{d}_i}')$
    add $k_i$ to K,$v_i = 1$
    for $j = 1 \rightarrow N$ do
        $d_j = min(d_j, |x_j - x_{k_i}|)$
return $K$
```

3.2. Geometry-aware feature extraction (GeoFE)

The point locations generally change sharply at the boundaries of the objects. For example, for the cars or pedestrians, the points on the background are generally far from the boundary points on the objects. These relative location differences are important features to identify the objects. Therefore, how to fuse these information into the feature aggregation step is a meaningful problem. Despite the fact that CNN is a promising approach for representing contextual shapes, its convolution process has not encoded these geometric information on the 3D point clouds.

Given a point cloud with N points $P = \left\{p_i | i = 1, \ldots, N\right\} \in \mathbb{R}^{N \times 3}$, each point contains 3D coordinates $p_i = (x_i, y_i, z_i)$ and also includes additional coordinates representing color, surface normal, and so on. The input feature map F can be denoted as $F = \left\{f_i | i = 1, \ldots, N\right\} \in \mathbb{R}^{N \times C_{in}}$, and the output feature map G of P can be denoted as $G =$

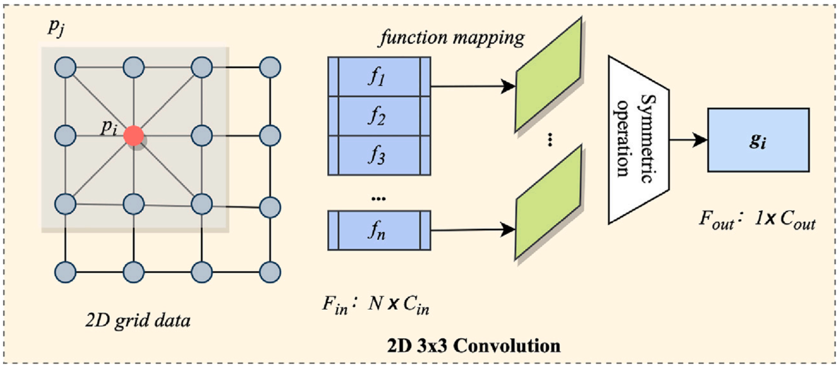


Fig. 3. Diagram of 2D grid convolution where symmetric operation is the aggregation function like Max, SUM or AVG.

$\{g_i | i = 1, \ldots, N\} \in \mathbb{R}^{N \times C_{out}}$. To this end, we formulate a general convolutional operation g_i for each point p_i as:

$$g_i = \sigma(\mathcal{A}(\{\mathcal{K}(p_i, p_j) f_j, \forall p_j\})), d_{ij} < r, \forall p_j \in \mathcal{N}_i \quad (4)$$

where r is the radius of the sphere and d_{ij} is the Euclidean distance between p_i and p_j. $\mathcal{N}$ denotes the neighborhood points p_j, $\mathcal{A}$ is the aggregation function like Max, SUM or AVG, $\mathcal{K}(p_i, p_j)$ is a function that returns convolutional weights based on the position relation. σ is a nonlinear activator.

Limitations of general convolution. In the above definition, the point convolution may be thought as a specific example of 2D convolution. In classic CNN, the neighborhood (convolution kernel) of 3 × 3 kernel lies in a rectangular patch centered on pixel i. $\mathcal{K}$ is not shared over each point in $\mathcal{N}_i$, causing the incapability to process irregular and unordered point set P. Furthermore, it always implies a definite positional relationship in the regular grid between p_i and its neighbor p_j. As shown in Fig. 3, when $p_1, p_2, \ldots, p_n$ represent image pixels on a regular grid, $\mathcal{K}$ is actually constrained to encode fixed position (left, right, top, down) in the learning process. To overcome this shortage of CNN, we provide a new geometry-aware feature extraction (GeoFE) method. This extension is methodology simple but effective.

Feature extraction. The objective is to generate an inductive representation f_P within the neighborhood centered on p_i, which should contain geometry information in addition to the original feature. The blue points shown in Fig. 4 are the key points obtained by FocFPS in the previous stage. When extracting features, we first get the neighbor p_j of point p_i. Because in the neighborhood of 3D space, the coordinates of the points may be used to acquire the global shape structure, and the coordinate difference can be used to gain the local neighborhood information. Therefore, we integrate the coordinates and coordinate differences into a relation vector h and add it to feature extraction, as shown in the purple part. In detail, we replace $\mathcal{K}(p_i, p_j)$ with a learning mapping $\mathcal{M}$ of a relation vector h, the preset geometric priors between p_i and p_j, with the purpose of $\mathcal{M}$ being to abstract relation expression between two points, which may indicate their spatial arrangement. In this way, we get the feature F'_{in} that contains the geometry information.

Fusion manners. As shown in Fig. 4, in addition to geometrically encoding the features, we keep the original feature extraction unchanged, and directly concatenate the original features F_{in} to the geometric features F'_{in}. Despite various fusion techniques like cross-attention or summation, are widely used for efficiency [39], we choose the most concise concatenation. Then the final representation F'_{out} is obtained through following calculations.

$$g_i = \sigma(\mathcal{A}(\{\mathcal{M}(h(p_i - p_j, p_i, p_j)) f_j, \forall p_j\})), d_{ij} < r, \forall p_j \in \mathcal{N}_i \quad (5)$$

This extraction model is simple, including two Conv2D layers, two BatchNorm2d layers and a relu layer. Spatial arrangement may be encoded using $\mathcal{M}$. Because of its excellent mapping capabilities, we use a shared multi-layer perceptron (MLP) in our implementation. The aggregation function $\mathcal{A}$ is symmetric function max pooling, ReLU [40] is used as nonlinear activator σ. This elegantly converts $\mathcal{K}(p_i, p_j)$ to $\mathcal{M}$, whose feature is important to both p_i and p_j.

