ESSDet: Enhancing Spatial Shape for 3-D Object Detection

Hiep Anh Hoang
Department of Information Communication Convergence
Soongsil University
Seoul, Korea Republic of.
hiepbk97@soongsil.ac.kr

Myungsik Yoo
School of Electronic Engineering
Soongsil University
Seoul, Korea Republic of.
myoo@ssu.ac.kr

***Abstract*—Recently, point cloud 3-D object detection has attracted a lot of interest and developed into a vital area of research for the 3-D computer vision community, including robotics and autonomous driving. The capacity to perceive 3-D space is essential for self-driving cars to understand the vehicle's surroundings, enabling subsequent systems to respond appropriately. However, 3-D spatial objects frequently exhibit shape imperfections attributed to diverse outdoor factors. To overcome this problem, we present ESSDet, a novel LiDAR-based 3-D object detection model, which enhances the perception of object shape in 3-D space. ESSDet first reproduces the 3-D point cloud geometry of the occluded object by using a deep learning network. Subsequently, our model is trained to estimate the spatial occupancy, signifying whether a given region encompasses object shapes.**

***Index Terms*—Point cloud, LiDAR, 3-D Object Detection, Occluded Object, Autonomous Driving**

I. Introduction

The advent of self-driving cars has ushered in a transformative era in the automotive industry, with an emphasis on developing robust perception systems that enable vehicles to navigate and interact with the surrounding environment autonomously. Among the array of sensors employed in autonomous vehicles, Lidar (Light Detection and Ranging) has emerged as a pivotal technology for achieving precise and comprehensive three-dimensional (3-D) object detection.

Lidar sensors provide an unparalleled capability to capture high-resolution point cloud data, offering a detailed representation of the vehicle's surroundings. This wealth of information becomes instrumental in deciphering the complex urban environment, identifying objects, and ensuring safe navigation. In this context, 3-D perception plays a crucial role by enabling the vehicle to recognize and understand the spatial attributes of encircling vehicles. From there, it is possible to discern objects in a three-dimensional environment, such as the shape of each object or the distance from the object to the ego vehicle.

In this research, we introduce ESSDet, a two-stage approach that is resilient to point clouds with uneven density and achieves precise 3-D object detection. Concretely, our model has the capability to predict regions in space containing obscured objects; in other words, it can estimate the missing spatial regions of the occluded objects at the voxel level. Furthermore, our ESSDet method surpasses both state-of-the-art models [1] and [2] with impressive performance.

II. Methodology

A. Overall Architecture

Our model's flow is partitioned into two parts: The first part comprises Shape Enhancement and voxel Segmentation, while the second part includes 3-D Sparse Backbone, Region Proposal Network (RPN), and Proposal Refinement. Fig. 1 illustrates the comprehensive structure of our ESSDet method. Firstly, we use the Shape Enhancement Module to recover the target object from the raw point cloud. After that, the target object is then placed into the corresponding boxes, combines the raw input, and voxelized to generate the corresponding feature pairs. These feature pairs are fed into the Shape Segmentation Network Θ, a supervised deep learning network, for object segmentation and reconstruction at the voxel level. A sparse 3-D convolutional network Υ then takes a 3-D point cloud as input, and its layers are concatenated with the sparse probability tensor of the Shape Segmentation Network to extract spatial features. Following that, we design a RPN, that utilizes the output features of Υ and Θ to generate proposal boxes. Each proposal comprises the bounding box's center location represented as (x_p, y_p, z_p), with the 3D box's dimensions, heading, and confidence specified by (l_p, w_p, h_p), θ_p and p_p, respectively. The local geometric features consist of Θ, and the multi-scale features from Υ are condensed using RoI grid pooling to capture the region's object shape interest. We integrate the grid features and pool the local geometric features to produce the final precise bounding box predictions.

