STN: Saliency-Guided Transformer Network for Point-Wise Semantic Segmentation of Urban Scenes

Lingfei Ma, *Member, IEEE*, Jonathan Li, *Senior Member, IEEE*, Haiyan Guan, *Senior Member, IEEE*, Yongtao Yu, *Senior Member, IEEE*, and Yiping Chen, *Senior Member, IEEE*

***Abstract*— Accurate and effective road object semantic segmentation plays a significant role in supporting extensive intelligent transportation system (ITS)-related applications. However, most existing image-based methods and point-based methods cannot deliver promising solutions with respect to segmentation accuracy and robustness, especially in complex urban road scenes. Thus, we design a saliency-guided transformer architecture (STN) in this letter for point-wise semantic segmentation from mobile laser scanning (MLS) point clouds. First, four types of feature saliency maps are constructed to obtain more compact feature spaces for enhancing the feature encoding semantics. Then, integrated with offset attention mechanisms and edge convolutions, an effective point-wise transformer network is proposed to extract high-level features for point-wise label assignment of road objects. The STN model is evaluated on the Pairs-Lille-3D (PL3D) dataset and achieves satisfactory experimental results with 87.2% overall accuracy (OA) and 81.7% mean intersection-over-union (IoU), respectively. Comparative studies with five deep learning-based methods also prove the superior performance of the STN model for large-scale semantic segmentation tasks.**

***Index Terms*— Edge convolution, feature saliency, offset attention, point cloud, semantic segmentation, transformer network.**

I. Introduction

THE point-wise segmentation task aiming to determine the semantic label point-by-point in the entire point clouds is a remarkably essential process to support extensive applications, including intelligent robotics, autonomous vehicles, and digital twins. Compared to the 2-D optical images, the 3-D point clouds could more precisely and frequently monitor the spatial information, orientation, and geometric shape attributes of road objects. Most significantly, they are less sensitive to illumination conditions, shadow influence, and viewpoint variations [1]. Unlike 2-D images with a regular grid structure, 3-D point clouds captured by light detection and ranging (LiDAR) sensors are in an unorganized data format and disordered distributions, making it challenging to achieve efficient and accurate road object semantic segmentation, especially in complex and large-scale urban areas [2].

A potential solution for point-wise road object segmentation is rule-based or thresholding-based methods, which, yet, require ample prior knowledge and have limited performance when dealing with varying test scenarios. More recently, due to the dominant capabilities of extracting representative and multilevel features in an end-to-end manner with few human interventions, deep learning-based approaches are being thoroughly investigated to segment road objects from point cloud data. These deep learning-based approaches commonly adopt two processing tactics, that is, image-based tactic and point-based tactic.

The image-based tactic transforms point clouds into a collection of georeferenced feature images, thus, the well-designed learning-based methods in image processing domains could be performed. Accordingly, in [3], a multiview convolutional neural network (MVCNN) as the pioneer was proposed to investigate a 3-D–2-D dimension reduction strategy for 3-D shape representation. First, multiple views were captured using a view pooling operation without assigning specific orders. Then, such views were fed into CNN models separately and combined via max pooling layers to encode the high-level and inherent features. Furthermore, Qi *et al.* [4] improved the MVCNN model performance by proposing an anisotropic probing kernel for image rendering. Compared with MVCNN, this method introduced multiresolution filtering and sphere rendering algorithms to obtain more spatial information in multiscale, which achieves view-invariant and enhances the model robustness for object shape diversities. Dai and Nießner [5] leveraged the imagery and geometry data for semantic scene segmentation. Feature maps were first extracted from images. Depending on differentiable back projectors, these feature maps were thus projected into the 3-D information. Then, a multiview pooling method was conducted to learn more discriminative features. Meanwhile,

Manuscript received 5 April 2022; revised 7 June 2022; accepted 10 July 2022. Date of publication 14 July 2022; date of current version 20 July 2022. This work was supported by the National Natural Science Foundation of China under Grant 42101451 and Grant 41871380. *(Corresponding author: Lingfei Ma.)*

Lingfei Ma is with the School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 102206, China (e-mail: l53ma@cufe.edu.cn).

Jonathan Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Haiyan Guan is with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: guanhy.nj@nuist.edu.cn).

Yongtao Yu is with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an 223003, China (e-mail: allennessy@hyit.edu.cn).

