Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070812
ISSN 2371-8838 Vol. 7 Num. 8

Multi-modal Feature Fusion 3D Object Detection

Yiwen Jin, Rong Zhang, Yisu Hu, Hongliang Luo, Yongqiang Bai

Zhejiang Wanli University, Ningbo, Zhejiang, 315000, China

Keywords: Multi-modal; 3D Object Detection; Feature Fusion; point cloud; image

Abstract: For the existing 3D small object detection is prone to false detection and missed detection and other deficiencies. A 3D object detection method based on multi-modal feature fusion is proposed. Firstly, a feature extraction module is designed. The input image data is down-sampled through the image feature extraction network, and the input point cloud data is sampled and grouped through the point cloud feature extraction network to obtain the feature information at different scales. Secondly, a multi-modal feature fusion module is constructed to realize the point correspondence between point cloud features and image features by projection operation, and then the image features and point cloud features are splicing and fused to generate the final point cloud features to compensate the deficiency of single modal feature information. The experimental results show that compared with the existing algorithms, the algorithm in this paper improves the average detection accuracy of small object by 2.03%.

1. Introduction

3D object detection is a significant research area in the field of computer vision, with extensive applications in domains such as autonomous driving, robotics, medical research, and security systems [1]. According to the different input data modalities, 3D object detection can be divided into two detection methods, image-based and point cloud-based [2]. Image-based 3D object detection utilizes 2D object detectors to generate 2D bounding boxes on images. Then, by combining geometric principles, the method derives 3D bounding boxes for the detected objects. In 2019, Garrick Brazil et al. proposed the M3D-RPN method, which analyzes and processes the input data to obtain dense 3D candidate boxes using prior information about the target scene. The method then utilizes depth-aware convolutional layers to learn spatially aware features and predict 3D object boxes [3]. However, this method overlooks the differences between occluded objects and non-occluded objects, leading to suboptimal detection accuracy. In 2021, Zhang et al. proposed the MonoFlex. This method introduces an edge fusion module to extract the position coordinates and angle parameters of the four boundaries from the feature map and performs offset processing on them. Then, it utilizes convolutional layers to learn the features of the objects. Finally, the boundary parameters are fused with the features to achieve object detection [4]. In 3D object detection, images can provide rich information such as appearance, color, and texture. However, they lack depth information, which makes it difficult to determine the pose, size, and dimensions of the objects.

3D object detection based on point clouds utilizes laser radar to scan real-world scenes and generate a large collection of points. Through a 3D object detection network, object features are extracted, resulting in 3D bounding boxes. In 2016, Qi et al. first proposed the PointNet network, which introduced spatial transformation networks and MaxPooling to address the rotation invariance and unorderedness issues in point clouds, enabling effective detection of objects in point cloud data [5]. In 2017, Qi et al. further developed the PointNet++ algorithm, which introduced a multi-scale feature extraction structure. The algorithm starts by using the farthest point sampling (FPS) algorithm to select a subset of key points from the point cloud. For each selected key point, a local region is defined, and the k nearest neighbors within that region are grouped together. These k points are then fed into the core PointNet network for feature extraction, thereby improving the accuracy of 3D object detection [6]. In 2018, Yan et al. proposed the SECOND network, which replaced regular 3D convolutions with sparse 3D convolutions to improve network training speed. They also introduced an orientation regression loss to enhance orientation estimation performance. However, when the number of points and voxels exceeds the capacity, they can be discarded, resulting in feature loss and impacting the performance of object detection [7]. These 3D object detection methods are able to preserve the spatial features of objects in 3D space quite well. However, they lack information such as color and texture, which results in suboptimal precision in object detection [8].

Therefore, a multi-modal feature fusion-based 3D object detection approach is proposed, which integrates the depth information and contour information of point clouds with the color and texture information of images. This approach can address the issues of missed detections caused by sparse point cloud data for small objects and false detections caused by the absence of features in a single modality, thereby improving the detection accuracy. The main contributions of this paper can be summarized as follows:

1) The input images and point cloud data are processed separately by image and point cloud feature extraction networks to extract features at different scales, thereby increasing the receptive field and improving the detection performance for small objects.