B. 3-D Object Occupancy Estimation

1) Shape Enhancement Module: The key idea of our Shape Enhancement Module is to predict the entire shapes and employ them as labels for the subsequent network Θ. We leverage the 3-D bounding box information from the annotations to extract points associated with the incomplete object. The set of a partial point cloud of an incomplete object is denoted as $\mathcal{F} = \{\mathcal{F}_i, i = 1, ..., N\}$, where N represents the total number of points, and $\mathcal{F}_i$ is a vector with (x, y, z) coordinates. Initially, we employ Region of Interest pooling [3] based on the bounding box region to obtain the output to obtain the

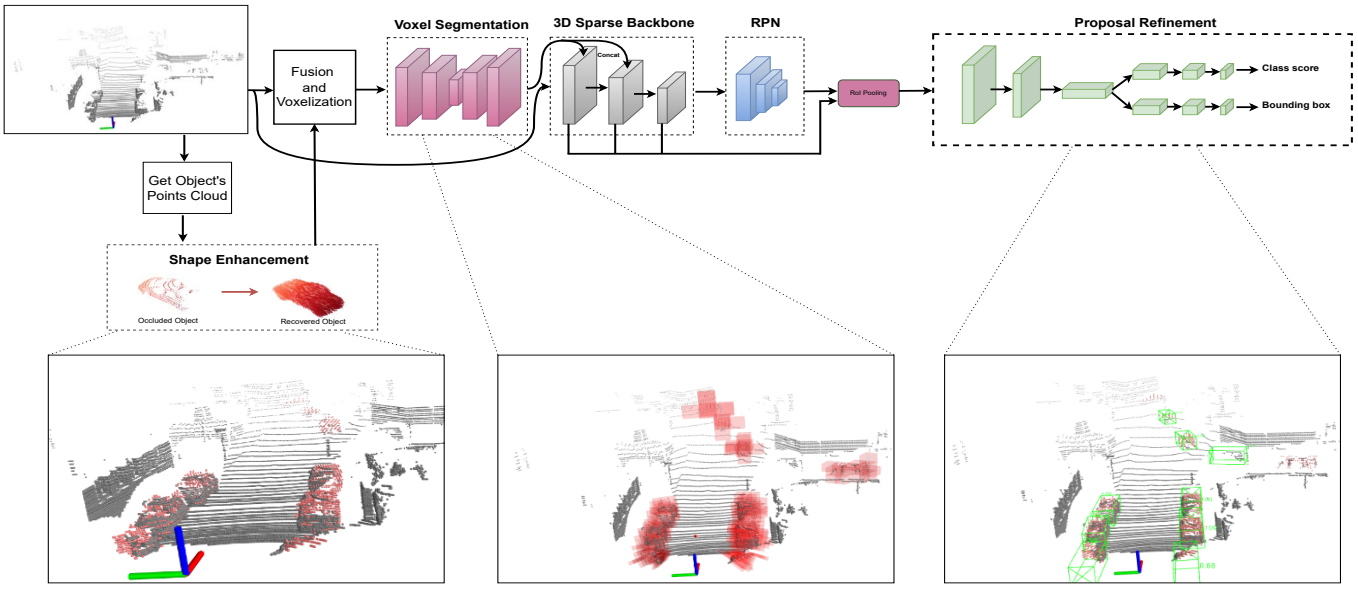

Fig. 1. The architecture of our proposed ESSDet model. First, from the raw input point cloud, Shape Enhancement Module takes the partial point cloud set of object to generate the training target. Next, the Shape Segmentation Network leverages this target for segmenting and estimating the occupancy of the complete shape. Following this, the 3-D Sparse Convolution extracts features from the point cloud, which are then concatenated with the feature maps from the Shape Segmentation Network. Subsequently, the Region Proposal Network (RPN) processes the combined output to generate 3-D proposals. The final predictions, including bounding boxes and confidence scores, are obtained by pooling each proposal and passing them through the Proposal Refinement Network.

incomplete shape of the object, expressed as $\mathcal{F}$ with the grid pooling size is $(s_p \times s_p \times s_p \times 3)$. For simplicity, we take advantage of PointTr [4] as a module to predict the missing point cloud, and follow the encoder and decoder process:

$$\mathcal{V} = \mathbf{E}(\mathcal{F}), \quad \mathcal{H} = \mathbf{D}(\mathcal{Q}, \mathcal{V}), \tag{1}$$

where $\mathbf{E}$ and $\mathbf{D}$ represent the encoder and decoder functions. $\mathcal{F} = \{\mathcal{V}_i, i = 1, ..., N\}$, $\mathcal{Q} = \{\mathcal{Q}_i, i = 1, ..., M\}$, $\mathcal{H} = \{\mathcal{H}_i, i = 1, ..., M\}$, and $\mathbf{M}$ are the output features of the encoder, the queries of the decoder, the completion point cloud, and the total number of completion point cloud, respectively.