Yiping Chen is with the School of Geospatial Engineering and Science, Sun Yat-sen University, Zhuhai 519082, China (e-mail: chenyp79@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/LGRS.2022.3190558

Kanezaki *et al.* [6] introduced a CNN-based model that inputs multiview images of 3-D objects and outputs their object categories. Different from preceding studies that take known viewpoint labels for training, this model was trained in an unsupervised way by taking these viewpoint labels as potential variables. To reduce information loss during projection, SnapNet [7] was presented to generate a depth and RGB image pairs using snapshots derived from input point clouds. Each pair of images was accordingly labeled pixel-by-pixel based on fully convolution networks. Likewise, SnapNet-R [8] was developed by directly generating multiple views and capturing dense 3-D point markers for segmentation performance improvement. Additionally, Yang *et al.* [9] tested a multiview semantic learning network (MVSLN) to obtain the representative features for 3-D object detection. In their work, to preserve much more low-level features, four views of an object were first created by projecting 3-D point clouds to planes with specific angles. Then, a spatial recalibration operation was performed for deviation correction due to different projection angles and adjusted the relative feature locations of these four views, followed by a region proposal network for object detection. On the whole, the image-based tactic could deliver high efficiency in processing large scenarios, yet some important geometric spatial information will be lost during the transformation from 3-D spaces to 2-D images, as well as causing redundant information.

On the contrary, the point-based tactic could directly consume point clouds without information loss. PointNet [10], as a pioneering framework for 3-D point-wise classification and segmentation, was designed to extract spatial features of unordered point clouds through multilayer perceptron (MLP) operations and then fused these features by max-pooling layers. In addition, to address the dilemma of transformation invariance, PointNet presented a data-dependent spatial transformer network to canonicalize the raw point clouds before feeding them into MLP layers, so as to greatly improve the experimental results. However, PointNet is less effective to acquire local features of point clouds due to max-pooling operations, resulting in less robust to intricate scenarios and multigrained fashions. To solve this problem, PointNet++ [11] was further constructed to extract more local features using the multiscale sampling and grouping mechanisms from coarse layers to fine layers, followed by PointNet for global high-level feature extraction. Li *et al.* [12] designed a $\mathcal{X}$-transformation operator that could directly convolve kernels on point features, contributing to inherent feature encodings and object shape information preservation. Meanwhile, Wang *et al.* [13] proposed a dynamic graph convolutional neural network (DGCNN) for effective geometric feature encodings in local areas through edge convolutions. These edge convolutions encapsulated in most mainstream network architectures could dynamically adjust the given fixed graph for each layer output, incorporate local information in local regions, and are suitable for multilayer structures to capture global shape properties. Likewise, Ma *et al.* [14] improved the DGCNN performance by designing a multiscale feature extraction scheme, followed by conditional random field postprocessing for 3-D point-wise segmentation refinement. According to the self-attention mechanism and point-wise operations, Zhao *et al.* [15] proposed a Transformer-based self-attention neural network, called Point Transformer, for semantic scene segmentation tasks from indoor 3-D point clouds. These point-based methods can well preserve the spatial and geometric information of road objects, which are more suitable for accurate and robust 3-D object segmentation, particularly for complex test scenes.

In this letter, we introduce a saliency-guided transformer network, called STN, combined with overall accuracy (OA) mechanisms and edge convolutions for road object semantic segmentation from mobile laser scanning (MLS) point clouds. The STN model takes generated feature saliency maps derived from raw MLS point clouds as the input and outputs point-wise road object semantic labels. The main contributions are summarized as follows: 1) four types of feature saliency maps are constructed to obtain more compact feature spaces for improving the feature encoding semantics and 2) an effective point-wise transformer network integrated with offset attention and edge convolutions is introduced to extract high-quality features for point cloud semantic segmentation.

II. Point-Wise Semantic Segmentation

The proposed STN model mainly contains four modules: feature saliency construction, point-wise transformer network, offset attention, and edge convolution. Fig. 1 details the workflow of the STN framework.

