Study of 3D Object Detection with Normal-map on Point Clouds

TP18051 Jishu Miao Supervisor: Takayoshi Yamashita

1. Introduction

Object detection is one of the most crucial tasks in autonomous driving. In this task, both accuracy and speed are important. There are various methods have been studied to improve accuracy while accelerating the detection speed. Although RGB images captured by cameras are used for object detection, point clouds captured by LiDAR are also used for this task. Because it is insensitive to visible light and can capture objects day or night. However, point clouds are sparse and different from images in that the order is irregular, leading to a slow processing speed as 2D convolution cannot be performed directly. Surface normals extracted from the points have a better ability to represent the shape features of the object than 3D coordinates, which we believe will improve the performance of object detection. In this study, we propose a novel point clouds-based 3D object detection method for achieving higher-accuracy. The proposed method employs You Only Look Once v4 (YOLOv4) as a feature extractor and gives Normal-map as additional input. Our Normal-map is a three channels Bird-eye view (BEV) map, retaining detailed surface normal vectors. It makes the input information have more enhanced spatial shape information and can be associated with other hand-crafted features easily.

2. Bird-eye View Representation

The BEV map represents the point clouds as a 2D pseudo-image from the bird-eye view as shown in Figure 1. This approach converts the unordered point clouds into a sequence ordered image. Conventional methods are to generate three maps by a mapping function $\mathcal{P}_{\Omega i\to j}$, representing normalized point cloud density (R), maximum height (G), and maximum reflection intensity (B), as

$$\begin{aligned} z_r\left(\mathcal{S}_j\right) &= \min(1.0, \log(N+1)/64)\; N = \left|\mathcal{P}_{\Omega i\to j}\right|, \\ z_g\left(\mathcal{S}_j\right) &= \max\left(\mathcal{P}_{\Omega i\to j}\cdot[0,0,1]^T\right), \\ z_b\left(\mathcal{S}_j\right) &= \max\left(I\left(\mathcal{P}_{\Omega i\to j}\right)\right). \end{aligned} \tag{1}$$

Since 2D convolution can be applied to the BEV map, the detection task can be accelerated by using a fast object detection network such as YOLO[1]. However, different points may be arranged to a same pixel in the BEV map, which is less expressive than original data. Besides, the object shape will be lost due to the 3D data is compressed to 2D.

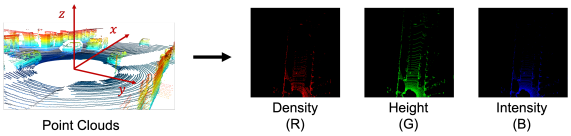


Figure 1 : Bird-eye View Representation.

3. Proposed Method

We propose a method for 3D object detection using BEV map with additional normal information. Figure 2 shows an overview of the proposed method.

3.1 Normal Feature Extraction

The normal vector is estimated from the pre-processed point clouds by Principal Component Analysis (PCA) with the search radius of 30cm and the maximum search number of 50 points. To make all normals point in the same direction, the normals opposing the LiDAR are reversed by an orienting system. Normal-map is generated for enabling the 2D convolution of each point's normals. The mapping function $f_{\mathcal{PS}}$ shown in Eq. (2) is used for creating the Normal-map, which allows us to use the normal vector $\vec{x}$, $\vec{y}$, $\vec{z}$ of the highest point in $\mathcal{P}_{\Omega i\to j}$ when mapping each point into 2D space.

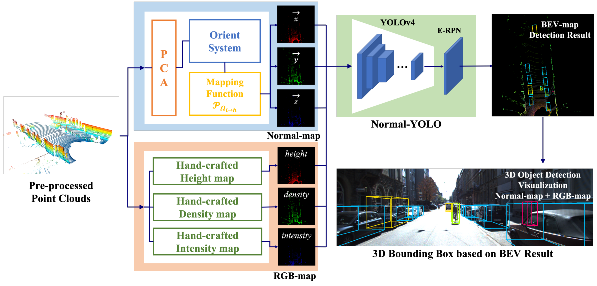


Figure 2 : Overview of Proposed Method.

$$\mathcal{P}_{\Omega i\to j} = \left\{\mathcal{P}_{\Omega i} = [x,y,z]^T \middle| \mathcal{S}_j = f_{\mathcal{PS}}(\mathcal{P}_{\Omega i}, g)\right\} \tag{2}$$

