Multi-Stage Refinement Network for Point Cloud 3D Object Detection

Feng Wang Zhichao Li Yan Yan Naiyan Wang
Tusimple
{feng.wff, leeisabug, yy19940828, winsty}@gmail.com

Abstract

This report describes our multi-stage solution for the Waymo Open Dataset (WOD) challenge. Generally speaking, we split the space into voxels and extract features using PointNet in each voxel. The voxel grids are then organized under bird eye view, with the z-axis voxel features concatenated as channels. The 2D feature map is fed forward into a Faster RCNN-like detection network. Different from the upright objects in 2D detection, the objects in 3D detection all have orientations, so we add extra orientation targets for the RPN module. Then a rotated ROI align operation is applied on the feature map to extract object features for bounding box refinement and re-scoring. Finally, we extract the original point cloud using the refined bounding box and utilize a PointNet for further refinement. Our solution relies on single frame point cloud only, without temporal and camera information. Our solution won 3rd place in the WOD challenge, with the top two solutions all use temporal or camera information.

1. Introduction

LiDAR point cloud 3D object detection has attracted more and more attention in the autonomous driving community because of its high reliability, low environmental sensitivity and direct depth perception ability. Through cutting-edge 3D object detectors, autonomous driving systems can get very accurate 3d locations of objects, which are crucial information for downstream tasks.

With the rapid development of past years, the performance of 3D object detection has already saturated on the previously released dataset KITTI [2]. The community is looking for more challenging datasets, which should cover more scenes, categories and weather conditions. The newly released Waymo Open Dataset (WOD) [12] is a high-quality, large-scale dataset for autonomous driving. It contains source data from 5 LiDARs and 5 cameras, with temporal and calibration information. In the previous datasets, the LiDAR point cloud is usually presented as sparse and irregular 3d coordinates. But in WOD, the LiDAR data are in the form of range images, with the points from the same beam successively arranged in one row. The range image form opens a door for further studies of dealing with the LiDAR point cloud just like camera image processing.

In this challenge, we still treat the point clouds as sparse data and process them in bird-eye-view (BEV). Our solution is a three-stage detection framework, with a two-stage CNN detector and one stage of PointNet refinement. During the competition, we have tried several techniques to improve performance. Among those techniques, we think three of them are worth to be discussed:

(1) We find that dividing the z-axis into multiple bins performs better than Point Pillars' encoding, which has no z-axis division. Thanks to the Dynamic Voxelization[15], there is no extra computation and memory cost for dividing the z-axis into more voxels.

(2) When doing random object insertion, we propose to remove the points which are occluded by the newly inserted objects, which can make better simulation and detection performance.

(3) We discover an ambiguity problem in the classification target for the PointNet object refinement module[9]. To address this problem, we propose to add virtual grid points into the pooled real points to eliminate the ambiguity.

2. Methodology

Our overall network architecture is shown in Figure 1. It consists of a voxel feature extractor, a backbone, a Region Proposal Network (RPN), a RCNN header and a PointNet refinement module. In this section, we will introduce these modules.

2.1. 3D Voxel Feature

Point Pillars [5] proposes to voxelize the space into 2D grids, with no division on heights. The benefit of this method is that the generated feature map can be directly fed into modern 2D detectors. However, Point Pillars' features may lose information on the z-dimension. With only one fully connected layer and a max-pooling layer, Point Pillars only extracts features on the surface of each voxel, ignoring

