Robust 3D Object Detection for Moving Objects Based on PointPillars

Ryota Nakamura
Kyushu Institute of Technology
Iizuka-shi, Fukuoka, Japan
nakamura.ryota660@mail.kyutech.jp

Shuichi Enokida
enokida@ics.kyutech.jp

Abstract

Deep learning techniques have been applied successfully to detecting objects from video images, and the use of three-dimensional (3D) point clouds obtained from light detection and ranging (LIDAR) with VoxelNet [11] and other techniques have previously been proposed for use in highly accurate object detection methods that are robust against lighting changes. However, while object detection from video images with deep learning has been observed to be continuous and stable, there are times when a few continuous frames suddenly go undetected, thereby resulting in a phenomenon known as a momentary missed detection. Extending the methodology discussed in a previous paper that examined the cause of these momentary missed detection in object detection in 3D point clouds with VoxelNet, this study proposes a robust network for detecting moving objects while considering the cause of similar momentary missed detection in PointPillars, which is an encoder developed based on VoxelNet.

1. Introduction

In recent years, automakers have been active in development of autonomous driving systems and advanced driving assistance systems (ADAS), both of which essentially require high-accuracy recognition of the surrounding environment. Within this context, various object detection techniques are being studied. Some of the onboard devices used to recognize vehicle surroundings include cameras, millimeter-wave radar systems, and light detection and ranging (LiDAR) sensors.

Camera-based sensors are capable of acquiring detailed data on the vehicle surroundings but are susceptible to issues such as low light at night or backlighting glare. In contrast, LiDAR sensors are robust against lighting changes, thus allowing for accurate recognition of the vehicle surroundings even at night. Such sensors emit laser beams and measure the elapsed time and intensity of the reflected beam that return to the sensor in order to acquire the shape of surrounding objects as three-dimensional (3D) point cloud.

In recent years, LiDAR sensors have garnered significant attention due to their excellent ranging performance and ability to accurately measure the distance to a target object, and a number of prior studies have proposed various object detection technologies using LiDAR sensors for purposes such as for vehicle recognition [5, 9, 11], pedestrian recognition [4], and curb recognition [3].

Some typical image-based techniques for object detection that are capable of detecting objects with high accuracy include Faster Region-based Convolutional Neural Networks (Faster-R-CNN) [8], You only Look Once (YOLO) [2], and Single-Shot MultiBox Detector (SSD) [6]. However, while image-based object detection methods can detect most objects accurately, a few frames will occasionally go undetected over time when continuously processing video frames.

While only momentary, such missed detection can delay braking operations and ultimately lead to serious accidents when operating these sensors in conditions such as when driving at high speed on highways or when encountering oncoming vehicles on local roads.

This phenomenon, called momentary missed detection, occurs occasionally when objects are detectable, and the sensor is detecting the objects most of the time. Thus, it is not a limitation of detector performance, which means that clarifying and resolving its causes should allow us to improve overall detector accuracy, thereby producing a safer system.

With that point in mind, this paper proposes a network that takes into account the relative position between a LiDAR sensor and a target object, which is a previously published factor found to cause momentary missed detection in 3D point clouds [7].

2. Robustness of moving object detection

Ideally, autonomous driving systems and ADASs should be able to continuously and correctly detect objects whether or not the sensing vehicle or the vehicles being sensed are in motion. Therefore, moving object detection robustness

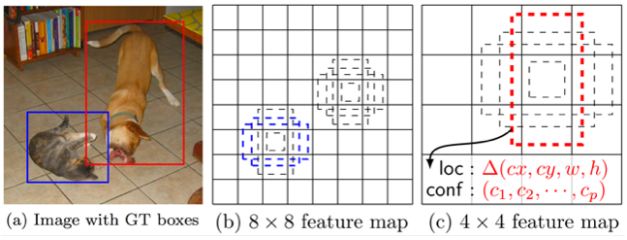

Figure 1. Illustration of SSD feature map (From Ref. [6])

