3D object detection method based on voxel index R-CNN

Dong Guo[a], Xianghua Ma*[a]
[a] School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shang hai,
* Corresponding author: xhuam@sit.edu.cn

ABSTRACT

In terms of 3D target detection, a voxel-based detection method is proposed: Voxel Index R-CNN, aiming at the balance between the accuracy of detection results and detection efficiency. In order to improve the timeliness of target detection, this paper proposes a voxel index query method, which uses the index difference constraint to reduce the computational loss of the query process for quantified spatial voxels. On this basis, a voxel feature extraction module suitable for it is designed. apply index query to optimize the ROI pooling layer to speed up voxel feature extraction. The experimental results on the 3D dataset of KITTI show that the results are 90.63%, 81.74%, and 77.23% on three different detection difficulty levels of cars, and the frame rate of detection processing is 28.5FPS. Compared with other methods, this method of a faster detection speed can be achieved while maintaining a higher accuracy rate.

Keywords: 3D object detection; Voxel index query; ROI pooling

1. INTRODUCTION

3D object detection is an important part in the field of autonomous driving and robotics, combining spatial information such as the pose and size of objects to provide input for downstream path planning and decision-making tasks. Although the recent development of deep learning has caused an upsurge in object detection using 2D images [1-3]. Due to the sparseness and unstructured characteristics of point clouds, it is difficult to apply these methods to 3D point clouds. In addition, 3D object detection applications have higher requirements on the timeliness of detection, which makes it difficult to design a point cloud space. 3D detection methods appear more difficult.

Existing point cloud-based 3D detection methods can be roughly divided into two categories, namely, voxel-based methods and point-based methods. The voxel-based methods VoxelNet[4], SCOND[5] and STD[6] divide the point cloud into voxels with a grid. Because of its regular spatial distribution, it is more suit-able for convolutional neural networks, and the efficiency of feature extraction is also high. Higher; the disadvantage is that the voxelized point cloud loses precise spatial information. Current point-based detection methods perform better. For example, PointNet [7] takes the original point cloud as input, and selects a part of key points to group and extract features. Point-based methods 3DSSD[8], SST[9] have excellent performance on experimental platforms such as KITTI[10], Waymo[11], although point-based methods have high detection accuracy, but in general, based on The point method is less efficient because the process of querying the nearest neighbor point cloud features using keypoints consumes a lot of computational resources.

As current detection methods mature, a new challenge is faced when deploying these detection methods into real-world application systems, how to design a method that is as accurate as point-based methods and as fast as voxel-based methods, in order to achieve this goal, we adopt a voxel-based approach and try to improve its accuracy. Through a comparative analysis of current detection methods, voxel-based methods usually perform detection on bird eye view (BEV) representations, even if the input data is 3D voxels. In contrast, point-based methods usually rely on keypoints to represent 3D features and fuse based on point-wise features. The main disadvantage of existing voxel-based methods is that they convert 3D feature volumes to BEV representations without ever recovering the 3D structure feature.

Based on the above analysis, a voxel-based detection method is designed in this paper: voxel-indexed R-CNN. Voxel index R-CNN adopts a new query method - voxel index query, which directly extracts 3D voxel features of key point neighborhoods and converts them into BEV representations. On this basis, 2D backbone network and RPN are applied to generate 3D regions. Proposal, optimize voxel set abstraction to aggregate voxel features, and perform feature extraction on 3D region proposal regions, avoiding the problem of missing 3D structural features caused by direct detection on BEVs.

Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022),
edited by Xianye Ben, Proc. of SPIE Vol. 12462, 124621C · © The Authors. Published under a
Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2661004

2. VOXEL INDEX R-CNN

In this section, the design of voxel index R-CNN, a two-stage 3D object detection framework based on RCNN, is introduced. As shown in Figure 1, the voxel index R-CNN network includes (1) a 3D backbone network, (2)) The 2D backbone network is followed by a region proposal network, (3) a voxel feature extraction module, and (4) a voxel region of interest (ROI) pooling network. In voxel indexed R-CNN, the original point cloud is firstly divided into regular voxels, and then the 3D backbone network is used for feature extraction, and then the sparse 3D voxels are converted into BEV representation, and the 2D backbone network is applied on this basis. and RPN to generate 3D region proposals. The voxel feature extraction module uses the voxel center where the key points are located to query and aggregate adjacent voxel features through the voxel index, and fuse with the 3D proposal region generated by the region proposal network (RPN). , which uses a voxel ROI pooling layer to extract ROI features and feed these features to the detection module for object classification and bounding box regression. These modules are discussed in detail below.