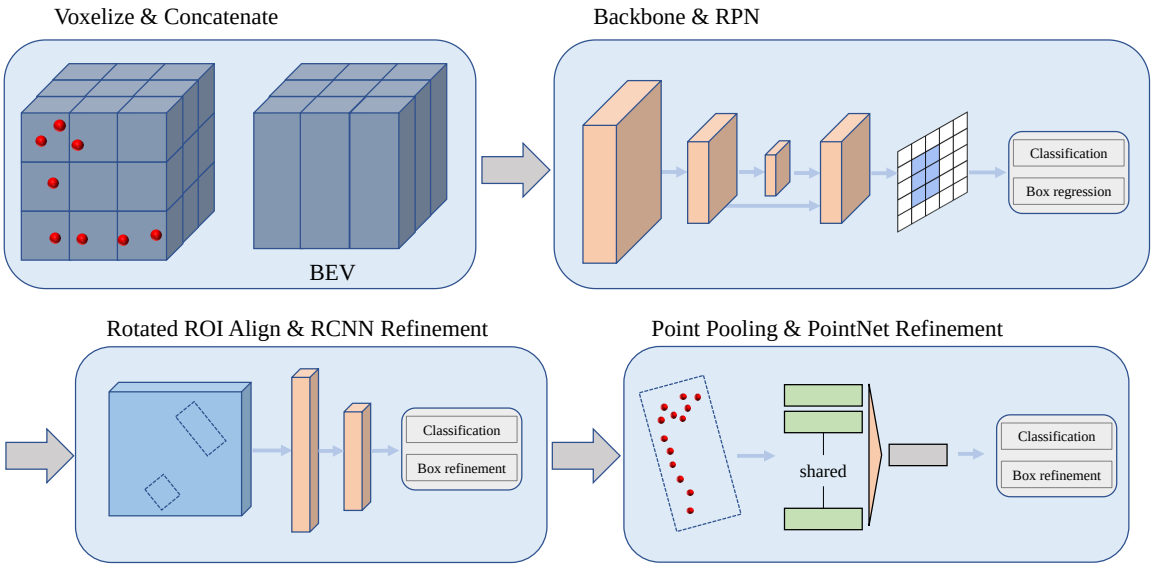


Figure 1. The whole pipeline of our solution.

the points inside[1].

We propose two ways to solve this problem. The first is simply using two fully connected layers instead of one to get more complex space partition. The second is to divide the z-axis into multiple bins, i.e. voxelize the space into 3D grids. Thanks to the Dynamic Voxelization[15], there is no extra computation nor more memory consumption if the number of z-bin $\times$ feature dimension equals the feature dimension in one pillar.

2.2. Detection Network

Our detection network is modified based on Faster RCNN[7]. To handle the orientation of the objects, we add two extra regression targets, sine and cosine values of the yaw angles in RPN. During inference, the orientation is calculated as the arctan of the two values.

For the second stage, we use Rotate ROI Align [6] instead of original ROI Align to extract features in the rotated bounding box. The regression target is set to the offset of each bounding box parameters in the canonical coordinates of the bounding box. Since the overlap threshold of the metrics are based on 3D IoU but we only have 2D boxes in RPN stage, we add 0.1 to the overlap threshold to get the classification label for the RCNN stage, i.e. 0.8 for vehicle and 0.6 for pedestrian and cyclist.

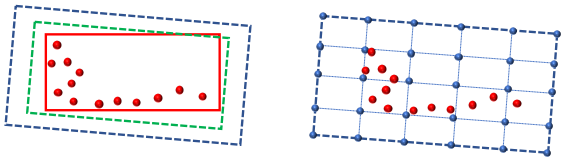
[1]Weight and bias in the FC layer can be seen as multiple one dimension lines in the feature space, and the features are projected on the lines to get activations. Through max-pooling, only the max activation could be reserved.

2.3. PointNet Refinement

The PointNet bounding box refinement was firstly proposed in Point RCNN [9]. The deep CNN features are highly entangled and down-sampled. They usually contain more semantic information than localization information. To get the accurate localization information back, we utilize the Point RCNN module to regress bounding box refinement parameters from the original points.

During experiments, we find that the Point RCNN module suffers from an ambiguity problem on the classification target. The classification target is usually calculated based on the IoU between the proposal and the ground truth. Since there are few points outside of objects except for some ground points, the pooled points from an accurate bounding box and an enlarged bounding box are usually very similar (Figure 2a). It is very difficult for the Point RCNN module to distinct the size change of the proposal. As we described in the last section, the IoU threshold for classification is very high. Even a small size change may make the IoU decrease below the threshold, which will lead to the target change.

The core reason for this problem is that the Point RCNN module has no perception of the proposal size. In Part-A^2 Network [10], the authors propose to voxelize using the proposal to let the RCNN module get the proposal information. However, this solution only bypasses the ambiguity problem, not solves it. In this work, we propose to add some virtual grid points into the pooled real points (Figure 2b). The grid points are generated evenly within the proposal. We also append one binary feature to the points to indicate whether the points are real or virtual. Through the virtual points, the RCNN module will have the ability to perceive



(a) (b)

Figure 2. (a) Through point pooling operator, the two dashed bounding boxes extract the same points. However, the two dashed bounding boxes have different IoU with the ground truth (red box). This will confuse the Point RCNN module. (b) By adding the virtual grid points, we send the proposal bounding box information to the Point RCNN module. It will be able to know the proposal is too large because the border virtual points have no real points in their near neighbor.

the size of the proposal, leading to better classification performance.

2.4. Data Augmentation

Randomly sampling objects from other frames has been proven to be effective in point cloud object detection [14]. A typical pipeline is creating object database, sampling object from the database, discarding the objects which have overlap with other objects, estimating the ground plane and put the sampled objects on the ground. In this work, we propose to add one more step. Notice that the inserted objects may block the original laser radiation, some of the original points are no longer visible. To simulate this character, we create 3d convex hull for each inserted object to cover the object and the space it blocks. Then the original points in the convex hulls are removed. Object labels that are totally occluded will also be deleted (usually pedestrian). In figure 3, we illustrate the effect of our simulation algorithm.