3.3. Loss function

For model optimization, the total segmentation loss is divided into two parts, foreground segmentation loss $\mathcal{L}_{seg}$ and boundary segmentation loss $\mathcal{L}_{bry}$. The foreground segmentation loss $\mathcal{L}_{seg}$ is computed with a cross entropy (CE):

$$\mathcal{L}_{seg} = -\sum_{k=1}^{m} \sum_{i=1}^{N_k} CE(o_i^k, \hat{o}_i^k) \quad (6)$$

where o_i^k denotes the predicted foreground score, and $\hat{o}_i^k$ denotes the ground-truth label of the ith point in the kth SA layer. Similar to foreground segmentation loss $\mathcal{L}_{seg}$, we calculate the boundary segmentation loss $\mathcal{L}_{bry}$ by a CE loss function:

$$\mathcal{L}_{bry} = -\sum_{k=1}^{m} \sum_{i=1}^{N_k} \left(w_1 \cdot b_i^k \log \hat{b}_i^k + w_2 \cdot (1 - b_i^k) \log(1 - \hat{b}_i^k)\right) \quad (7)$$

Where b_i^k and $\hat{b}_i^k$ denote the ground-truth boundary label and the predicted boundary score of the ith point in the kth SA layer. N_k is the total number of input points, w_1 and w_2 are used to balance the difference between the numbers of the two categories. The overall segmentation loss is $\mathcal{L}_{seg} + \mathcal{L}_{bry}$.

3.4. Implementation details

In this section, we describe how to implement our FocusSA in one-stage point-based model 3DSSD and two-stage point-based model PointRCNN.

FocusSA+3DSSD. Firstly during training phase, we automatically label the boundary points for each input point cloud. Specially, the points with more than 60% of the 64 neighbor points who do not belonging to the same category are considered to be boundary points. Then we evaluate the variance of the color features of the 64 neighbors for each point based on the neighborhood information.

To capture a more adequate geometric relationship, the key points that are picked from FocFPS are used to perform GeoFE. In each neighborhood, a certain number of neighbors are chosen at random for batch processing, and they are normalized to utilize the centroid as the origin. We defined a vector, i.e., p_i, p_j, $p_i - p_j$ and Euclidean distance as the relationship vector between points. Then we use a three layer shared MLP that can adapt arbitrary continuous mappings. Batch normalization [41] is applied in each MLP.

3DSSD provides two distinct point sampling algorithms, and each method samples half of the total key points. To better retain more important points from the foreground and the boundary, we leave the number of sampled key points unchanged and replace F-FPS parts with our proposed FocFPS. The FocFPS key points are utilized as candidate points to produce equivalent voting points. As shown in Fig. 5, we start implementing our FocFPS from the second SA layer, because the first level's raw point input cannot create meaningful semantics.

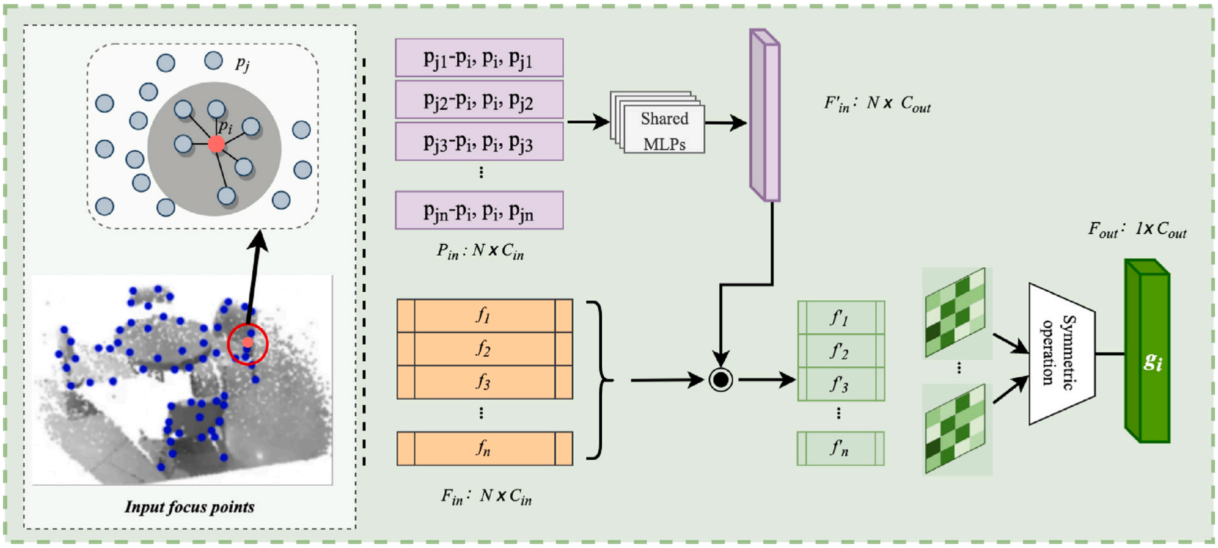


Fig. 4. Illustrates Geometry-aware Feature Extraction (GeoFE). The blue points represent the key points after FocFPS sampling, red point is the points that need to extract features, and the gray points are its neighbors. We use the relationship between points to re-extract features and fuse them with the original features with boundary weights.