2) Voxel Shape Segmentation: The output feature maps of the Shape Enhancement Module are fused with the raw input into a pair of features. We then voxelize these pair features and feed into Shape Segmentation Network. Denote $\bar{S}$, $\mathcal{O}_{\bar{S}}$ are complete shape of foreground object and the foreground object occupancy, respectively. We set $\mathcal{O}_{\bar{S}}$ = 1 for the voxels that include $\bar{S}$, and $\mathcal{O}_{\bar{S}}$ = 0 for the others. The Shape Segmentation Network is trained to estimate the probability $\mathcal{P}(\mathcal{O}_{\bar{S}})$ for voxels containing points of $\bar{S}$. The network comprises two downsampling sparse-convolution layers and two upsampling inverse convolution layers. Additionally, each layer incorporates multiple submanifold sparse-convolutions [5]. The dimensions of the output features for these layers are 16, 32, 64, 32, and 32, respectively.

C. 3-D Spatial Object Detector

A sparse 3-D convolutional backbone Υ of the detection feature extraction network follows BtcDet [6]. The point cloud is transformed into Cartesian voxels, we utilize the mean voxel method to derive the representation of each occupied voxel based on their x, y, z coordinates. We take two channels from $\mathcal{P}(\mathcal{O}_{\bar{S}})$ to concatenate with layers of the backbone network Υ. Additionally, layers Υ of are concatenated with the sparse probability tensor of the foreground object occupancy. We proceed to remove the height dimension from the features to obtain bird's-eye view 2D features. The following Region Proposal Networks [7] are then used to propagate the features and output residues of two anchors on the output feature maps. Subsequently, we apply a RoI pooling to consolidate features from the proposal. Finally, the proposal refinement module infers a class confidence score associated with IoU and the disparities between the 3-D proposal boxes and the actual ground truth bounding boxes. The proposal refinement module consists of two branches, one dedicated to class confidence scores and the other to box regression. The 3-D IoU confidence of each RoI is computed as follows:

$$y_g = \begin{cases} 1, & \text{if } IoU > 0.75, \\ 2.IoU - 0.5, & \text{if } 0.25 < IoU \leq 0.75, \\ 0, & \text{if } IoU \leq 0.25, \end{cases} \tag{2}$$

To refine bounding boxes, we employ the widely used box encoding equation:

$$x_r = \frac{x_g - x_p}{d_p}, y_r = \frac{y_g - y_p}{y_p}, z_r = \frac{z_g - z_p}{z_p} \quad with\ d_p = \sqrt{l_p^2 + w_p^2} \tag{3}$$

TABLE I
COMPARISON RESULT ON KITTI DATASET FOR 3-D OBJECT DETECTION UNDER 40 RECALL THRESHOLDS, EVALUATED BY THE 3-D AVERAGE PRECISION (AP)

Model	Reference	Modality	Car Easy	Car Moderate	Car Hard
PointPillars	CVPR 2019	Lidar	87.75	78.39	75.18
Second	Sensor 2018	Lidar	90.97	79.94	77.09
SA-SSD	CVPR 2020	Lidar	92.23	84.30	81.36
PV-RCNN	CVPR 2020	Lidar	92.57	84.83	82.69
Voxel R-CNN	AAAI 2021	Lidar	92.38	85.29	82.86
Our	-	Lidar	**93.15**	**86.38**	**83.99**

$$w_r = log(\frac{w_g}{w_p}), l_r = log(\frac{l_g}{l_p}), h_r = log(\frac{h_g}{h_p}) \\ \theta_r = \theta_g - \theta_p, \tag{4}$$

where $\{x, y, z\}, \{w, l, h\}$ represents the coordinates of the bounding box center, the 3-D dimension of the bounding box, respectively. θ denotes the rotation angle around the z-axis of the boxes. The subscripts r, p, g indicate residue, 3-D proposal, and ground truth, respectively.

D. Loss Functions

Our total loss is the multitask loss including shape enhancement loss, voxel shape segmentation loss, bounding box regression loss, and label classification loss.