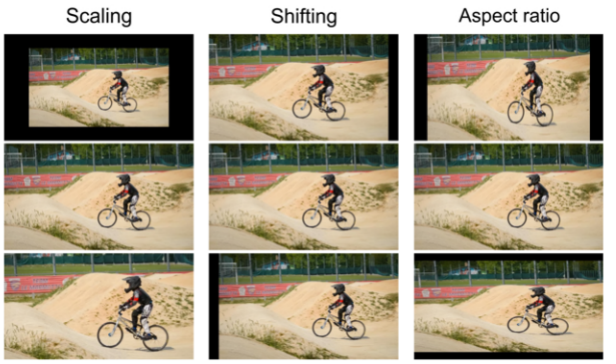

Figure 2. Examples of the transformed images by scaling, horizontal shift, and aspect ratio (from left to right).

refers to whether the target object is detected continuously and correctly in continuous time series data. However, the focus of this paper is on the abovementioned undetected frames that occur in in continuous time series data, even when the object is detected in most frames.

2.1. Prior Studies on Momentary Missed Detection in Video by Frame

This section explains Ref. [10], which examines momentary missed detection in video frames using SSD [6], which is an effective technique for image-based object detection.

2.1.1 Single Shot Multibox Detector

As shown in Fig.1, SSD is an object detection technique that generates a feature map of the input image and plots a map grid to estimate the area populated by the candidate object(s). More specifically, the input images are convoluted and downsampled to generate multiple feature maps with different numbers of grid segments. For the grid of each generated feature map, default boxes are set for several aspect ratios in order to estimate the candidate segments of the target object from the loss factor and Jaccard index. Creating multiple feature maps with different numbers of grid segments makes it possible to detect large objects from feature maps with coarser grids, as shown in Fig.1, or to detect smaller objects from feature maps with finer grids.

Since SSD uses a finite number of anchor boxes, as

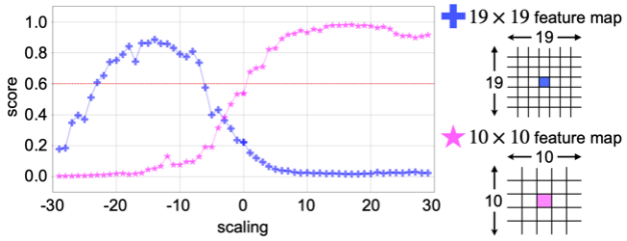

Figure 3. Detection scores for scaling changes (From Ref. [10]).

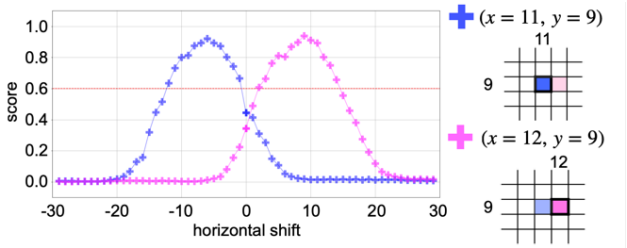

Figure 4. Detection scores for position changes (From Ref. [10]).

shown in Fig.1, the adjacent default boxes will change depending on the size, position, and aspect ratio of the detected object. With that point in mind, Ref. [10] examines the hypothesis that frames will occasionally go undetected when transitioning between two anchors of differing sizes, positions, or aspect ratios for the default boxes.

2.1.2 Technique of Momentary Missed Detection Assessment

In Ref. [10], momentary missed detection is defined as series of frames in which an object that has been detected correctly over time in most frames is undetected for a short period. Such missing frames are evaluated as fitting the following conditions:

$$p^c_{t-1} \geq \gamma_{min} \quad and \quad p^c_{t+1} \geq \gamma_{min} \tag{1}$$

$$\frac{p^c_t}{p^c_{t-1}} \geq \gamma_{ratio} \tag{2}$$

Where p^c_t is the score of the object of class c that is given by our detector for frame t, γ_{ratio} is the permissible deviation in detection probability compared to the previous frame, and γ_{min} is the probability threshold in determining whether the object can be detected.