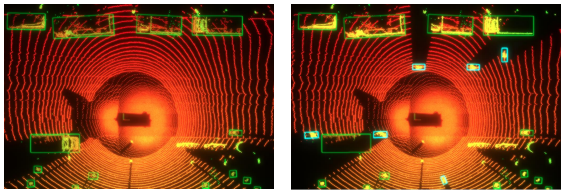
In WOD, the amounts of vehicle and pedestrian in each frame are sufficient enough. Almost all frames have vehicles and about 80% frames have pedestrians in it. However, only 19% frames have cyclists, which is much less than the other two categories. In this work, we only sample cyclists for augmentation to make the dataset more balance.

3. Experiments

In this section, we will describe our training details and experimental results to show the effectiveness of our proposed method.

3.1. Training Details

For the bottom feature representation, the resolution of the voxelized feature map is $1024 \times 1024 \times 8$. In each voxel, we use two-layer PointNet with nodes of $[16, 4]$ to extract the feature of points. We concatenate the features on the z-axis together as the feature of the pillar, so the input feature is $1024 \times 1024 \times 32$ for the backbone.



(a) Original point cloud. (b) Inserted some cyclists.

Figure 3. Effect of removing the occluded points after several cyclists are inserted to the point cloud.

method	vehicle	pedestrian	cyclist
baseline	0.5176	0.3318	0.3153
+3d voxel	0.5321	0.3535	0.3438
+high IoU thresh	0.5398	0.3567	0.3451
+double head[13]	0.5265	0.3342	0.3255
+filter bbox	0.5310	0.3234	0.3231
+cyc sampling	0.5254	0.3447	0.4321

Table 1. Ablation studies on validation set for the first two stages. Here the "filter bbox" means to remove bounding boxes that has no points in it. "high IoU thresh" is described in Section 2.2.

method	vehicle	pedestrian	cyclist
baseline	0.5517	0.3294	0.3223
no grid	0.5444	0.3727	0.3353
grid	0.571	0.4119	0.4098
+high IoU thresh	0.6066	0.435	0.416

Table 2. Ablation studies on validation set for the Point RCNN refinement. This baseline is based on the 3d voxel and filter bbox tricks.

We choose ResNet50[3], ResNet101[3], SENet50[4] and HRNet[11] as our backbone to train different models for ensemble. We use SGD with cosine learning rate to optimize our network. The model is trained with 36 epochs. During inference, top 1000 proposals are kept for RCNN and PointNet refinement. We use weighted NMS with 0.1 IoU threshold for suppression and 0.5 IoU threshold for weighted sum.

Our method is implemented in MXNet [1]. All experiments are trained using NVIDIA 2080ti distributedly.

3.2. Results

In Table 1, we present the ablation study of all the improvements we have tried. The cyclist sampling method improve the performance of cyclist dramatically. Vehicle and pedestrian's performance are also enhanced a little because removing the occluded points can be seen as data augmentation, and this kind of occlusion may happen in real world. Table 2 shows the improvements of the Point RCNN module. From the table we can infer that adding the virtual grid points promotes the performance of all the categories significantly.

Table 3 shows the final results of WOD challenge. Note that we do not use the temporal information in our model.

method	Frames	All NS
HorizonLiDAR3D	5	0.7711
PV-RCNN[8]	2	0.7152
TS-LidarDet(ours)	1	0.6553
Simple Baseline v2[10]*	4	0.6365
Det3D-Waymo-3D-FS-VS[16]*	1	0.6304

Table 3. Overall performance on testing set. * means the results are re-implemented and submitted by other researchers, not the original authors.

There are lots of parking lot scene in WOD. The objects are heavily occluded by each other. Many objects have few points in it. Fortunately, the lost information can be retrieved from previous frames. How to utilize the temporal information will be very interesting topic in the future.

4. Conclusion

In this report, we present our multi-stage refinement network for the WOD challenge. We analyze some problems in current state-of-the-art 3D object detection algorithms and propose three novel improvements to promote the performance.

References

[1] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

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

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

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

[6] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based cnn for ship detection. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 900–904. IEEE, 2017.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[8] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. *arXiv preprint arXiv:1912.13192*, 2019.

[9] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[10] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[11] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[12] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. *arXiv*, pages arXiv–1912, 2019.

[13] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Double-head rcnn: Rethinking classification and localization for object detection. *arXiv preprint arXiv:1904.06493*, 2, 2019.

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

[15] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. *arXiv preprint arXiv:1910.06528*, 2019.

[16] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.