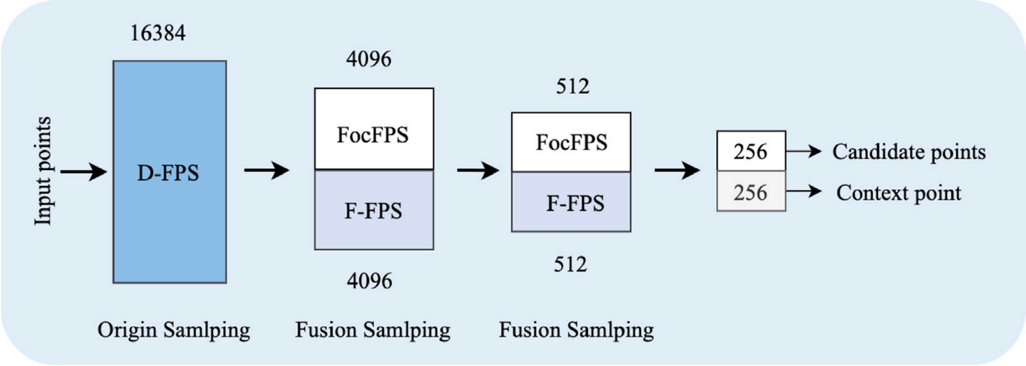


Fig. 5. Illustration of 3DSSD backbones with Focused Furthest Point Sampling. Through the use of three FocusSA layers and a fusion sampling approach, it creates global features for all representative points using the input of the raw point cloud (x, y, z, r). We sample 16,384, 4096, and 512 points using F-FPS and FocFPS, respectively, and then combine the two sets for the grouping process in a FocusSA layer.

FocusSA + PointRCNN. PointRCNN samples important points using standard FPS, as shown in Fig. 6. Followed by the original data augmentation strategies, we apply FocFPS in the level 2 and level 4, retaining the initial implementation's basic framework in place, including FP layers. During training phase, we use the same strategy like implement in 3DSSD to annotate the target boundary points and capture more sufficient geometric relation. The segmentation loss weights are set at 0.001, 0.01 and 0.1 for the three levels.

4. Experimental setup

This section demonstrates the performance of our method and compares it to other state-of-art methods on KITTI dataset. After that, extensive ablation investigations are carried out. To highlight significant changes from earlier work, we also show visualization results.

4.1. KITTI dataset

KITTI dataset [42] provides three categories in 3D object detection task, which is a prevalent benchmark, namely, Car, Pedestrian, and Cyclist. It includes 7841 training images/LiDAR point clouds and 7518 test samples. All training examples are separated into two categories: train groups (3712 samples) and val groups (3769 samples), the train split is used to train all experimental models, and the val split is used to evaluate them. We assess on the Car classes and apply the KITTI official evaluation procedure for submission to the KITTI test server. We utilize the average precision (AP) measure to compare different approaches to a set of state-of-the-art methods.

4.2. Experiment settings

According to the different point cloud networks mentioned in Section 3.4, we integrate FocusSA on it, and assess them on KITTI object detection task. We use the OpenPCDet toolbox to build our experimental models.

We train our model with ADAM optimizer for 80 epochs. The batch size is 8 on four NVIDIA 2080Ti GPU cards and momentum for BN starts with 0.9. We apply one-cycle learning rate schedule with the peak learning rate at 0.01 and decays with a rate of 0.5 every 20 epochs. Further, we set balance parameter α=1 in FocFPS. To avoid over-fitting, we employ a variety of data augmentation methods, such as each point cloud is randomly flipped along x-axis, add a random translation $(\triangle x, \triangle y, \triangle z)$; and random rotation following a uniform distribution

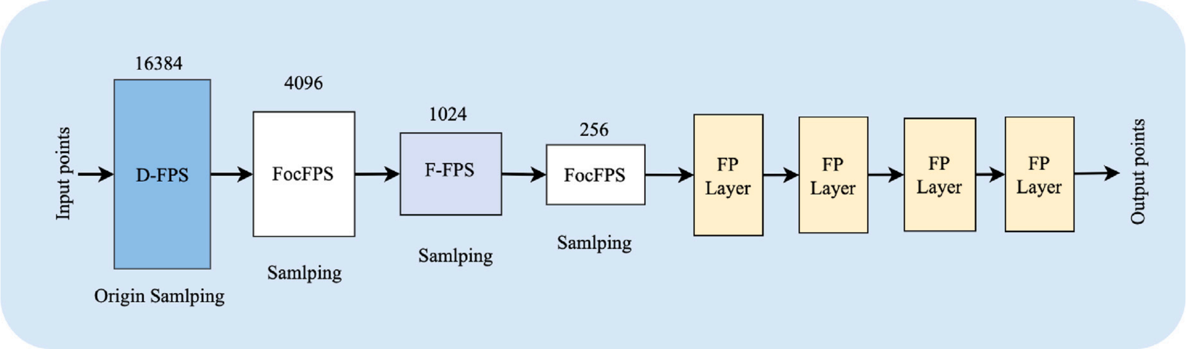


Fig. 6. Illustration of PointRCNN backbones with Focused Furthest Point Sampling. Through the use of four FocusSA layers and a fusion sampling approach, it creates global features for all representative points using the input of the raw point cloud (x, y, z, r). With F-FPS and FocFPS, we sample 16,384, 4096, 1024, and 256 points to pass to the subsequent grouping procedure.

Table 1
Comparison with the state-of-the-art methods on the KITTI test set for Car 3D detection. The evaluation metric is the AP calculated on 40 recall points. FocusSA obviously improves two baselines and surpasses other state-of-the-art methods.