2) By using projection operations, the coordinates of image and point cloud features are aligned. Then, through concatenation fusion, the features are combined to compensate for the limitations of single-modal data, augmenting the feature information of the objects and enhancing the performance of 3D object detection.

2. Proposed algorithm

The diagram of the proposed algorithm is shown in Figure 1, which mainly consists of three modules, feature extraction module, feature fusion module, and detection heads. First, the image data and point cloud data are separately input into the feature extraction module. The image data is processed by an image feature extraction network to perform down-sampling and extract image features. Simultaneously, the point cloud data is processed by a point cloud feature extraction network to perform sampling and grouping operations for extracting point cloud features. Next, the image features and point cloud features are fed into the multi-modal feature fusion module. By using a projection operation, the point cloud features are aligned with the corresponding coordinates of the image features. Then, the image features and point cloud features are concatenated and fused together to generate the final fused point cloud features. Finally, the fused point cloud features are input into the detection head, where an RPN network and an ROI Pooling module generate object detection boxes, completing the 3D object detection process.

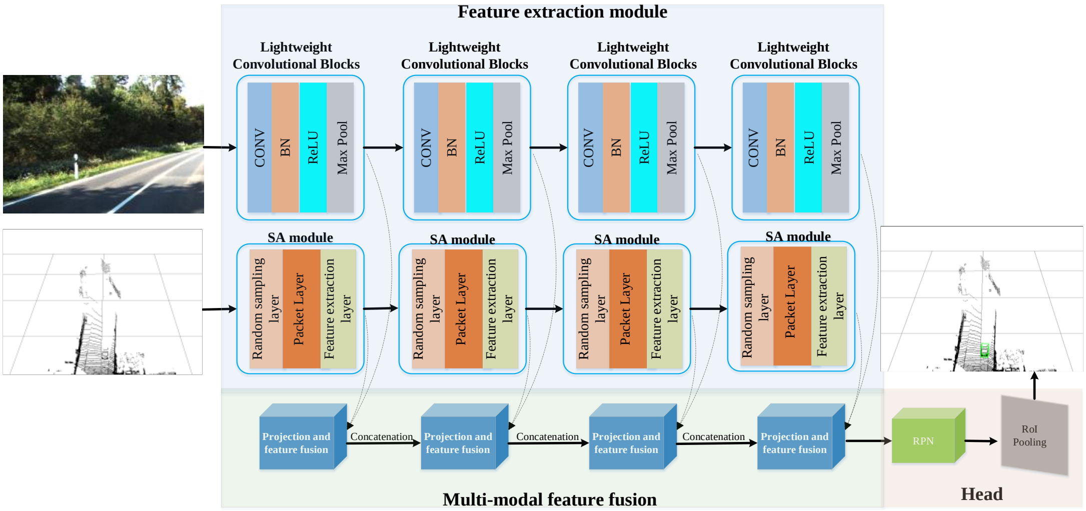

Figure 1: Algorithm diagram

2.1 Image feature extraction

The image data with dimensions of H×W×C is fed into the image feature extraction network. To enhance the detection performance for small objects, four lightweight convolutional blocks are utilized. Each block consists of a 3×3 convolutional layer, a batch normalization layer (BN), a ReLU activation function, and a max pooling layer (MaxPool). These operations down-sample the image by a factor of two and extract features, enlarging the receptive field to enable subsequent convolutional kernels to learn more comprehensive feature information. After four down-sampling operations, image features at different scales F_i(i=1,2,3,4) are obtained, providing color, texture, and other feature information at various scales for 3D object detection, as shown in Figure 1.

2.2 Point cloud feature extraction

The point cloud data with dimensions of H×W×D×C is fed into the point cloud feature extraction network. To capture local features of 3D feature maps at different scales, four SA (Sampling and Aggregation) modules are employed. Each SA module consists of a random sampling layer, a grouping layer, and a feature extraction layer. First, the random sampling layer uses FPS (farthest point sampling) method to randomly select n center points from the input point cloud. Then, these center points are passed to the grouping layer, where points within a radius of r from each center point are grouped together using the ball query method. Finally, the n groups are input into the feature extraction layer, where the features of the center points are extracted as global features for each group using MLP (multi-layer perceptron) and max pooling layers. The extracted features are concatenated to obtain the point cloud feature P_1. As the number of SA modules increases, the number of selected center points decreases, but each center point contains more feature information. After four SA modules, point cloud features P_i(i=1, 2, 3, 4) are obtained at different scales.