A. Feature Saliency Construction

In this letter, we investigate four new features for the unordered point clouds to amplify the distinction of different point features and enhance intraclass compactness. These new features could not only help generate more compact feature spaces, but also strengthen input embedding performance in the proposed STN framework. These four new features are height context feature (HCF), intensity context feature (ICF), density context feature (DCF), and normal context feature (NCF). For each point $p_i(x_i,\ y_i,\ z_i,\ r_i)$, $i = 1, 2, \ldots, n$, where n denotes the number of input points, $s_i(x_i,\ y_i,\ z_i)$ represents the spatial information, and r_i indicates the intensity values. Accordingly, two cotangent functions are employed to calculate both HCF and ICF feature saliency as follows:

$$\begin{cases} \mathrm{HCF}_i = \cot\left(\dfrac{1}{1+e^{-\delta\times z_i}}\right) \\ \mathrm{ICF}_i = \cot\left(\dfrac{1}{1+e^{-\delta\times r_i}}\right) \end{cases} \tag{1}$$

where HCF_i and ICF_i indicate the new salient height features and intensity features of each point p_i, $\cot(\cdot)$ is a cotangent function. δ is a transformation degree, which is predefined as $\delta = 1$ in this letter.

Next, the DCF can be determined by computing the number of adjacent points of each point in a given neighborhood as follows:

$$\mathrm{DCF}_i = \mathcal{N}(R,\ \text{range}\ (0, 1)) \tag{2}$$

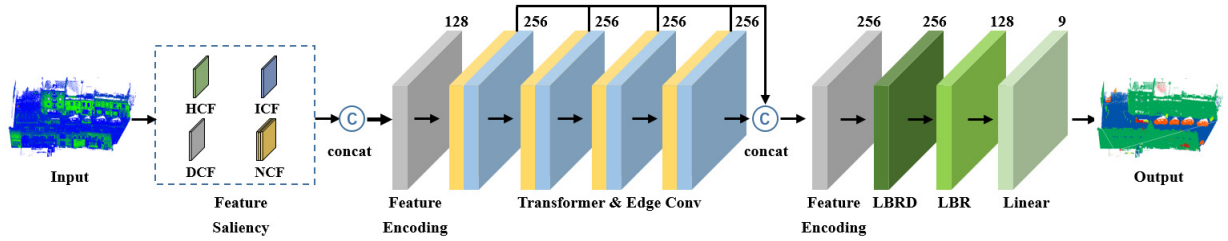

Fig. 1. Workflow of the STN framework. Numbers on the above of different modules present their output channels.

where DCF_i represents the new spatial DCF, R denotes the searching radius of the spherical neighborhood of each point, and $\mathcal{N}$ normalizes the input value to the range [0, 1]. Specifically, $R = 0.5$ m is set in this letter.

Moreover, the normal context feature $\mathrm{NCF}_i(Nx_i, Ny_i, Nz_i)$ of each point is further calculated to enhance the spatial coordinate distinction among different points. Finally, all these new features are concatenated together to generate the new input features as $np_i(Nx_i, Ny_i, Nz_i, \mathrm{HCF}_i, \mathrm{ICF}_i, \mathrm{DCF}_i)$ of each point p_i, then, we can obtain a new output dataset, that is, $\mathcal{M} = \{np_1, n\ p_2, \ldots, n\ p_n\}$, which can be fed into the transformer-based neural network to extract more high-level features.

B. Point Transformer Network

The straightforward way to directly apply Transformer to 3-D point clouds is to consider the whole input point clouds as a sentence, while points are treated as different words. This point-wise transformer network is performed by conducting a feature embedding and constructing attention layers with the self-attention (SA) modules.

Similar to word embedding in natural language processing, feature-based point embedding focuses on grouping points together in different embedding feature spaces if these points are more semantically similar. To this end, the point cloud dataset $\mathcal{M}$ generated in Section II-A is fed into a d_f-dimensional space $\mathcal{F} \in R^{N\times d_f}$ using four cascaded transformer and edge convolution layers, each with a d_f-dimensional output. To make a tradeoff between computational costs and model performance, $d_f = 256$ is set based on prior knowledge [16].

Moreover, self-attention modules, as the key element in transformer-based networks, are employed to calculate semantic affinities between different points. Specifically, assume $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ are the query, key, and value matrices, respectively. These matrices are calculated by a series of linear operations of the input features $\mathcal{F}_{\text{in}} \in R^{N\times d_f}$ using the following equations:

$$(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \mathcal{F}_{\text{in}} \cdot \left(\mathcal{W}_q, \mathcal{W}_k, \mathcal{W}_v\right) \tag{3}$$

where $\mathcal{W}_q$, $\mathcal{W}_k$, and $\mathcal{W}_v$ denote the shared learnable weights, $\mathcal{W}_q, \mathcal{W}_k \in R^{d_f\times d_t}$, $\mathcal{W}_v \in R^{d_f\times d_f}$, $\mathcal{Q}, \mathcal{K} \in R^{N\times d_t}$, $\mathcal{V} \in R^{N\times d_f}$, and d_t denotes the dimensions of the query and key vectors. We set $d_t = (d_f/4)$ in this letter for computational efficiency.