Method	Modality	AP_{3D} (%)				AP_{BEV} (%)			
		Mean	*Mod.*	Easy	Hard	Mean	*Mod.*	Easy	Hard
VoxelNet [17]	LiDAR	66.5	64.17	77.82	57.51	82.00	79.26	89.35	77.39
SECOND [21]	LiDAR	74.33	75.96	84.65	68.71	81.80	79.37	88.07	77.95
PointPillars [22]	LiDAR	74.11	74.31	82.58	68.99	84.76	86.10	88.35	79.83
ContFuse [43]	LiDAR+RGB	71.38	68.78	83.68	61.67	85.10	85.35	**94.07**	75.88
PointPainting [44]	LiDAR	73.67	71.70	82.11	67.08	87.97	88.11	92.45	83.36
MV3D [9]	LiDAR+RGB	64.20	63.63	74.97	54.00	78.45	78.93	86.62	69.80
F-PointNet [45]	LiDAR+RGB	70.86	69.79	82.19	60.59	83.54	84.67	91.17	74.77
AVOD [26]	LiDAR+RGB	73.52	66.47	76.39	60.23	85.14	84.82	90.99	79.62
PI-RCNN [46]	LiDAR+RGB	76.41	74.82	84.37	70.03	86.08	85.81	91.44	81.00
PointRCNN [15]	LiDAR	76.67	75.64	86.96	70.70	87.41	87.39	92.13	82.72
F-ConvNet [47]	LiDAR+RGB	76.81	76.39	87.36	66.69	82.44	83.08	89.69	74.56
SAT-GCN [48]	LiDAR	79.46	78.12	86.55	73.72	88.13	88.06	92.83	83.51
SMS-Net [49]	LiDAR	77.89	76.21	87.01	70.45	–	–	–	–
Semi-super [50]	LiDAR	79.36	76.28	86.74	75.07	–	–	–	–
RE-Det3D [51]	LiDAR	78.19	78.19	–	–	88.07	88.07	–	–
FocusSA+PointRCNN (Ours)	LiDAR	80.54	79.43	**87.91**	**74.29**	88.40	88.76	92.72	83.71
FocusSA+3DSSD (Ours)	LiDAR	**80.73**	**80.65**	87.41	74.15	**89.00**	**88.88**	92.49	**85.65**

$[-\pi/4, +\pi/4]$. We also employ 3D non-maximum suppression (NMS) with a 0.01 threshold during the inference phase to eliminate duplicate predictions.

5. Results and discussion

This section demonstrates the performances of our method. We first discuss the main results on official KITTI test set and compare it to other state-of-art methods. To highlight significant changes from earlier work, we also show visualize experimental results. After that, extensive ablation investigations and compatibility analysis are carried out.

5.1. Main results

We implement FocusSA in pointRCNN and 3DSSD networks for 3D object detection. The result networks are called FocusSA+pointRCNN and FocusSA+3DSSD respectively. We compare their object detection performances with other state-of-the-art models. We ran studies on a regularly used automotive category and compared the outcomes using average precision (AP) with an 0.7 IoU threshold. The dataset is divided into three complexity categories, i.e., easy, moderate, and hard, depending on item size, occlusion, and truncation.

In Table 1, we compare our methods, i.e., FocusSA+pointRCNN and FocusSA+3DSSD with other 3D detectors on KITTI test set, The results evaluated using 3D and BEV APs at a 0.7 IoU threshold. It can be seen that our proposed networks not only outperform their baseline, but also outperform almost all state-of-the-art methods. Our technique beats 3DSSD and PointRCNN on the primary measure, AP on "moderate" cases, by 1.08% and 3.79%, respectively. Our method also outperforms PointRCNN on "hard" instances by 3.59%. Furthermore, the bird's-eye-view (BEV) APs also outperforms the "moderate" cases in the class Car. It takes around 10.1 fps on the KITTI dataset. Our strategy delivers considerable gains at the moderate and difficult levels, which can preserve enough important foreground & boundary points and extract geometric information for better detecting objects. The results indicate its important implications for utilizing the point features in the FocusSA layers. Furthermore, the total number of parameters utilized to train the FocusSA models is 2.85M. This demonstrates its enormous potential for real-time applications such as scene parsing in autonomous driving.

It should be noted that at a high computational cost, the current SOTA approaches extract point cloud features using a transformer-based backbone with several parameters. FocusSA portable, lightweight applicable to point based methods but incompatible with Transformer based SOTA techniques. We did not compare some SOTA algorithms for this reasons.

5.2. Visualization

The sampled important points by the latest SA layers of FocusSA on the KITTI dataset are seen in Fig. 7. The red dots in the picture indicate the 256 key points sampled by the final SA layer, while the white ones represent the background points. FocusSA can keep more

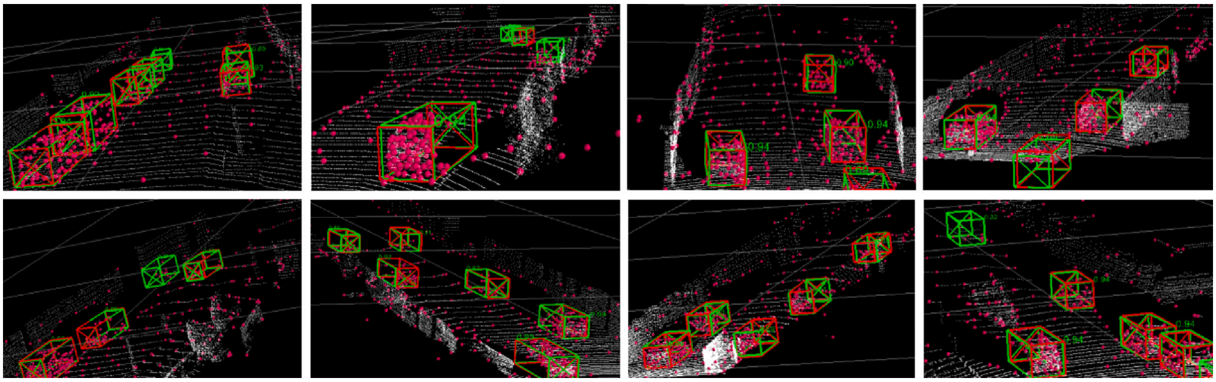


Fig. 7. Visualizing detection results on KITTI val split. The labels for the predictions and the ground truth are colored in red and green respectively. Pink dots denotes the 512 key points sampled in final last FocusSA layer.