2.3 Multi-modal feature fusion

Due to the information loss caused by feature extraction from a single modality, this paper constructs a multi-modal feature fusion module, including four projection layers and a feature fusion layer. The point cloud features extracted initially are projected to generate image features, establishing correspondences between point clouds and images. Then, in the feature fusion layer, the point cloud features are concatenated and fused with the image features, leveraging the information from the image features at different scales to enhance the point cloud features. This process results in fused point cloud features at four different scales. Finally, the fused point cloud features are obtained by concatenating and fusing the features.

2.3.1 Feature projection

The extracted image features F_i and point cloud features P_i are separately input into the multi-modal feature fusion module. First, the point cloud features P_i are passed through the projection module, where they are transformed into projected image features using the coordinate transformation formula. The projection relationship is used to determine the correspondence between the point cloud features P_i and the image features $\mathrm{F_i}$ at different scales. For a point P(Xw,Yw,Zw) in the point cloud, its corresponding point P'(X,Y) in the projected image is calculated using the following formula.

$$(X, Y, 1) = U[R(X_W, Y_W, Z_W) + T \tag{1}$$

Where R denotes the rotation matrix, T denotes the translation vector, U denotes the camera intrinsic matrix, and 1 is meaningless in the projection space.

Then, the sigmoid function is used to calculate weights for each point in the image, aiming to enhance the effectiveness of the image features. The formula is as follows:

$$F_{f1} = f_{sigmoid}(F_1) \tag{2}$$

Where F_1 denotes the image features, $f_{sigmoid}$ denotes the weighted calculation of the image features, and F_{f1} denotes the image features after weighting.

2.3.2 Feature fusion

Due to the issue of missing feature information in a single modality data, different scales of point cloud features and image features are fused through concatenation to compensate for the limitations of a single modality data and enhance the feature information of the objects. As shown in Figure 2.

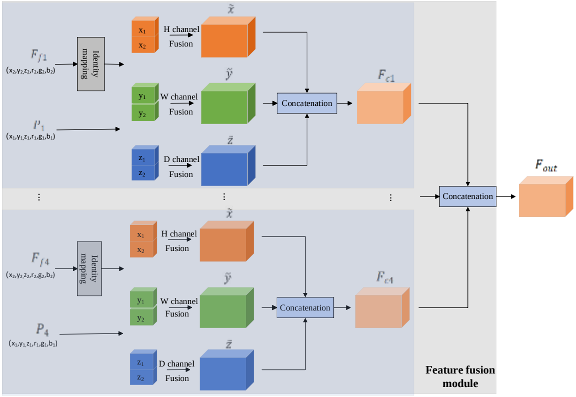

Figure 2: Feature Fusion Module

First, the weighted image feature $\mathrm{F_{f1}}$ and the point cloud feature P_1 are separately input into the fusion module. To achieve feature fusion, the feature representation of each point in the point cloud feature P_1 is denoted as $(x_1, y_1, z_1, r_1, g_1, b_1)$, and the feature representation of each point in the image feature F_{f1} is denoted as $(x_2, y_2, z_2, r_2, g_2, b_2)$. Since the image feature F_{f1} lacks feature information in the z-axis direction, an identity mapping operation is applied to fill the missing z-axis feature information $\mathrm{z_2}$ with zeros. Then, the feature information on the *x*, *y*, and *z* axes is separately input into the H, W, and D channels, respectively. Within each channel, the features are fused to increase the feature information in each dimension, resulting in features $\tilde{x}$, ỹ, and $\tilde{z}$. Finally, the features from different dimensions are concatenated and fused to generate the fused point cloud feature F_{c1}. The formula is as follows:

$$\tilde{x} = (x_1 + x_2) * k_x \tag{3a}$$

$$\tilde{y} = (y_1 + y_2) * k_y \tag{3b}$$

$$\tilde{z} = (z_1 + z_2) * k_z \tag{3c}$$

Where x_1, y_1, and z_1 denote the feature information in different dimensions of the point cloud feature P_1, x_2, y_2, and z_2 denote the feature information in different dimensions of the image feature F_1, k_x, k_y, k_z denote the convolution kernels in different dimensions, and $*$ denotes convolution.