2.1.3 Scaling Change Assessment

In Ref. [10], the momentary missed detection is determined by setting $\gamma_{min} = 0.5$ and $\gamma_{ratio} = 0.9$ in the equation defined in Section 2.1.2.

As mentioned in Section 2.1.1, Ref. [10] examines the hypothesis that small number of frames go undetected at

the boundaries between two anchors of differing scale, position, or aspect ratio. Accordingly, in order to examine anchor transitions in the scale direction, input images from the missing frame are scaled and detection is performed on each image to check for a transition between the anchors used.

The result of this assessment is shown in Fig.3, where the vertical axis represents the detection scores for each anchor, and the horizontal axis represents the degree of image scaling. The crosses and stars show the transition of the scores of two neighboring anchors when the image is scaled down or up. The cross shows the anchors in the 19*19 feature map, and the star shows the anchors in the 10*10 feature map.

This figure shows that when the original image is reduced, the anchor detection score increases for the grid ' s smaller feature maps. In contrast, when the original image is enlarged, the anchor detection score increases for the grid ' s next largest feature map. At all boundaries for switching anchors, the anchor outputs are below the threshold, which indicates that momentary missed detection has occurred due to a scaling change in the corresponding anchors.

2.1.4 Position Change Assessment

As in Section 2.1.3, in order to examine anchor transitions for position change in one or more directions, the input images are moved, and detection is performed on each image to check for a transition between the anchors used. The result of this assessment is shown in Fig.4, where it can be seen that, as with scaling changes in the input image, the detection score falls below the threshold required to switch anchors. This indicates that a position change in the corresponding anchors could trigger momentary missed detection. Changes in image aspect ratios were similarly examined, even though Ref. [10] states that aspect ratio change do not trigger momentary missed detection.

Based on these validations, Ref. [10] lists target object position changes on the grid and target object size changes relative to the image as factors triggering momentary missed detection in SSD.

2.2. Prior Studies on Momentary Missed Detection in 3D Point Clouds

In this section, in light of the results shown in Ref. [10], explains the results of the author's examined of momentary non detection in object detection from 3D point clouds.

As the technique used for detecting objects identified from 3D point clouds, Ref. [7] examines VoxelNet [11], which was selected for its similarities with SSD, as addressed in Ref. [10]. Just as SSD plots a grid on the feature map and extracts the features, VoxelNet extracts feature after splitting the input point cloud into voxels.

Additionally, as in SSD, the authors considered the possibility that missed detection may also occur in VoxelNet when the target object approaches a voxel boundary. Finally, other techniques that split the 3D input point cloud into voxels or pillars, such as SECOND [9] and PointPillars [5] have been proposed in recent years. Thus, it was considered likely that clarifying the factors related to missed detection in VoxelNet may be useful in developing future techniques.

2.2.1 Technique for Momentary Missed Detection Assessment

In Ref. [7], in an effort to exclude missed detection resulting from VoxelNet performance limits, momentary missed detection were assessed as follows for all frames in which an object was detected correctly in three frames before and after the assessed frames:

$$p_t < f_\tau \tag{3}$$

$$1 - \frac{p_t}{p_{t-1}} \geq df_\tau \tag{4}$$

As in 2.1.2, f_τ is the threshold for determining whether the object can be detected, and df_τ is the permissible deviation in the detection score compared to the previous frame. Frames meeting the above criteria were deemed to be missing.

2.2.2 Discussion of Position and Scaling Changes

In Ref. [10], the two factors triggering momentary missed detection was defined as being transitions between the anchors used from either position changes due to detected object movement or scaling changes. However, due to 3D point cloud features, the size of the object in the point cloud is independent of the object's position relative to the LiDAR sensor. Thus, in Ref. [7], the detected object's position relative to the voxel was examined.