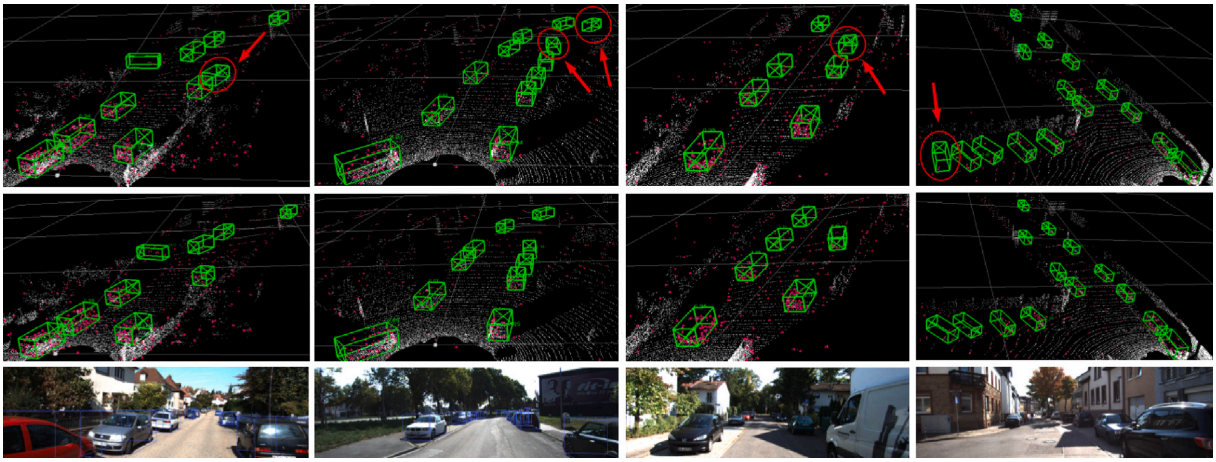


Fig. 8. Visualizing results of FocusSA (top) and 3DSSD (middle) on KITTI validation set. The predicted bounding boxes are shown in green. Bottom line is a 2D color map for easy observation.

points on foreground and boundary for challenging cases, even for heavily obstructed or small objects. Fig. 9 depicts the sampling results of various existing methods. It is evident that our approach retains more key points. As a result, our suggested FocFPS sampling technique is more likely to detect objects.

The outcomes of FocusSA on KITTI are shown in Fig. 8 (top). The expected bounding boxes are colored green, and the whole point cloud is colored white. The things that our technique successfully detects but that are not recognized by 3DSSD are shown by red arrows and circles. In the bottom row, we also display the images gathered from the 2D scene for ease of viewing and comparison. The graphic makes it very evident that our technique is more accurate in detecting far-off objects. Additionally, it is possible to identify distant objects that are obscured.

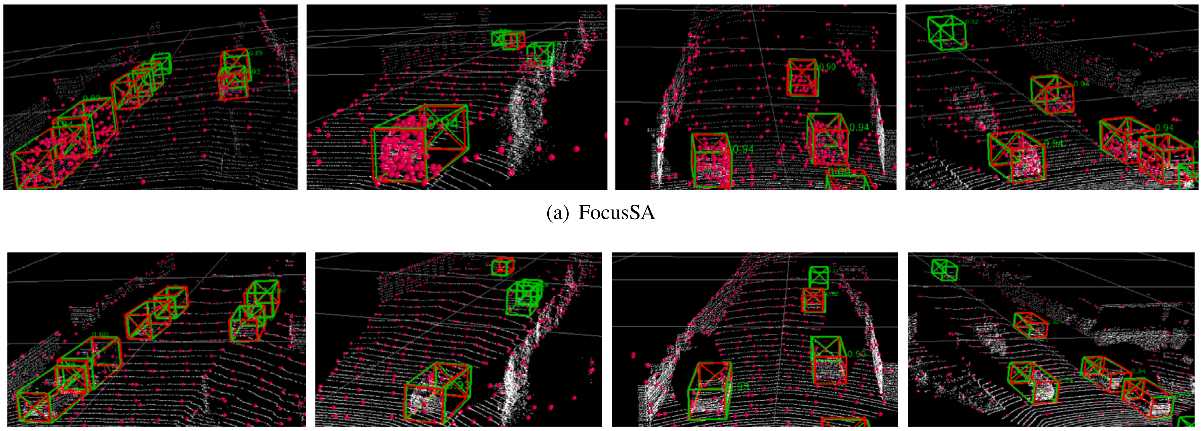
5.3. Ablation study

In order to prove the effectiveness of our method that combines the semantic and shape relation information of the objects proposed in this paper, this section conducts more specific experiments. All ablation investigations are performed on the KITTI dataset [42].

Effects of Focused Furthest Point Sampling. As shown in Table 2, the baseline object detection network is 3DSSD [18] which uses F-FPS. We compare the performances when using and not using FocFPS or GeoFE in the SA layer. The APs of our method are all greater than those without these operations under various assignment schemes. In Table 2, the 1st row displays the original model's findings without any of our approaches. The 4th row depicts the result of employing FocFPS. This sampling technique provides a substantially higher mAP, 1.14% better than the one using F-FPS [18] alone. This is because our method is easier to recognize small objects in challenging cases. The 2nd and 3rd rows illustrate the results of employing semantic information and boundary information as key point selection criteria, respectively. It illustrates that both of them can benefit the classification results.

Effects of Geometry-aware Feature Extraction. The last row of Table 2 demonstrates that using GeoFE, our method has been enhanced by 0.01%, 2.67% and 0.83% accordingly on the basis of FocFPS. This demonstrates how adding object geometry relations into feature extraction improves object detection accuracy. In addition, the last row is not the highest on the "easy" cases, the reason is that, we utilize KITTI dataset in the experiment, which feeds all data into the network simultaneously. Hence, we output the network results together instead of training each case separately. When evaluating the model's performance, the "easy", "moderate", and "hard" cases must be all taken into account. In fact, it is not always the case that "easy", "moderate", and "hard" will increase simultaneously. Throughout the training process, the model will be updated based on data sets of varying difficulty to improve robustness and generalization ability.