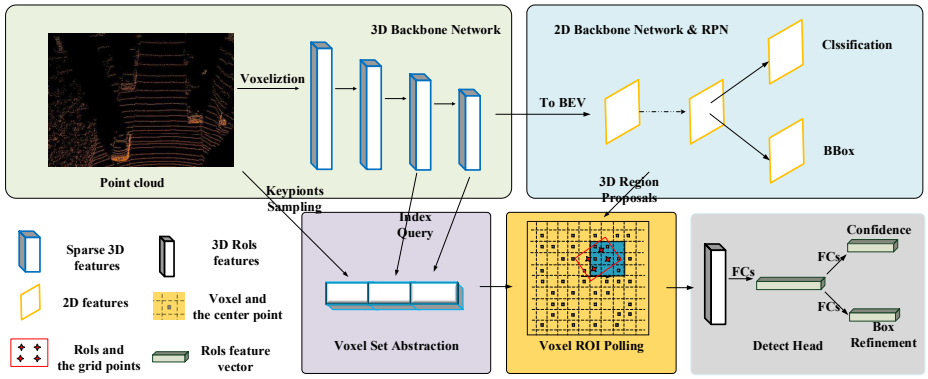

Figure 1 Voxel Index R-CNN Framework

2.1 Data acquisition module

The point cloud is voxelized according to the method in Voxelnet. Referring to the grid division method of the point cloud in PV-RCNN and the characteristics of the experimental data set KIITI, the size of the unit voxel is set as the detection of the vehicle category. , the paper clips the ground truth distribution of the point cloud at [0, 70.4], [−40, 40], [−3, 1]m along the X, Y, Z axes, which will get 352×400×40 Voxel collection.

The voxel index RCNN network model is constructed with reference to PV-RCNN. For the 3D backbone network, the 3D backbone network is mainly divided into four modules, and the number of convolution kernels is 16, 32, 48, 64 respectively. The point cloud is converted into gridded features, and the feature maps obtained by downsampling are stacked together along the Z axis and con-verted into BEV feature maps. The 2D backbone network consists of two modules: a feature extraction sub-network and a multi-scale feature fusion sub-network that upsamples and concatenates top-down features. Both modules con-sist of 5 convolutional layers with feature dimensions of (64, 128) respectively, the first module maintains the same resolution as the 3D backbone output in the X and Y axes, while the second module's resolution rate is half of the former.

2.2 Voxel feature extraction and ROI pooling

The voxel index R-CNN designs a voxel ROI pooling layer to aggregate spatial information from 3D voxel features, and uses voxel grid space to represent points. The positions and features use non-empty voxel center coordinates respectively $\{v_i = (x_i, y_i, z_i)\}$, where the range of i, j, k is (0,N) and eigenvector represented by $\{\varepsilon_i\}_{i=1}^{N}$.

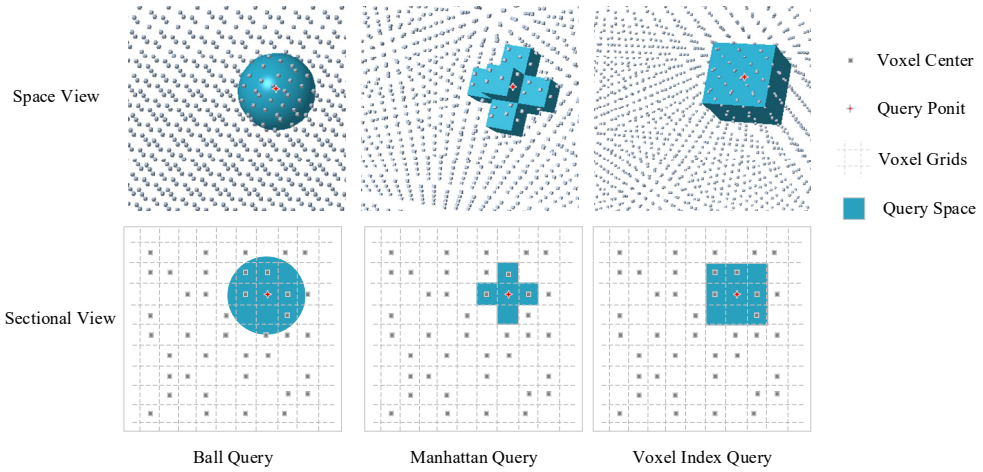

Figure 2 Spatial and Sectional Representations of Ball, Manhattan Distance, and Voxel Index Query