2.2.3 Verification Testing for Momentary Missed Detection

Using VoxelNet, continuous time series data from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [1] was tested for missing frames. Although they were few in number, 21 missing frames were confirmed in testing. Figure 5 shows the mean detection scores of the three frames before and after the missing frames. In this figure, it can be seen that even though the object is detected correctly in the preceding and following

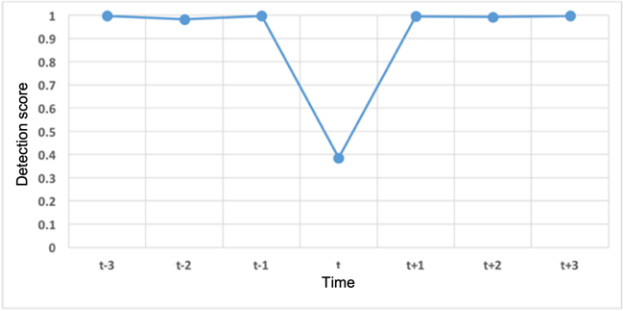

Figure 5. Mean detection scores of the three frames before and after missing frames.

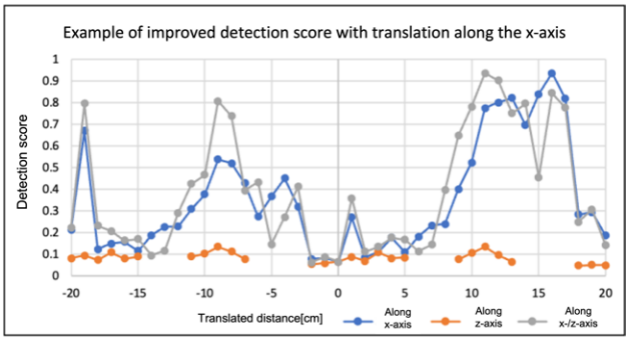

Figure 6. Example of improved score with translation along the x-axis.

frames, one frame, in particular, has a significantly lower detection score.

2.2.4 Position Change Assessment

The reference points of the missing frame was translated along the plane in the x- and z- axis, and in height along the y-axis. Since vehicles being detected will generally move along the surface plane, the object was first moved in three patterns: planar translations along the x-axis, planar translations along the z-axis, and planar translations diagonally along the x- and z- axis simultaneously.

The changes in the detection score for the translated objects was examined. Give the voxel size in the planar direction of 20 cm, the reference points for segmentation was translated from the default position in 1 cm increments from -20 to +20 cm along each direction.

As shown in Figs. 6 and 7, planar translation of the reference points for segmentation improved detection scores in 13 of the 21 scenes.Next, for the eight scenes that did not show improved detection scores, the reference points for segmentation vertically (in the y-axis direction) to check for detection score changes.

Given the voxel height of 40 cm in the y-axis direction, the reference points for segmentation in 2 cm increments from -40 to +40 cm along the y-axis.As a result, the detection score improved in seven out of the eight remaining

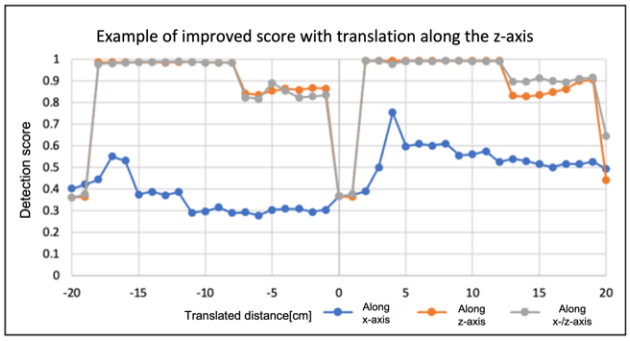

Figure 7. Example of improved detection score with translation along the z-axis.

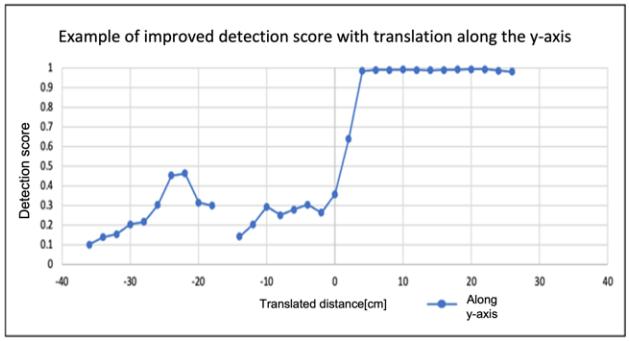

Figure 8. Example of improved detection score with translation along the y-axis.

scenes, as shown in Fig. 8. Based on these results, we confirmed that the cause of momentary missed detection in VoxelNet was associated to the position of the target object in relation to the voxel.