Effects of Balance Factor. We replace the F-FPS section with suggested FocFPS, at the same time, we keep other sample parameters constant, as illustrated in Fig. 5. We also compare FocFPS with various balancing factor values $alpha$ from the third to sixth row of Table 3. It demonstrates that whether FocusSA is effective or not is largely limited by the sampling points. A big or small $alpha$ could not adequately increase detection accuracy. When α is close to 0, FocFPS will degrade to vanilla FPS, when α becomes extremely large, key points may be crowded into a few easily identifiable instances while failing



(a) FocusSA

(b) 3DSSD (c) PointRCNN (d) CIA-SSD (e) PointNet

Fig. 9. Visual comparison of the FocusSA sampling method and various SOTA sampling methods. FocusSA can get more significant foreground and boundary points.

Table 2
Ablation study on the 3DSSD baseline modules, we provide the Car AP on KITTI val split. Here, "FocFPS." and "GeoFE" denote the Focused Furthest Point Sampling Model and Geometry-aware Feature Extraction respectively. *S., and *B. denote using semantic information and boundary information as the criterion to choose key points respectively.

FocFPS (*S.)	FocFPS (*B.)	FocFPS	GeoFE	Easy	Mod.	Hard	mAP
×	×	×	×	91.57	82.24	80.45	84.75
✓	×	×	×	92.36	83.02	80.77	85.38
×	✓	×	×	**92.92**	82.74	80.48	85.37
✓	✓	✓	×	92.32	83.04	82.31	85.89
✓	✓	✓	✓	92.33	**85.71**	**83.14**	**87.15**

Table 3
Comparison of the performance of FPS, F-FPS, and FocFPS with various balancing parameters.

Sampling method	FocFPS	Fusion sampling	Mod.	Easy	Hard
FPS	×	×	82.75	91.08	79.93
F-FPS	×	✓	83.46	91.54	82.18
FocFPS (α=0.1)	✓	✓	83.07	91.78	80.38
FocFPS (α=1)	✓	✓	**85.71**	**92.33**	**83.14**
FocFPS (α=10)	✓	✓	83.51	92.08	82.32
FocFPS (α=100)	✓	✓	83.03	91.57	80.28

Table 4
The results (Moderate AP%) of four intuitive geometric relations vector, Model A is the original model; Model B applies 3D Euclidean distance (3D-ED); Model C use the coordinates difference and 3D-ED; Model D adds the coordinates of two points.

Model	Geometric relation vector	Channels	Mod.
A	–	0	83.04
B	3D Euclidean distance	1	85.24
C	(3D Euclidean distance, $p_i - p_j$)	4	85.36
D	(3D Euclidean distance, p_i, p_j, $p_i - p_j$)	10	**85.71**

Table 5
Complexity of FocusSA in point cloud classification.

Method	Input data		#params
	LiDAR	IMG	
PCNN [52]	✓		8.20M
PartA^2 [25]	✓		6.38M
SECOND [21]	✓		5.33M
PointRCNN [15]	✓		4.40M
ImVoteNet [53]	✓	✓	4.12M
PointNet [11]	✓		3.50M
MVXNet [54]	✓	✓	3.35M
ImVoxelNet [55]	✓	✓	3.12M
Ours	✓		**2.85M**

to cover distant or occluded instances, thus failing to capture object shape information. Therefore, $\alpha = 1$ is a suitable choice, as all three difficulty levels achieve adequate performance at the same time. It is worth noting that there are green false detection boxes on the first and second subgraphs of Fig. 9. Due to oversampling, model training may prefer to anticipate positive samples while discarding negative data, resulting in an imbalance of positive and negative samples. As a result, it is necessary to adjust the sampling weight in accordance with the practical applications.

Geometric relation h. Understanding how to define "relation" is a problem worth studying because the core of GeoFE to learn from the relationships between points. Relation can be defined whatever you like as long as it can represent underlying form discriminatorily. We experiment with four intuitive connection definitions as examples in order to test this claim and make it easier for people to grasp. The findings are shown in Table 4.

It can be seen that only using 3D Euclidean distance in model B reach 85.24%, higher than model A by 2.2%. This demonstrates the effectiveness of geometric relation in feature extraction. Additionally, we introduce a new connection involving the difference between the coordinates (model C) and the coordinates themselves (model D). The findings are all far better than the original model. Experiments shows that when 3D Euclidean distance, difference between the coordinates and the coordinates themselves are included, the model D performs the best, achieving 85.71%. This further highlights the utility of the GeoFE.

5.4. Compatibility analysis

FocusSA is effective and easy-to-plug-in in point-based detection methods. It obtains notable enhancement on both one-stage model 3DSSD and two-stage model PointRCNN. In this section, we will test its compatibility.

As shown in Table 6, algorithms embedded in FocusSA improve the detection performance for all difficulty levels. In moderate mode, our model outperforms by 1.95% and 2.25% AP compared to one-stage method 3DSSD, and outperforms by 0.64% and 0.96% AP in Hard case. Furthermore, when compared to the two-stage method, PointRCNN, our method outperforms by 1.86% and 0.55% AP in the Hard case. These difficult objects inherently have few 3D points, which are difficult to preserve during sampling. Our method focuses on more important points and fuses the geometric feature, which can preserves rich information and accomplish accurate classification results. Table 5 summarizes the number of params and the modality data of FocusSA in classification. Compared with PointRCNN and PointNet, FocusSA

Table 6
Compatibility study of FocusSA in two different point-based detection method, We assess it using the KITTI val split of Car class. *FF. *FE represents FocFPS and GeoFE respectively.