2.2.1 Voxel index

Compared with the disordered point cloud, the voxels are regularly arranged in the quantized space, which is easy to query. Inspired by the traditional ball query and Manhattan distance query methods, this paper proposes a method called difference index query, Constrain the index i, j, k of the key voxel $\alpha(i, j, k)$, so that $|\Delta i|, |\Delta j|, |\Delta k| \in (0, K)$ will obtain (2K+1) 3-1 voxels adjacent to the key point voxel, for example, K=1, will obtain 26 voxels around the key point voxel; K = 2, obtain 124 voxels in two layers adjacent to the key point voxel. For the traditional ball query to sample uniformly distributed voxels through farthest point sampling, it is necessary to calculate the distance D between all non-empty voxels and key point voxels to determine whether they belong to the sampling range; while the voxel-based Manhattan Distance sampling method requires It is to determine whether the voxel is in the sampling range by calculating the Manhattan distance between the voxel and the non-empty voxel. These two methods need to consume a lot of computing resources. In contrast, index query only needs to determine whether the voxel of the query is empty, avoiding many distance calculations and improving the efficiency of the query.

2.2.2 Voxel Feature Extraction Module

2048 key points are sampled from the original point cloud through FPS, which can ensure that the sampled key points can be evenly distributed in non-empty voxels, which better represent the entire scene, and then pass these key points to the 3D backbone network. Each module performs voxel feature extraction. Specifically, it determines the voxel $\{v_i = (x_i, y_i, z_i)\}$ where the key point p_i is located and uses the center coordinates of the voxel to aggregate the features of adjacent non-empty voxels through index query to obtain accurate position information and irregular voxels. features, and finally input these voxels semantic information together with the 3D proposal regions generated by the region proposal network into the voxel ROI for pooling.

2.2.3 Voxel ROI pooling layer

First, based on the original voxel, the proposed region of fusion voxel features is divided into G×G×G regular sub-voxels, and the center point of the sub-voxel is the same as the original voxel. Because the 3D voxels are too sparse, and the space occupied by non-empty voxels is less than 3%, the maximum pooling layer in Fast-RCNN cannot be used to directly extract features for each sub-voxel. The features of the voxels are aggregated together. For example, given the center point g_i, the paper first uses the index difference query to obtain the set of adjacent voxels, and then uses the Point Net module to aggregate the features of these adjacent voxels:

$$\zeta_i = \max_{k=1,2,\cdots,K} \{\sigma(v_i^k - g_i; \varepsilon_i^k)\} \qquad (1)$$

Among them, the maximum pooling operation $max(\cdot)$ represents the aggregated feature vector ξ_i obtained along the number of channels, and $\sigma(\cdot)$ represents a layer of multilayer perceptron, ε_i^k represents the voxel feature of v_i^k,

$$\hat{l_i}(\mathrm{IoU_i})=\begin{cases}0 & \theta_L > \mathrm{IoU_i} \\ \dfrac{\mathrm{IoU_i}-\theta_L}{\theta_H-\theta_L} & \theta_L \le \mathrm{IoU_i} \le \theta_H \\ 1 & \mathrm{IoU_i} > \theta_H\end{cases} \quad (2)$$

$v_i - g_i$ represents the relative coordinate between v_i and g_i. In order to improve the computational efficiency, this method does not aggregate the voxel features of the four stages in the 3D backbone network as in PV-RCNN, but only aggregates the voxel features of the latter two stages of the 3D backbone network. For each stage, set the size of the index difference to group voxels of multiple scales.