3. Proposed Technique

This section reports on our proposed technique for detecting objects in a 3D point cloud, which extends the technique discussed in Ref. [7]. The proposed technique accounts for what causes missed detection when the 3D point cloud is segmented. More specifically, we examine PointPillars, which is based on VoxelNet, the technique examined in Ref. [7].

3.1. About PointPillars

PointPillars was designed based on VoxelNet. However, while features in VoxelNet are extracted per voxel (a cuboid unit), features in PointPillars are extracted per pillar and then reduced in weight using two-dimensional (2D) convolution. Because it provides a good balance between prediction accuracy and processing speed, PointPillars will likely see future use as a technique for detecting objects in 3D point clouds. With that point in mind, and given the cause of momentary missed detection as described in Section 2, we performed a similar examination of PointPillars to de-

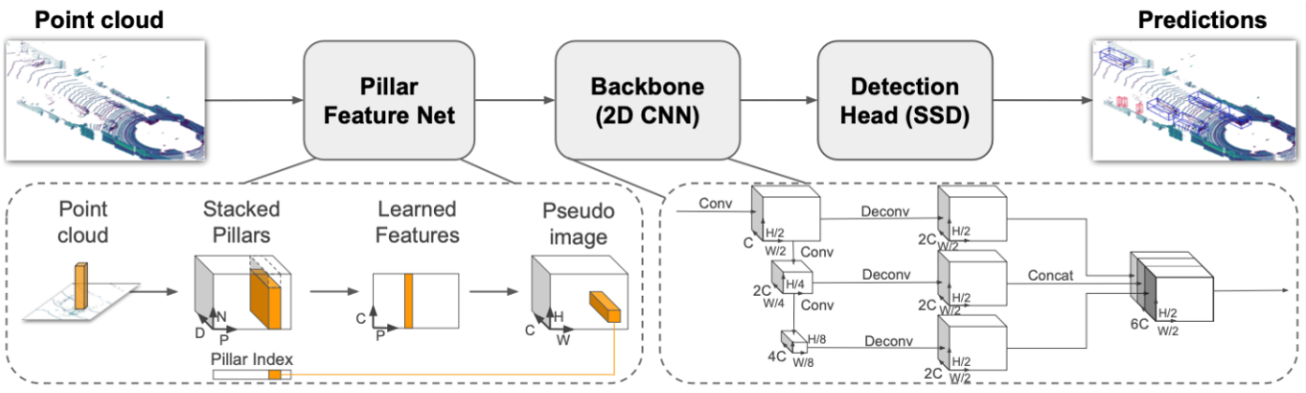

Figure 9. PointPillars network structure (from Ref. [5]).

Table 1. Test parameters.

	Pillars Size	Max Number of Points Per Pillar	Number of feature dimensions of the Backbone				
			Feature1	Feature2	Feature3	Feature4-6	Feature7
Orig.	0.25m*0.25m	100	64	128	256	128	384
Ours1	0.25m*0.25m	100	256(64*4)	512	1024	512	1536
Ours2	0.25m*0.25m	100	256(64*4)	128	256	128	384