First, depending on both $\mathcal{Q}$ and $\mathcal{K}$ matrices, the attention weights are computed through the matrix dot product, the output weights are then normalized as follows:

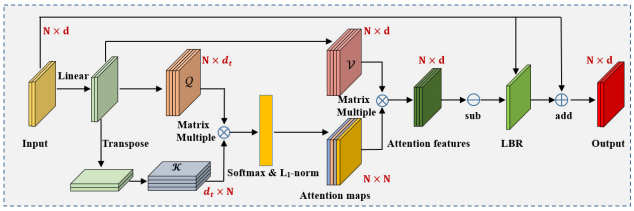

Fig. 2. Offset attention architecture. N indicates dimensions and d indicates feature channels.

$$\tilde{A} = (\tilde{a})_{i,j} = Q \cdot \mathcal{K}^T \tag{4}$$

$$\bar{a}_{i,j} = \mathrm{softmax}\left(\tilde{a}_{i,j}\right) = \frac{\exp\left(\tilde{a}_{i,j}\right)}{\sum_k \exp\left(\tilde{a}_{k,j}\right)} \tag{5}$$

$$a_{i,j} = \frac{(\tilde{a})_{i,j}}{\sum_k \bar{a}_{i,k}} \tag{6}$$

where $A = (a)_{i,j}$ is the attention weight. Furthermore, the intermediate output features $\mathcal{F}_{\text{mid}}$ that are the weighted sum of value vectors, which are determined as follows:

$$\mathcal{F}_{\text{mid}} = A \cdot \mathcal{V}. \tag{7}$$

The $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ matrices are ascertained by taking the both shared linear transformation matrices and the input feature $\mathcal{F}_{\text{in}}$ into consideration, which are therefore invariant to permutations. Additionally, weighted-sum and softmax operations are both order-independent. Herein, the self-attention operation is permutation-invariant, which makes it more suitable for the unorganized and discrete point clouds.

C. Offset Attention

Moreover, as demonstrated in [16], point-wise transformer networks could achieve better segmentation performance, if the SA mechanism is improved by an OA mechanism. Fig. 2 illustrates the architecture of offset attention layers. More specifically, the OA layers compute the differences between the input features $\mathcal{F}_{\text{in}}$ and the OA features $\mathcal{F}_{\text{mid}}$ via element-wise subtraction. These differences are subsequently fed into LBR networks (i.e., linear, batch normalization, and ReLU layers) to calculate the out features $\mathcal{F}_{\text{out}}$ as follows:

$$\mathcal{F}_{\text{out}} = \mathrm{OA}(\mathcal{F}_{\text{in}}) = \mathcal{F}_{\text{in}} + \mathrm{LBR}(\mathcal{F}_{\text{in}} - \mathcal{F}_{\text{mid}}) \tag{8}$$

where $\mathcal{F}_{\text{in}} - \mathcal{F}_{\text{mid}}$ is similar to a discrete Laplacian operator. Consequently, this OA-based transformer network could not only sharpen the attention weights, but also alleviate the problem of noisy points, contributing to the feature extraction during the segmentation tasks.

D. Edge Convolutions

Transformer-based networks are effective for global feature encodings; however, they ignore the local features that are remarkably significant in point cloud segmentation. Herein, we improve the EdgeConv operations developed in [13] to improve the point embedding to strengthen the local feature extraction capability of the STN model. More specifically, the KNN algorithm is utilized to determine the k-nearest neighbors of a given point. The number of neighbors can dynamically adjust between adjacent layers and accordingly compute the sequence of edge feature embedding. To this end, we construct a graph $g = (\upsilon, e)$ in d-dimensional space, and υ and e indicate vertices and edges, respectively. Let edge features be $e_{ij} = h_\psi(Nx_i, N\ x_j)$, where $h_\psi \in R^{d\times d}$ indicates a nonlinear transformation. Accordingly, the improved EdgeConv operator is proposed by performing a channel-widen symmetric aggregation function on edge features. Hence, the output at the ith vertex is calculated using the following equations:

$$Nx_i' = \max_{j:(i,j)\in e} h_\psi\left(Nx_i, N\ x_j\right) \tag{9}$$

$$h_\psi\left(Nx_i, N\ x_j\right) = h_\psi\left(Nx_i, \left(Nx_j + Nx_i\right)/2\right) \tag{10}$$

where $\max_{j:(i,j)\in e}$ indicates a max operation as the symmetric aggregation function. Finally, all global features learned from transform-based networks and local features learned by EdgeConv operators are concatenated together and put into LBRD (LBR & dropout) and LBR layers to output segmentation results.

III. Experiment and Result Analysis

A. Data Descriptions

In this letter, the Paris-Lille-3D (PL3D) dataset [17] was used to evaluate the STN model performance. This dataset has a total length of 1.9 km with over 143 million points. Moreover, nine different object classes, for example, Natural, Cars, Pedestrian, Barrier, Trash Can, Bollard, Poles, Building, and Ground, are manually labeled point-by-point for the semantic segmentation task. The PL3D dataset was collected from complex urban road scenarios, which typically represent real-world road conditions with many moving obstacles, various point densities, and occlusions, hence leading to significant challenges for accurate point-wise semantic segmentation. Specifically, the whole point cloud dataset is separated into 70% and 30% data subsets for training and testing, respectively. We proposed and evaluated the STN model using TensorFlow 2.3.0, Python 3.6.9, and Nvidia RTX 3090 graphics card with 24-GB memory on the Ubuntu 20.04 LTS operating system. Based on prior knowledge and extensive experiments, we predefined the initial learning rate, dropout rate, batch size, and the number of iterations as 0.0001, 0.5, 32, and 200 in the training stage, respectively.

B. Experimental Results

To provide an accurate assessment of the point-wise semantic segmentation, two evaluation metrics, that is, intersection-over-union (IoU) and OA, were employed to calculate the model performance. Table I indicates the point cloud semantic segmentation results obtained on the PL3D dataset, by estimating the OA and IoU across all the object classes. To visually inspect the experimental results, Fig. 3 illustrates the road object semantic segmentation in complex urban road scenes. Nine different colors denote different types of road objects.

TABLE I
POINT-WISE SEMANTIC SEGMENTATION RESULTS ON THE PL3D DATASET

Object Category	OA (%)	IoU (%)
Natural	93.4	91.8
Car	96.5	96.0
Pedestrian	67.1	59.5
Barrier	80.5	70.1
Trash Can	73.4	61.9
Bollard	89.3	83.4
Pole	85.6	75.3
Building	99.1	97.8
Ground	99.6	99.5
Average	87.2	81.7

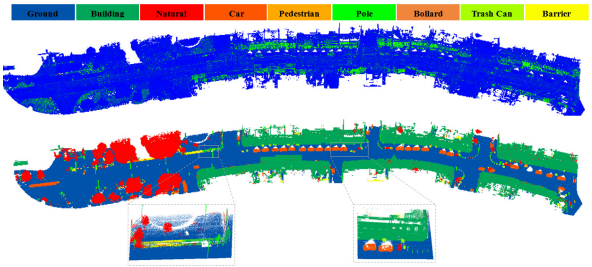

Fig. 3. Point-wise segmentation results in this letter. (Top row) Raw input point clouds. (Middle row) Experimental results by using the STN model. (Bottom row) Two zoomed-in views.

As shown in Table I, the proposed STN model could achieve 87.2% OA and 81.7% mean IoU on the PL3D dataset, respectively. Apparently, the STN model delivered superior accuracies in extracting the road object types of Ground and Building. On the contrary, a relatively low segmentation accuracy was obtained on the road object type of Pedestrian. Moreover, similar segmentation accuracies were achieved on the road object types of Pole, Bollard, Trash Can, and Barrier. However, according to the magnified views in Fig. 3, it is seen that some points belonging to barriers were misidentified as natural points, while some natural points were misidentified as pedestrians. The reasons leading to different segmentation accuracies are: the complexity of road scenes and varying object structures have considerable influences on the feature encoding ability of the STN model. Moreover, the moving road users (e.g., cyclists), distortion, and background interference in the PL3D dataset could also cause incorrect point-wise segmentation results. To sum up, benefiting from the design of several feature saliency maps and the point-wise transformer network with the assistance of OA mechanisms and edge convolutions for feature learning, the STN model could behave effectively and promisingly on point-wise semantic segmentation of MLS point clouds.