2.3 Loss function

Referring to the loss function used in PointPillars [12], the loss function of RPN is determined, which is composed of the object classification loss function and the bounding box regression loss function. The formula is:

$$L_{RPN} = \frac{1}{N_f}[\sum_i L_{cls}(\mathrm{p}_i, \hat{\mathrm{c}_i}) + \mathbb{Q}(\hat{\mathrm{c}_i} \ge 1)\sum_i L_{reg}(\sigma_i, \hat{t_i})] \quad (3)$$

Among them, p_i and σ_i represent the output of target classification and bounding box regression, respectively, $\hat{\mathrm{c}_i}$ and $\hat{t_i}$ represent the classification label and regression target, respectively, $\mathbb{Q}(\hat{\mathrm{c}_i} \ge 1)$ means that only the regression loss of anchor points on foreground objects is calculated, and N_f represents the number of anchor points on foreground objects. The object classification function is expressed by Focal loss [13] as:

$$L_{cls} = -(1-\mathrm{p}^t)^{\gamma}\log(\mathrm{p}^t) \quad (4)$$

γ is a constant, when it is 0, L_{cls} is consistent with the cross-entropy loss function, and p^t is the probability of the current prediction.

The regression loss function is represented by Huber loss, and the formula is:

$$L_{reg} = \begin{cases}\dfrac{1}{2}(\mathrm{y}-\mathrm{f}(\mathrm{x}))^2 & |\mathrm{y}-\mathrm{f}(\mathrm{x})| \le \delta \\ \delta|\mathrm{y}-\mathrm{f}(\mathrm{x})| - \dfrac{1}{2}\delta^2 & \text{else}\end{cases} \quad (5)$$

δ is a hyperparameter. When δ tends to 0, it will tend to mean absolute error; when δ tends to ∞, it will tend to mean square error.

2.4 Detection module

The detection module takes the region-of-interest features as input for bounding box regression. Specifically, the shared 2-layer MLP first converts ROI features into feature vectors, and then passes the vectorized features into two parallel branches: one for object bounding box regression and the other for confidence prediction. The bounding box regression branch predicts the difference from the 3D proposal region to the ground truth, and the confidence branch predicts the confidence associated with the IOU.

3. EXPERIMENT

3.1 Training

The entire framework of Voxel Index R-CNN is optimized using the Adam optimizer. In the training phase, the network is set to train for 80 rounds, the number of samples inputted each time is 12, the learning rate is initialized to 0.01, and the cosine annealing strategy is used to update. In the detection module, the threshold for the foreground is set to 0.75, and the threshold for the background is set to 0.25. Threshold for bounding box regression is set to 0.6. In this paper, 128 regions of interest are randomly selected as the training samples of the detection module. Among the sampled ROIs, half are samples of, which coincide with the corresponding ground-truth boxes. In the inference stage, the paper first performs Non-Maximum Suppression on the RPN with a threshold of 0.7, and retains the first 80 proposal regions as the input of the detection module. After the bounding box regression prediction, NMS is applied again with a threshold of 0.1 to remove redundant predictions result.

3.2 Result

The results of Voxel Index R-CNN on the KITTI validation set for 3D target detection AP and BEV target detection AP are shown in Table 1, where the 9 threshold is 0.7.

Table 1 3D object detection AP and BEV object detection AP on KITTI val set

IOU	$AP_{3D}(\%)$			$AP_{BEV}(\%)$		
	easy	medium	hard	easy	medium	hard
0.7	91.67	85.36	82.75	94.53	92.32	89.76

Figure 3 shows the 3D detection results of three of these scenes on the KITTI test set.

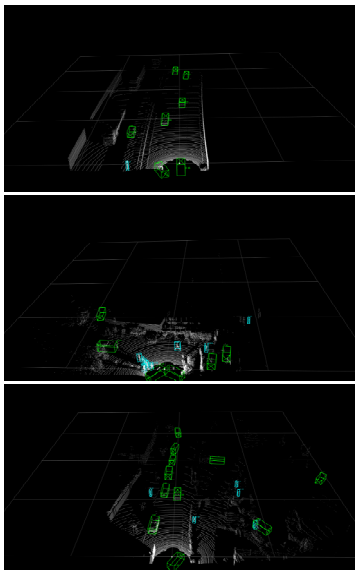

Figure 3 Detection effect on KITTI test set

The results of Voxel Index R-CNN are compared with several state-of-the-art methods on the KITTI test set. The results of the three detection difficulty levels of easy, medium and difficult are shown in Table 2.