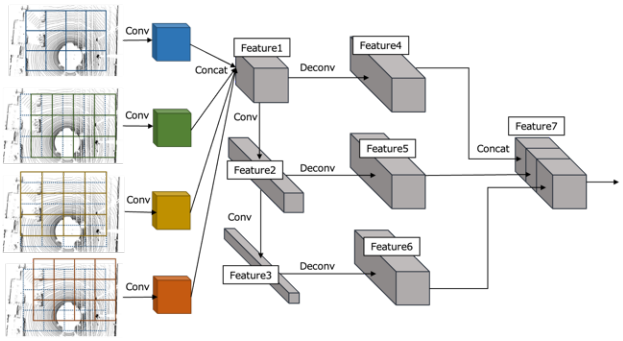

Figure 10. Proposed network.(Blue:Features extracted from default segmentation, Green:Features extracted from segmentation with the reference points changed by 10cm in the x-axis, Yellow:Features extracted from segmentation with the reference points changed by 10cm in the z-axis, Orange:Features extracted from segmentation with the reference points changed by 10cm in the x- and z-axis simultaneously)

Table 2. Results(AP).

Car	BEV			3D		
	Easy	Mod	Hard	Easy	Mod	Hard
Orig.	89.70	81.41	80.35	81.14	68.04	66.52
Ours1	89.72	86.16	84.46	80.89	72.88	67.79
Ours2	89.83	84.11	80.12	83.04	70.75	66.98
Cyclist	**BEV**			**3D**		
	Easy	Mod	Hard	Easy	Mod	Hard
Orig.	68.09	59.95	52.99	66.10	52.55	50.10
Ours1	77.16	61.08	57.61	74.89	57.18	54.29
Ours2	69.14	56.66	53.97	68.28	55.11	51.64
Pedestrian	**BEV**			**3D**		
	Easy	Mod	Hard	Easy	Mod	Hard
Orig.	42.43	39.53	37.47	32.45	30.49	28.95
Ours1	46.04	45.40	43.80	36.25	35.96	34.34
Ours2	41.47	39.47	37.62	30.11	30.50	29.42

termine if accuracy improvements might be obtained in a likewise manner.

3.2. Proposed Technique for Momentary Missed Detection

From Ref. [7], the reference points for segmentation were changed in order to change detected object position relative to the segment, thereby confirming that a detectable object was present.

In addition, we confirmed that the number of false positives would increase if we were to simply operate networks in parallel while taking point clouds by the reference position for segmentation is moved in the direction of each axis by half the size of the segmented area as input and then combining the detection results. For this reason, the method proposed for this study changes the reference points upstream in the network before incorporating the 3D point

Figure 11. Evaluation results (top: original technique; bottom: ours) (a: car / b: cyclist / c: pedestrian / d: correct detection of a false positive)(blue box:detection result / pink box:ground truth)

cloud features. More specifically, the network structure of the PointPillars Backbone (2D CNN) was changed as shown in Figure 10.

4. Demonstration Test

4.1. Dataset

This examination use the KITTI object detection benchmark dataset [1], which consists of 3D point cloud data retrieved from in-vehicle LiDAR, camera images, and driving data. It also includes labels for the objects detected. The KITTI dataset was originally divided into 7481 training and 7518 test images. For the test, the KITTI training images were divided into two groups of 3712 training images and 3769 test images. The KITTI dataset has three annotations: cars, cyclists, and pedestrians. The data values are categorized as easy, moderate, or hard according to the object size and the level of overlap. All of these images were compared.

4.2. Test Overview

For comparison purposes, the parameters for the original technique were set as shown in Table 1. In Proposed Technique 1, multiple pseudo images were generated by changing the reference position for segments in the Pillar Feature Net, as shown in Figure 10. From that point, the pseudo images are concatenated and input to the Backbone. The segment reference position was shifted a distance of 0.125 m, or half of one side of the 0.25 m $\times$ 0.25 m pillar size. The reference position was shifted in three directions: along the x-axis, along the z-axis, and simultaneously along the x- and z-axes. The other parameters were set as shown in Table 1.

As in Proposed Technique 1, Proposed Technique 2 also concatenates multiple pseudo images with changed segment reference positions and inputs them to the Backbone. From there, the number of feature dimensions was changed as shown in Table 1. The detection accuracy of the original and two proposed techniques was then compared.

4.3. Test Results

The test results are given in Table 2, where it can be seen that the average precision (AP) was confirmed to improve in nearly all of the categories for car, cyclist, and pedestrian. Accuracy was especially improved in Proposed Technique 1 for cyclists, pedestrians, and other small objects.