As shown in Eq. (3), the normal vectors of each point are extracted as $normal_{\vec{x}}$, $normal_{\vec{y}}$, $normal_{\vec{z}}$ from the $\mathcal{P}_{\Omega_{i\to h}}$ for each axis. The normal vectors are represented by a 3-channel 2D pseudo image, as

$$\begin{aligned} normal_{\vec{x}}(\mathcal{S}_j) &= \vec{x}\left(\mathcal{P}_{\Omega j\to h}\right), \\ normal_{\vec{y}}(\mathcal{S}_j) &= \vec{y}\left(\mathcal{P}_{\Omega j\to h}\right), \\ normal_{\vec{z}}(\mathcal{S}_j) &= \vec{z}\left(\mathcal{P}_{\Omega j\to h}\right). \end{aligned} \tag{3}$$

Since the search range of the normal estimation is wider than the pixel representation range of the BEV map, it can include a wider range of information. Moreover, since the normal-map calculated from the normal vectors is also a BEV map, it can be freely combined with other BEV maps to be used as the input data.

3.2 Input Details

Our network uses RGB-map and Normal-map as input. The RGB-map is similar to the BirdNet[2], and consists of the height map, the density map, and the intensity map. The Normal-map is the normal vectors in the x, y, and z axes. Thus, the input is a 6-channel BEV map consisting of these maps concatenated in the channel axis.

3.3 Object Detection Network

The network predicts the class and size of an object with the Euler-Region Proposal Network (E-RPN) for 3D object detection as shown in Figure 3. We employ YOLOv4 as the basis network for 3D object detection. E-RPN predicts the height, width, angle, objectness, and class probability of the bounding box coordinates. In this study, the number of object classes is 6: Car, Van, Truck, Person, Cyclist, and Tram. The loss function is based on Mean-Square Error.

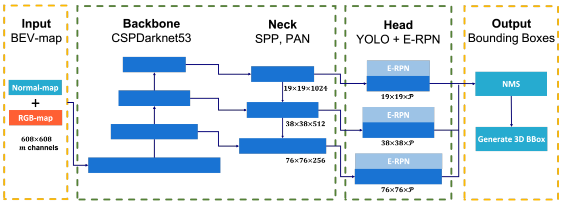


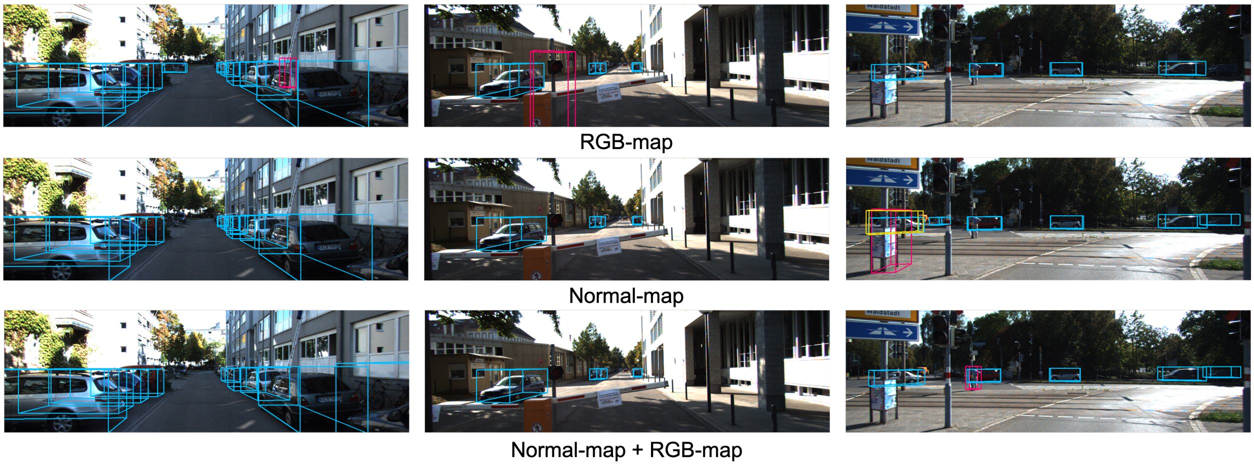
Figure 3 : Normal-YOLO Network Architecture.

4. Evaluation Experiments

In order to examine the effectiveness of the proposed method, some comparison experiments are conducted using the KITTI dataset. We also evaluate the accuracy of the object angle detection by adding a function to calculate the yaw angle.

Table 1 : Evaluation Results for Bird-eye View Performance on the KITTI Benchmark.