Method	Easy ↑	Mod. ↑	Hard ↑	mAP ↑
3DSSD [18]	91.54	83.46	82.18	85.73
FocusSA(*FF)	92.29 (+0.75)	85.41(+1.95)	82.82 (+0.64)	86.68 (+1.11)
FocusSA(*FE)	92.33 (+0.79)	85.71(+2.25)	83.14 (+0.96)	87.15 (+1.58)
PointRCNN [15]	91.57	82.24	80.45	84.75
FocusSA(*FF)	92.32 (+0.75)	83.04 (+0.8)	82.31 (+1.86)	85.89 (+1.14)
FocusSA(*FE)	92.36 (+0.79)	82.90 (+0.66)	81.00 (+0.55)	85.43 (+0.67)

reduces the params by 35.2% and 18.6%, respectively, which demonstrates its high potential for real-time applications such as scene parsing in autonomous driving.

5.5. Summary

The extensive experiments evaluated FocusSA from different perspectives. These perspectives include official test set, data visualization, serial ablations and compatibility analysis. The key findings are summarized below.

1. Our method has a higher AP than compared methods on the official KITTI test set, particularly in the “moderate” cases, and beats PointRCNN on “hard” cases by 3.59%.
2. The addition of semantic information allows the sampling method to preserve more foreground boundary points and improves long-distance and small object detection accuracy.
3. The ablation experiment shows that using geometric information can better enhance feature extraction and detection outcomes, and the mAP is raised by 1.26%.
4. The comparison experiments demonstrate that FocusSA can be used to both the two-stage algorithm as well as the single-stage algorithm. The structure’s architecture is straightforward and efficient.

6. Conclusion

In this work, FocusSA, namely, Focused Set Abstraction has been proposed. It suggests the FocFPS and GeoFE two modules in the set abstraction layer. The FocusSA architecture is easily incorporated into existing pipelines for point-based point cloud perception. The core of FocusSA contains two components. The first is FocFPS, which incorporates semantic and boundary information to guide FPS to better sample potential objects. The second is GeoFE, which enables explicit reasoning about the spatial relationship for discriminative shape awareness. The experimental results show that FocusSA performs better in 3D object identification when measured against the KITTI benchmark’s official ranking criterion. It outperforms 3DSSD and PointRCNN by 1.08% and 3.79% on “moderate” instances in Car class. It is an easy-to-plug-in module which can enhance many 3D object detectors, especially point-based method, including 1-stage and 2-stage ones. We believe that our findings will inspire the scientific community to address the bottleneck of autonomous driving technology, increase the vehicle’s capacity for object detection in hard instances.

CRediT authorship contribution statement

Zhe Huang: Methodology, Data curation, Writing & editing. **Yongcai Wang:** Conceptualization, Methodology, Project administration, Review & editing. **Jie Wen:** Resources, Review & editing. **Peng Wang:** Visualization, Validation. **Xudong Cai:** Investigation, Review.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yongcai Wang reports statistical analysis was provided by National Key Research and Development Program of China. Yongcai Wang reports statistical analysis was provided by National Natural Science Foundation of China.

Data availability

Data will be made available on request.

References

[1] Ying Yu, Chenglin Yu, Gangyan Xu, Ray Y Zhong, George Q Huang, An operation synchronization model for distribution center in E-commerce logistics service, Adv. Eng. Inform. 43 (2020) 101014.
[2] Shan-Huen Huang, Ying-Hua Huang, Carola A Blazquez, Chia-Yi Chen, Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm, Adv. Eng. Inform. 51 (2022) 101536.
[3] Yuan Tian, Xinming Zhang, Binyu Yang, Jian Wang, Shi An, An individual-based spatio-temporal travel demand mining method and its application in improving rebalancing for free-floating bike-sharing system, Adv. Eng. Inform. 50 (2021) 101365.
[4] Rui Qian, Xin Lai, Xirong Li, 3D object detection for autonomous driving: a survey, Pattern Recognit. 130 (2022) 108796.
[5] Georgios Zamanakos, Lazaros Tsochatzidis, Angelos Amanatiadis, Ioannis Pratikakis, A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving, Comput. Graph. 99 (2021) 153–181.
[6] Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, Laurent Kneip, Devo: Depth-event camera visual odometry in challenging conditions, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 2179–2185.
[7] Shih-Wen Hsiao, Po-Hsiang Peng, Yi-Cheng Tsao, A method for the analysis of the interaction between users and objects in 3D navigational space, Adv. Eng. Inform. 50 (2021) 101364.
[8] Jun Fu, Chen Hou, Wei Zhou, Jiahua Xu, Zhibo Chen, Adaptive hypergraph convolutional network for no-reference 360-degree image quality assessment, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 961–969.
[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, Tian Xia, Multi-view 3d object detection network for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.
[10] Zhipeng Ding, Marc Niethammer, Votenet++: Registration refinement for multi-atlas segmentation, in: 2021 IEEE 18th International Symposium on Biomedical Imaging, ISBI, IEEE, 2021, pp. 275–279.
[11] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.
[12] Jingyu Gong, Jiachen Xu, Xin Tan, Jie Zhou, Yanyun Qu, Yuan Xie, Lizhuang Ma, Boundary-aware geometric encoding for semantic segmentation of point clouds, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 1424–1432.
[13] Mutian Xu, Runyu Ding, Hengshuang Zhao, Xiaojuan Qi, Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3173–3182.
[14] Jing Zhang, Dacheng Tao, Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things, IEEE Internet Things J. 8 (10) (2020) 7789–7817.
[15] Shaoshuai Shi, Xiaogang Wang, Hongsheng Li, Pointrcnn: 3d object proposal generation and detection from point cloud, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 770–779.

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, Leonidas J Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).

[17] Zhipeng Ding, Xu Han, Marc Niethammer, Votenet: A deep learning label fusion method for multi-atlas segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 202–210.

[18] Zetong Yang, Yanan Sun, Shu Liu, Jiaya Jia, 3Dssd: Point-based 3d single stage object detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11040–11048.