As seen in Fig. 11(a-c), it was also confirmed that the proposed technique could detect objects that went unde-

tected with the original technique, and that false positives, such as the dumpster detected as a car in Fig. 11(d), were correctly removed. From these results, it was considered to be more effective to change the segment reference position and add surrounding data as features instead of changing a single segment.

However, processing speed was significantly reduced from 66.11 fps with the original technique to 12.10 fps and 16.84 fps for Proposed Techniques 1 and 2, respectively. Although Proposed Technique 2 was further examined to determine if the processing speed reduction could be limited by scaling back the number of features in the Backbone and Detection-Head, the effect was inadequate. That is because the point cloud in this process chain takes the most time to pre-process, so increasing the input point cloud sets seriously slows processing speed. Therefore, it is likely that even scaling back the number of features from Backbone on would not limit the processing speed reduction.

The KITTI dataset was acquired at 10 Hz and can be processed in real-time. However, when considering future LiDAR sensor developments, the current processing speed is expected to be insufficient. Accordingly, it will likely be necessary in the future to propose a network with fewer input point cloud sets in order to limit processing speed reductions.

5. Conclusions

In this paper, we proposed a robust object detection technique for moving objects from 3D point clouds, as obtained from in-vehicle LiDAR, based on the cause of momentary missed detection as identified in Ref. [7]. In our VoxelNet examination, we found that momentary missed detection were being caused when the acquired 3D point cloud was split to extract the features and the object's point cloud was being included in multiple regions due to the target object's relative position to the LiDAR sensor, which prevented the features from being extracted correctly.

Therefore, this paper proposed an object detection technique in PointPillars which was developed based on VoxelNet, in which we set multiple segment reference positions for the input point cloud and extract features from each position. From the results obtained, we confirmed that our proposed technique improved detection accuracy. We also found that segmenting the 3D point cloud and then setting multiple segment reference positions for extracting the features may help facilitate accurate recognition of smaller objects.

However, since the proposed technique significantly decreases processing speed compared to the original technique, it will be necessary to implement a network with fewer input point cloud sets in the future in order to limit processing speed reductions.

In our future work, we will quantitatively test whether the momentary missed detection found in Ref. [7] also occur in PointPillars, and whether or not the proposed technique can limit those momentary missed detection.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP18H01463.

References

[1] Geiger Andreas, Lenz Philip, Stiller Christoph, and Urtasun Raquel. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013.

[2] Redmon J., Divvala S., Girshick R., and Farhadi A. You only look once: Unified, real-time object detection. *Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[3] Tan Jun, Li Jian, An Xiangjing, and He Hangen. Robust curb detection with fusion of 3d-lidar and camera data. *International Conference on Information and Communication Technology Robotics*, 2016.

[4] Kidono K, Miyasaka T, Watanabe A, Naito T, and Miura J. Pedestrian recognition using high-definition lidar. *Proc. 2011 Intelligent Vehicles Symposium*, pages 405–410, 2011.

[5] Alex H. Lang, Vora Sourabh, Caesar Holger, Zhou Lubing, Yang Jiong, and Beijbom Oscar. Pointpillars: Fast encoders for object detection from point clouds. *Computer Vision and Pattern Recognition*, pages 12689–12697, 2019.

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Berg Alexander C. Ssd: Single shot multibox detector. *European Conference on Computer Vision*, pages 21–37, 2016.

[7] Nakamura Ryota and Enokida Shuichi. Robust 3d object detection method to various positions by utilizing cnn based on voxelnet. *Vision Engineering Workshop*, 2020.

[8] Ren Shaoqing, He Kaiming, Girshick Ross, and Sun Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems 28*, pages 91–99, 2015.

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

[10] Hosoya Yusuke, Suganuma M., and Okatani Takayuki. Analysis and a solution of momentarily missed detection for anchor-based object detectors. *Winter Conference on Applications of Computer Vision*, 2020.

[11] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.