Therefore, we can obtain the fused point cloud features F_{ci}(i=1,2,3,4) at four different scales. Finally, these features are concatenated and fused to output the fused point cloud feature $\mathrm{F_{out}}$, which enables the fusion of image and point cloud features, compensating for the limitations of single-modal data and improving the performance of 3D object detection. The formula for this process is as follows:

$$F_{out} = \sum_i F_{ci} * K_i \tag{4}$$

Where K_i denotes the convolutional kernel for different scale feature channels, and $*$ denotes the convolution operation.

2.4 Head

The detection head takes the fused point cloud feature p from the feature fusion module as input. Firstly, it passes through the RPN network to perform classification and compute the offsets for the original point cloud coordinates, generating object candidate boxes. Then, these candidate boxes are inputted into the ROI Pooling module to map them to corresponding positions in the feature map. The mapped candidate boxes are divided into regions of the same size, and max pooling is applied to each region. Finally, fixed-sized object detection boxes are obtained.

2.5 Loss function

The main loss function in the paper is the RPN loss function. The paper designs the RPN loss as a combination of classification loss and regression loss, with the following formula:

$$L(\{p_i\},\{g_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\,(p_i, p_i^*) + I\,\frac{1}{N_{reg}}\sum_i p_i^*\, L_{reg}(g_i, g_i^*) \tag{5}$$

Where N_{cls} and N_{reg} denote the total number of objects and the size of the objects, respectively. p_i and p_i^* denote the predicted probabilities of being an object, while g_i and g_i^* denote the predicted bounding box offsets and the actual offsets, respectively. The value of I is 10, which is used to balance the weights of the classification loss and the regression loss. The classification loss is computed using the cross-entropy loss function, and the regression loss is computed using the $Smooth_{L1}$ loss function.

3. Experimental results and analysis

To verify the effectiveness of the proposed algorithm, experiments were conducted on the public datasets KITTI and nuScences, and the results are compared with other algorithms. The experimental equipment is Ubuntu 18.04 operating system, NVIDIA GeForce RTX3090 GPU server. The proposed algorithm is based on python 3.7, Pytorch 1.6.0 and the CUDNN 8.1.0 framework implementation, batch size set to 4 and learning rate set to 0.01.

3.1 Dataset and evaluation metrics

The KITTI dataset is currently the most widely used dataset in the field of 3D object detection [9]. It consists of 7,481 training samples. In this paper, the training samples are divided into a training set and a test set in approximately a 1:1 ratio. The training set contains 3,712 samples, while the test set contains 3,769 samples. The evaluation in this paper focuses on three classes: Car, Pedestrian (Ped.), and Cyclist (Cyc.) For each class, the dataset is further divided into three difficulty levels based on the size and occlusion of the 3D objects: easy, moderate, and hard. The performance of the models trained in this paper is evaluated using the Average Precision (AP) metric, specifically the Average Precision at 40 recall positions (AP_{R40}). The evaluation is conducted on the test set, and the AP is calculated separately for the Car, Pedestrian, and Cyclist classes. The Intersection over Union (IoU) threshold for Car is set to 0.7, while for Pedestrian and Cyclist, it is set to 0.5. The APR40 is used as the evaluation measure for the experimental results in this paper. The official evaluation metrics provided by the dataset are employed for the evaluation.

3.2 Evaluation on the KITTI Dataset

3.2.1 Comparison of Evaluation Metrics

Table 1: Comparison of Car Category object Detection Accuracy

Method	Car 3D APR40(%)			AP(%)
	Easy	Mod.	Hard	
PointPillar	87.75	78.39	75.18	80.44
VoxelNet	89.01	82.36	79.91	83.76
SECOND	90.91	83.82	81.36	85.36
BtcDet	93.15	86.28	83.86	87.76
Proposed	94.57	88.10	83.59	88.75

Table 2: Comparison of Ped. Category object Detection Accuracy

Method	Ped. 3D APR40(%)			AP(%)
	Easy	Mod.	Hard	
PointPillar	57.30	51.41	46.87	51.86
VoxelNet	57.86	53.42	48.87	53.38
SECOND	62.18	57.88	49.05	56.37
BtcDet	69.39	61.19	55.86	62.15
Proposed	70.72	63.87	57.96	64.18