[19] Chen Chen, Zhe Chen, Jing Zhang, Dacheng Tao, Sasa: Semantics-augmented set abstraction for point-based 3d object detection, in: AAAI Conference on Artificial Intelligence, Vol. 1, 2022, pp. 652–660.

[20] Yin Zhou, Oncel Tuzel, Voxelnet: End-to-end learning for point cloud based 3d object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490–4499.

[21] Yan Yan, Yuxing Mao, Bo Li, Second: Sparsely embedded convolutional detection, Sensors 18 (10) (2018) 3337.

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, Oscar Beijbom, Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12697–12705.

[23] Jongyoun Noh, Sanghoon Lee, Bumsub Ham, Hvpr: Hybrid voxel-point representation for single-stage 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14605–14614.

[24] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, Lei Zhang, Structure aware single-stage 3d object detection from point cloud, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11873–11882.

[25] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, Hongsheng Li, Part-a2 net: 3d part-aware and aggregation neural network for object detection from point cloud, 2020, arXiv preprint arXiv:1907.03670, 2 (3).

[26] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, Steven L Waslander, Joint 3d proposal generation and object detection from view aggregation, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 1–8.

[27] Ming Liang, Bin Yang, Yun Chen, Rui Hu, Raquel Urtasun, Multi-task multi-sensor fusion for 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7345–7353.

[28] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al., Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17182–17191.

[29] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, Zhi Tang, BEVFusion: A simple and robust LiDAR-camera fusion framework, 2022, arXiv preprint arXiv:2205.13790.

[30] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, Chiew-Lan Tai, Transfusion: Robust lidar-camera fusion for 3d object detection with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1090–1099.

[31] Sultan Daud Khan, Saleh Basalamah, Scale and density invariant head detection deep model for crowd counting in pedestrian crowds, Vis. Comput. 37 (8) (2021) 2127–2137.

[32] Yiru Shen, Chen Feng, Yaoqing Yang, Dong Tian, Mining point cloud local structures by kernel correlation and graph pooling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4548–4557.

[33] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, Justin M Solomon, Dynamic graph cnn for learning on point clouds, Acm Trans. Graph. 38 (5) (2019) 1–12.

[34] Hailiang Ye, Zijin Du, Feilong Cao, A novel 3D shape classification algorithm: point-to-vector capsule network, Neural Comput. Appl. 33 (23) (2021) 16315–16328.

[35] Linbo Hao, Huaming Wang, Geometric feature statistics histogram for both real-valued and binary feature representations of 3D local shape, Image Vis. Comput. 117 (2022) 104339.

[36] Yongcheng Liu, Bin Fan, Shiming Xiang, Chunhong Pan, Relation-shape convolutional neural network for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904.

[37] Sultan Daud Khan, Yasir Ali, Basim Zafar, Abdulfattah Noorwali, Robust head detection in complex videos using two-stage deep convolution framework, IEEE Access 8 (2020) 98679–98692.

[38] Sultan Daud Khan, Rafi Ullah, Mussadiq Abdul Rahim, Muhammad Rashid, Zulfiqar Ali, Mohib Ullah, Habib Ullah, An efficient deep learning framework for face mask detection in complex scenes, in: Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings, Part I, Springer, 2022, pp. 159–169.

[39] Boyue Wang, Yongli Hu, Junbin Gao, Yanfeng Sun, Fujiao Ju, Baocai Yin, Adaptive fusion of heterogeneous manifolds for subspace clustering, IEEE Trans. Neural Netw. Learn. Syst. 32 (8) (2020) 3484–3497.

[40] Ingrid Daubechies, Ronald DeVore, Simon Foucart, Boris Hanin, Guergana Petrova, Nonlinear approximation and (deep) ReLU networks, Constr. Approx. 55 (1) (2022) 127–172.

[41] Lechao Cheng, Chaowei Fang, Dingwen Zhang, Guanbin Li, Gang Huang, Compound batch normalization for long-tailed image classification, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 1925–1934.

[42] Andreas Geiger, Philip Lenz, Raquel Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.

[43] Ming Liang, Bin Yang, Deep continuous fusion for multi-sensor 3d object detection, in: Proceedings of the European Conference on Computer Vision, ECCV, 2020, pp. 641–656.

[44] Sourabh Vora, Alex H. Lang, Pointpainting: Sequential fusion for 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4604–4612.

[45] Zhixin Wang, Kui Jia, Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 1742–1749.

[46] Liang Xie, Zhengxu Yu, Guodong Xu, PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 12460–12467.

[47] Zhixin Wang, Kui Jia, Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 1742–1749.

[48] Li Wang, Ziying Song, Zhang, SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving, Knowl.-Based Syst. 259 (2023) 110080.

[49] Sheng Liu, Wenhao Huang, Yifeng Cao, SMS-net: Sparse multi-scale voxel feature aggregation network for LiDAR-based 3D object detection, Neurocomputing 501 (2022) 555–565.

[50] Jiacheng Zhang, Huafeng Liu, Jianfeng Lu, A semi-supervised 3D object detection method for autonomous driving, Displays 71 (2022) 102117.

[51] Yiqiang Wu, Weiping Xiao, Jiacheng Sun, Guozhu Tan, Xiaomao Li, RE-Det3D: RoI-enhanced 3D object detector, Image Vis. Comput. 121 (2022) 104430.

[52] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, Baoquan Chen, Pointcnn: Convolution on x-transformed points, Adv. Neural Inf. Process. Syst. 31 (2018).

[53] Charles R. Qi, Xinlei Chen, Or Litany, Leonidas J. Guibas, Imvotenet: Boosting 3d object detection in point clouds with image votes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4404–4413.

[54] Vishwanath A. Sindagi, Yin Zhou, Oncel Tuzel, Mvx-net: Multimodal voxelnet for 3d object detection, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 7276–7282.

[55] Danila Rukhovich, Anna Vorontsova, Anton Konushin, Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2397–2406.