For the shape enhancement loss, we use Chamfer Distance (CD):

$$\mathcal{L}_{prs} = \frac{1}{|P1|} \sum_{p_1 \in P_1} \min_{p_2 \in P_2} ||p_1 - p_2|| + \frac{1}{|P2|} \sum_{p_2 \in P_2} \min_{p_1 \in P_1} ||p_2 - p_1|| \tag{5}$$

The voxel segmentation loss is the sigmoid cross-entropy Focal Loss:

$$\mathcal{L}_{focal}(p_v) = -(1 - p_v)^{\lambda} log(p_v), \tag{6}$$

where $p_v = \begin{cases} \mathcal{P}(\mathcal{O}_s), & \text{if } \mathcal{P}(\mathcal{O}_{\bar{s}}) = 1 \text{at voxel } v \\ 1 - \mathcal{P}(\mathcal{O}_s), & \text{otherwise,} \end{cases}$

We follow [7] to apply the Smooth-L1 loss function L_{reg} to regress the target box:

$$\mathcal{L}_{reg} = SmoothL1(sin(\theta_{\hat{y}} - \theta_y)) \tag{7}$$

where $\theta_{\hat{y}}, \theta_y$ denote the angle of the predicted bounding box and the ground truth, respectively.

For the classification task, the focal loss is applied to address the issue of imbalanced classes:

$$\mathcal{L}_{cls} = -\alpha(1 - y_c)^{\gamma} \log(y_c) \tag{8}$$

where y_c denote the probability of classes and α and γ are the parameters of the classification loss.

The final loss function is synthesized from the loss functions above:

$$\mathcal{L}_{total} = \mathcal{L}_{prs} + \mathcal{L}_{focal}(p_v) + \mathcal{L}_{reg} + \mathcal{L}_{cls} \tag{9}$$

III. EXPERIMENT AND RESULT

A. Experimental Setup

1) Dataset: In our evaluation of the ESSDet framework, we utilize the extensively employed KITTI dataset [8], a widely used dataset in the realm of 3-D object detection. This dataset consists of a total of 14,999 LiDAR frames, with 7,481 frames designated for training and 7,518 frames for testing purposes. To facilitate model training, the training data is further divided into two subsets: a train subset comprising 3,712 frames and a val subset consisting of 3,769 frames. To assess the model's performance on the KITTI validation set, ESSDet is trained using 80% of the combined train and val data, while the remaining 20% is reserved for validation purposes.

2) Evaluation Metrics: To assess the model's performance, we conduct experiments on the primary object of interest in the KITTI dataset, the car category. These experiments encompass various levels of difficulty, as outlined in Table I. We compare our approach with methods that rely solely on LiDAR as the input data source. We calculate the average precision (AP) using a recall position of 40, considering a bounding box overlap of 70%.

3) Training Details: In all our experiments, we conduct model training with a batch size of 8 using 2 RTX 3060GPUs. We follow the setting of regular two-stage method [2] as our training strategy. We apply the warm-up mechanism for the first epoch during training. The learning rate, data augmentation configuration, and non-maximum suppression (NMS) threshold adhere to the settings in [1].

B. Results

Our ESSDet has demonstrated high performance, surpassing state-of-the-art models such as Pointpillar [9], SECOND [7], SA-SSD [10], PV-RCNN [1], and Voxel R-CNN [2], as illustrated in Table I. We utilize the KITTI dataset for all methods. Our model has achieved notable performance on the crucial

object category of vehicles across all three difficulty levels. By estimating the object shape's occupancy in 3-D space and predicting the likelihood of the missing parts of the object, our ESSDet model has significantly improved performance.

IV. Conclusion

Using a novel network named ESSDet, we present a new method for 3-D object detection in this study. By combining the approach of estimating object occupancy and predicting missing parts, our model demonstrates the capability to identify regions containing obscured objects in 3-D space, thereby enhancing overall detection performance significantly. As a result, our ESSDet method demonstrates an outstanding performance compared to state-of-the-art approaches.

Acknowledgment

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Goverment of South Korea (MSIT)(NRF-2021R1A2B5B01002559)

References

[1] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.

[2] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1201–1209, 2021.

[3] S. Shi, Z. Wang, J. Shi, X. Wang, and H. 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*, vol. 43, no. 8, pp. 2647–2664, 2020.

[4] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12498–12507, 2021.

[5] B. Graham and L. Van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[6] Q. Xu, Y. Zhong, and U. Neumann, "Behind the curtain: Learning occluded shapes for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 2893–2901, 2022.

[7] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving," in *The KITTI Vision Benchmark Suite, 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

[10] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11873–11882, 2020.