Table 3: Comparison of Cyc. Category object Detection Accuracy

Method	Cyc. 3D APR40(%)			AP(%)
	Easy	Mod.	Hard	
PointPillar	81.57	62.94	58.98	67.83
VoxelNet	79.15	57.75	51.10	62.67
SECOND	78.50	56.74	52.83	62.69
BtcDet	91.45	74.70	70.08	78.74
Proposed	92.08	76.16	72.08	80.11

To verify the effectiveness of the proposed algorithm, experimental tests were conducted on the

KITTI dataset to compare the experimental results of the proposed algorithm with those of mainstream algorithms. The comparative experimental results are shown in Tables 1, 2, and 3. The compared algorithms include PointPillar [10], VoxelNet [11], SECOND [7], and BtcDet [12]. The comparative experiments are conducted on different difficulty levels for the Car, Pedestrian, and Cyclist classes. The average precision (AP) on these three classes is improved by 0.99%, 2.03%, and 1.73%, respectively.

3.2.2 Visualization Analysis

The visualization of the 3D object detection algorithm on the KITTI dataset is shown in Figure 3 of the paper. A total of three sets of scenes are processed, and each set consists of four groups of images: RGB image, Ground Truth, BtcDet, and the visualized results of the proposed network.From the analysis of the point cloud visualization example in Figure 4a, it can be observed that the detection networks in the paper can effectively learn the information of the Car class and significantly improve the object detection accuracy. In Figure 4b, the BtcDet network produces numerous false positive results under occlusion, while the proposed detection network accurately detects the objects. In Figure 4c, the BtcDet network suffers from a high number of false positive detections for small objects, while the proposed network addresses this issue and accurately locates the car objects. These visualization results intuitively demonstrate the effectiveness of the proposed algorithm.