C. Comparative Study

In order to further demonstrate the superior performance of the proposed STN model for point-wise semantic

TABLE II
SEGMENTATION RESULTS USING DIFFERENT POINT-WISE FRAMEWORKS

Methods	**OA** (%)	**mIoU** (%)
PointNet	78.6	38.6
PointNet++	81.0	32.0
PointCONV	85.6	60.5
DGCNN	84.1	52.9
Point Transformer	86.5	78.7
Ours	**87.2**	**81.7**

segmentation, a comparative study was performed to compare the STN model with several point-wise segmentation methods, including PointNet [10], PointNet++ [11], PointCONV [12], DGCNN [13], and Point Transformer [15]. To make the comparison fair, all methods were evaluated using the same testing dataset, and the same training and testing protocols were applied under the same operating environment. In addition, all parameters involved in these methods were set in default.

Table II shows the comparison results. Note that, as the pioneer point-wise segmentation network, PointNet achieved 38.6% mIoU, which is far from promising segmentation results. Besides, as the advanced version of PointNet, PointNet++ employed sampling and grouping techniques in multiple scales for local feature extraction from unordered point clouds, yet the mIoU decreased by 6.6% due to nonuniform data distributions in urban road scenes and relatively limited global feature learning capability. Compared with the low-level feature encoding enforcement in both PointNet and PointNet++, the improved performance of PointCONV and DGCNN was because of the investigation of deep, high-level, and inherent feature descriptiveness through powerful transformation operators and edge convolutions. Consequently, PointCONV and DGCNN achieved 60.5% mIoU and 52.9% mIoU, respectively. As demonstrated in [16], the semantic segmentation performance will be improved if the SA mechanism is replaced by the OA mechanism. Thus, Point Transformer that used the SA layers achieved 78.7% mIoU, which was lower than the STN embedded with SA layers. In contrast, due to effective feature saliency construction, significant point-wise global feature encoding enhancement boosted by the OA operations, and improved edge convolutions for local feature promotion, the proposed STN model outperformed these point-wise segmentation networks with respect to both OA and mean IoU. On the whole, the STN model presented a promising solution to road object semantic segmentation from large-scale MLS point clouds.

IV. CONCLUSION

In this letter, we introduce a saliency-guided transformer network, called STN, for point-wise semantic segmentation from MLS point clouds of urban scenarios. Training the STN model with four types of salient features of MLS point clouds, the transformer-based architecture of the STN could extract inherent, descriptive, and high-level feature representations to achieve accurate road object semantic segmentation in complex urban road environments. Benefitting from OA mechanisms for effective attention weight enhancement and edge convolutions for powerful local feature encodings, both point semantics and feature quality are remarkably improved to boost the segmentation performance of the STN model. Experimental results show that the proposed methods could deliver an average of 87.2% OA and 81.7% mIoU across all the object categories of the Paris-Lille-3D dataset. Comparative experiments with five deep learning-based methods also demonstrate the superior performance of the STN model in point-wise semantic segmentation tasks, especially in large-scale and complex urban road scenes.

REFERENCES

[1] Y. Li *et al.*, "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2020.

[2] Y. Zhou *et al.*, "A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1981–1985, Nov. 2014.

[3] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[4] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.

[5] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 452–468.

[6] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5010–5019.

[7] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Comput. Graph.*, vol. 71, pp. 189–198, Apr. 2018.

[8] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat, "SnapNet-R: Consistent 3D multi-view semantic labeling for robotics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 669–678.

[9] Y. Yang, F. Chen, F. Wu, D. Zeng, Y.-M. Ji, and X.-Y. Jing, "Multi-view semantic learning network for point cloud based 3D object detection," *Neurocomputing*, vol. 397, pp. 477–485, Jul. 2020.

[10] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. NIPS*, vol. 30, 2017, pp. 1–10.

[12] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. NIPS*, vol. 31, 2018, pp. 1–11.

[13] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[14] L. Ma *et al.*, "Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 821–836, Feb. 2019.

[15] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2021, pp. 16259–16268.

[16] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[17] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *Int. J. Robot Res.*, vol. 37, no. 6, pp. 545–557, 2018.