Table 2. Time domain index statistics of vibration signal

Method	FPS (Hz)	$AP_{3D}\%$		
		easy	medium	hard
Image + Lidar				
MV3D	-	74.97	63.63	54.00
AVOD-FPN	10.0	83.07	71.76	65.73
PointSIFT+SENet	-	85.99	72.72	64.58
Lidar				
bashed on point				

STD	9.7	87.95	79.71	75.09
3DSSD	11.3	88.36	79.57	74.55
PV-RCNN	23.5	90.25	81.43	76.82
bashed on vexel				
VoxelNet	-	77.47	65.11	57.73
SECOND	25.7	83.34	72.55	65.82
Part-A2	-	87.81	78.49	73.51
PointPillars	36.4	82.46	74.28	68.61
SSA-SSD	24.1	87.68	79.76	73.89
Voxel R-CNN	25.2	90.79	81.62	77.05
Voxel Index R-CNN(ours)	28.4	90.63	81.74	77.23

The comparison results show that the voxel index R-CNN in this paper achieves a good balance between accuracy and efficiency among all methods. Using the voxel index query method, the voxel index R-CNN achieved an average accuracy (AP) of 90.63% on the medium and easy difficulty of the Car class, and reached an average accuracy of 81.74% on the medium difficulty. The average accuracy rate is 81.62%, and the frame rate of detection processing is 28.5FPS. Specifically, voxel-indexed R-CNN achieves comparable accuracy with the best performing models PV-RCNN and Voxel R-CNN. Voxel-indexed R-CNN is similar in structure to PV-RCNN, but the detection frame rate Compared with 4.9 FPS higher, it proves that the use of voxel index query indeed reduces the computational loss; the detection result is 0.12% and 0.22% higher than that of Voxel R-CNN at the medium and difficult levels, respectively, indicating that the feature extraction module has a good effect on the ROI area. It is effective to perform feature enhancement. Furthermore, the performance of voxel-indexed R-CNN is greatly improved over existing voxel-based models.

4. CONCLUSION

This paper proposes a 3D object detection method based on voxel index query - voxel index R-CNN. The model Voxel R-CNN takes voxels as input, first generates 3D region proposals from BEV feature representations, and utilizes a voxel ROI pooling layer to extract region features from 3D voxel features. The voxel feature extraction module fuses the extracted features with the 3D proposal regions generated by the RPN, and then uses a voxel region of interest pooling layer to extract features, which are fed to the detection module for object classification and bounding box regression. The test results on the KITTI dataset show that the voxel-indexed R-CNN in this paper achieves a good balance between detection accuracy and efficiency. The voxel-indexed R-CNN in this paper is a simple and effective 3D object detection method and it can be applied to the research of other downstream tasks such as autonomous driving and robotics.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China. (Grant:2020YFB2007700)

REFERENCES

[1] Ren, S., et al., Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015. 28.
[2] Szegedy, C., et al. Inception-v4, inception-resnet and the impact of residual connections on learning. in Thirty-first AAAI conference on artificial intelligence. 2017.
[3] Liu, W., et al. Ssd: Single shot multibox detector. in European conference on computer vision. 2016: Springer.
[4] Zhou, Y. and O. 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.
[5] Yan, Y., Y. Mao and B. Li, Second: Sparsely embedded convolutional detection. Sensors, 2018. 18(10): p. 3337.
[6] Yang, Z., et al. Std: Sparse-to-dense 3d object detector for point cloud. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
[7] Qi, C.R., et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[8] Yang, Z., et al. 3dssd: Point-based 3d single stage object detector. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[9] Yang, Z., et al. 3dssd: Point-based 3d single stage object detector. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.Geiger, A., et al., Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 2013. 32(11): p. 1231-1237.
[10] Geiger, A., et al., Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 2013. 32(11): p. 1231-1237.
[11] Sun, P., et al. Scalability in perception for autonomous driving: Waymo open dataset. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[12] Lang, A.H., et al. Pointpillars: Fast encoders for object detection from point clouds. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.