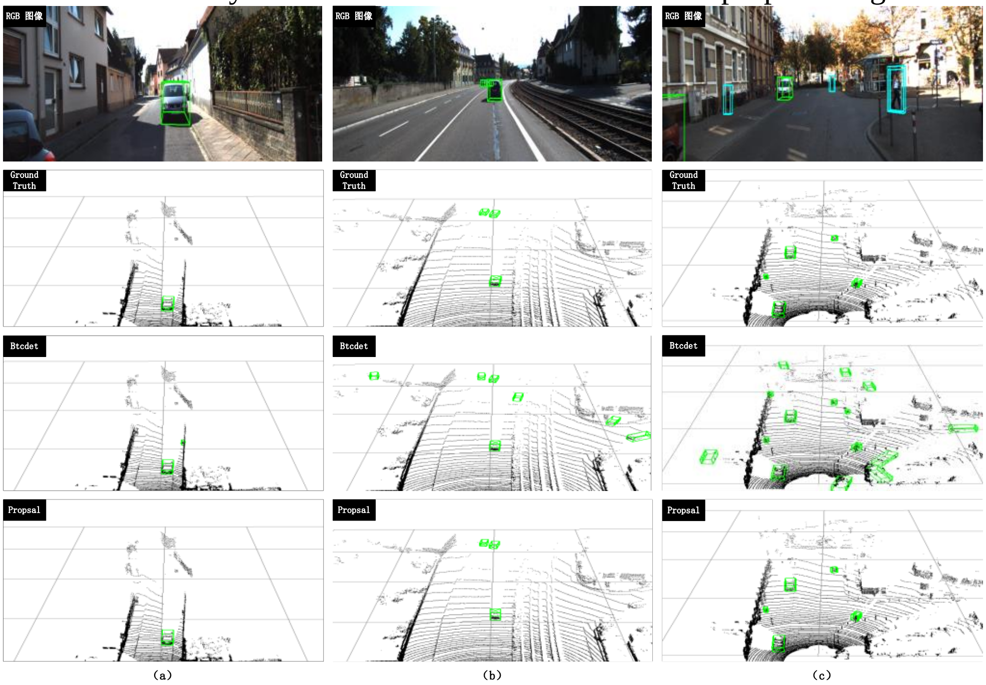

Figure 3: Visualization Image

3.3 Ablation experiments

The ablation experiments were conducted on the KITTI dataset, evaluating the 3D Average Precision (AP) at 11 recall positions (AP_{R11}) on medium-difficulty cars. The experiments were divided into three groups labeled (a), (b), and (c), as shown in Table 4. In method (a), the baseline model was used without any down-sampling or multi-modal feature fusion operations. In methods (b), (c), and (d), the output features of the point cloud branch after three subsequent SA (Set Abstraction) modules were fused with the features extracted from the image branch after three down-sampling operations, based on the baseline model of method (a).Method (d) represents the complete network architecture proposed in this paper. It extracts features from images and point clouds at four different scales and fuses them to enhance the detection accuracy of small objects and improve the overall precision of object detection. The performance of methods (b), (c), and (d) on medium-difficulty cars in terms of 3D AP_{R11} improved by 0.15%, 0.28%, and 1.00% respectively.

Table 4: Ablation Experiment

Method	Setting	(a)	(b)	(c)	(d)
Image feature extraction branch	Level1		√	√	√
	Level2			√	√
	Level3				√
Point Cloud Feature Extraction Branch	Level1		√	√	√
	Level2			√	√
	Level3				√
Result	3D APR11 (%)	87.10	87.25	87.38	88.10

4. Summary

The paper proposes a multi-modal feature fusion-based 3D object detection method. Firstly, a feature extraction module is designed to perform down-sampling on the input image data and sample grouping on the input point cloud data to obtain feature information at different scales. Secondly, a multi-modal feature fusion module is constructed to combine the feature information from both images and point clouds, compensating for the limitations of single data modalities and improving the detection performance for small objects. Experimental results demonstrate that the proposed algorithm achieves superior performance across multiple objective metrics. In future work, the detection of occluded objects will be further improved by incorporating prior information, enabling more accurate 3D object detection.

References

[1] Zhang Peng, Song Yifan, Zong Libo, et al. Advances in 3D Object Detection: A Brief Survey[J]. Computer Science, 2020, 47(4):94-102.

[2] Huang Zhe, Wang Yongcai, Li Deying. A survey of 3D object detection algorithms [J]. Chinese Journal of Intelligent Science and Technology, 2023, 5(01):7-31.

[3] Garrick Brazil, Xiaoming Liu. M3D-RPN: Monocular 3D Region Proposal Network for Object Detection[C]// IEEE/ CVF International Conference on Computer Vision (ICCV), 2019: 9286-9295.

[4] Yunpeng Zhang, Jiwen Lu, Jie Zhou. Objects are Different: Flexible Monocular 3D Object Detection[C]//Proceedings of the IEEE Computer Vision and Pattern Recognition(CVPR), 2021: 3289-3298.

[5] Charles R. Qi, Hao Su, Kaichun Mo, et al. PointNet:Deep Learning on Point Sets for 3D Classification and Segmentation [C]// Proceedings of the IEEE Computer Vision and Pattern Recognition(CVPR), 2017:652-660.

[6] Charles R. Qi, Li Yi, Hao Su, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space [C]//Proceedings of the Advances in Neural Information Processing Systems(NIPS), 2017:5105-5114.

[7] Yan Yan, Yuxing Mao, Bo Li. SECOND: Sparsely Embedded Convolutional Detection[J]. Sensors, 2018, 18(10): 3337.

[8] Gao Yue, Dai Meng, Zhang Qing. RGB-D Salient Object Detection Based on Multi-modal Feature Interaction [J]. Computer Engineering and Applications, 2022:1-11.

[9] Wei Liang, Pengfei Xu, Ling Guo. A survey of 3D object detection[J]. Multimedia Tools and Applications, 2021, 80(19): 29617-29641.

[10] Alex H. Lang, Sourabh Vora, Holger Caesar, et al. PointPillars: Fast Encoders for Object Detection from Point Clouds [C] //Proceedings of the IEEE Computer Vision and Pattern Recognition(CVPR), 2019: 12697-12705.

[11] Yin Zhou, Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018:4490-4499.

[12] Qiangeng Xu, Yiqi Zhong, Ulrich Neumann. Behind the Curtain: Learning Occluded Shapes for 3D Object Detection [C] //Association for the Advancement of Artificial Intelligence, 2021.