Method	FPS	Car			Pedestrian			Cyclist			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
BirdNet	9.1	**84.17**	59.83	57.35	**28.20**	**23.06**	**21.65**	**58.64**	**41.56**	**36.94**	45.93
Complexer-YOLO	16.7	77.24	68.96	64.95	21.42	18.26	17.06	32.00	25.40	22.88	38.68
Ours	5.5	72.84	**71.52**	**67.50**	26.71	21.19	20.17	42.50	36.06	31.18	43.30


RGB-map

Normal-map

Normal-map + RGB-map

Figure 5 : 3D Object Detection Visualization in Camera View.

4.1 KITTI Benchmark Evaluation Results

Table 1 shows the evaluation results. Compared with the conventional method, the proposed method achieves the detection accuracy of 72.84% in *Car* under the Easy mode, and the highest accuracy of 67.50% at the Hard mode. It achieves almost the same accuracy as BirdNet[2], which is also a BEV-based method. Even for the same class objects with different detection modes, our method shows a more robust performance by adding normal information. In addition, we achieve a higher Average Precision (AP) than Complexer-YOLO[3], which uses the same YOLO-based network. Although the input is BEV map, the accuracy for non-planar objects (e.g. pedestrians and cyclists) is better for each mode.

4.2 Evaluation of Angle Prediction

Since the normal is the object shape information, we assume that object angle accuracy can be improved by adding the Normal-map. Table 2 shows the result of calculating the average included angle θ_k from the estimated object angles and ground truth for 6 classes.

$$score_{class}\left(\theta_k\right) = \left(\frac{1}{n}\sum_{k=1}^{n} \arccos\theta_k\right)^{-1} \tag{4}$$

Normal-map improves the angle accuracy. In particular, the accuracy of objects with large and flat shapes like cars is further improved.

Table 2 : Yaw Angle Prediction of 6 Classes.

Input Map	Score of Angle Accuracy					
	Car	Van	Truck	Person	Cyclist	Tram
RGB	10.18	**7.49**	5.78	**2.22**	**4.44**	3.24
Normal	8.76	5.37	5.35	1.69	3.10	3.28
Normal+RG	9.27	6.43	**10.09**	2.14	3.34	3.55
Normal+RGB	**10.34**	6.57	8.89	2.06	3.86	**5.25**

4.3 Evaluation by Distance

Since the density of the point clouds changes depending on the distance, we evaluate the detection accuracy by distance. Figure 4 shows average precision over distance. From Figure 4, the detection accuracy of the group with added Normal-map does not decrease up to 30m, and the decrease is smaller than no-normal group even at 40m or more.

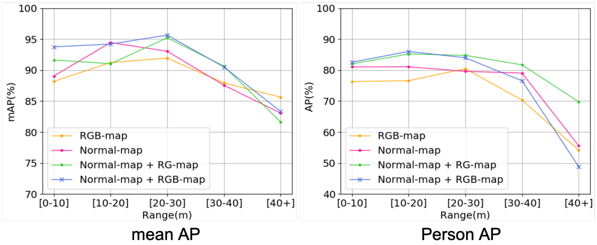


Figure 4 : Average Precision over distance.

4.4 Visualization Results

As shown in Figure 5, due to traffic lights, and traffic signs are cylindrical and have a height similar to a human, the no-normal group could easily make false positive prediction. In contrast, proposed method reduce the number of such mistakes with the addition of normal information. This indicates that the Normal-map is useful in avoiding the false detection of objects similar to human features.

5. Conclusion

We proposed a novel 3D object detection method with Normal-map on point clouds. We have confirmed that the accuracy of BEV map-based object detection is further improved when we introduced normal information to the BEV map. Since the Normal-map can keep high detection accuracy without intensity information, it is possible to use synthesized datasets by simulator with the virtual environment. In the future, to improve the accuracy of the method, we would like to explore deep learning methods for object normal estimation.

References

[1] A. Bochkovskiy, *et al.*, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv, 2020.

[2] J. Beltran, *et al.*, “BirdNet: a 3D Object Detection Framework from Lidar Information”, ITSC, 2018.

[3] M. Simon, *et al.*, “Complexer-YOLO: Real-time 3D Object Detection and Tracking on Semantic Point Clouds”, CVPR, 2019.

Research Achievements

[1] J. Miao, *et al.*, “3D Object Detection with Normal-map on Point Clouds”, VISAPP, 2021.

(Other: 1 conference presentation)