Point Cloud Denoising and Outlier Detection with Local Geometric Structure by Dynamic Graph CNN

Kosuke Nakayama
Graduate School of Fundamental
Science and Engineering
Waseda University
Tokyo, Japan
kosuke0013@fuji.waseda.jp

Hiroto Fukuta
Graduate School of Fundamental
Science and Engineering
Waseda University
Tokyo, Japan
taketomohiro@akane.waseda.jp

Hiroshi Watanabe
Graduate School of Fundamental
Science and Engineering
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

***Abstract*—The digitalization of society is rapidly developing toward the realization of the digital twin and metaverse. In particular, point clouds are attracting attention as a media format for 3D space. Point cloud data is contaminated with noise and outliers due to measurement errors. Therefore, denoising and outlier detection are necessary for point cloud processing. Among them, *PointCleanNet* is an effective method for point cloud denoising and outlier detection. However, it does not consider the local geometric structure of the patch. We solve this problem by applying two types of graph convolutional layer designed based on the *Dynamic Graph CNN*. Experimental results show that the proposed methods outperform the conventional method in AUPR, which indicates outlier detection accuracy, and Chamfer Distance, which indicates denoising accuracy.**

Keywords*—*point cloud, denoising, outlier detection, digital twin

I. Introduction

Digital technology is rapidly progressing in various fields toward the realization of the digital twin and metaverse. In particular, point clouds are attracting attention as a format for representing three-dimensional space. Point cloud data can be acquired with laser scanner. However, it contains outliers and noise due to instrument limitations. Noisy data reduces the accuracy of later processing such as object detection, segmentation, and reconstruction. Therefore, denoising and outlier detection are important technologies that form the basis of point cloud processing.

In general, there are statistical methods for point cloud denoising and outlier detection. One is bilateral filtering [1] based on the distance between each point and a plane fitted to the neighboring points. The other is moving least square [2] that projects each point onto a surface fitted to the neighborhood points. However, these two methods cannot handle all planes, surfaces, and edges and have the problem of excessive smoothing.

To address this limitation, deep learning methods for point cloud denoising and outlier detection have been developed in the last few years. In particular, *PointCleanNet* [3] is an effective deep learning-based method. However, it does not consider the local geometric structure of the patch. Therefore, it is difficult to capture noise and outliers.

To solve this problem, we apply two types of graph convolutional layers based on *Dynamic Graph CNN* [4]. The first is to dynamically construct a neighborhood graph in each layer. The second is to propagate the neighborhood search results in low-dimensional space to the backward layers. These proposed methods solve the problem of not considering the relationship between points in a patch. Experimental results confirm that the proposed methods outperform the conventional method in AUPR, which indicates outlier detection accuracy, and Chamfer Distance, which indicates denoising accuracy.

II. Related Works

A. PointCleanNet

PointCleanNet is a point cloud denoising and outlier detection method using deep learning. It consists of two stages: *1) Outlier Detector* for outlier removal and *2) Denoiser* for noise correction. Specifically, the outlier detector classifies outliers and discards them from the original point cloud. Next, the denoiser estimates correction vectors that project the noisy points onto the original clean surface. The method is efficient and robust to various levels of outliers and noise. It is also very easy to incorporate into existing shape processing pipelines because of its simplicity and versatility. However, there are problems with remaining outliers and unnatural distortion of planes and surfaces. This is because the local geometric structure within the patch is not considered in the input point cloud.

B. Dynamic Graph Convolutional Neural Networks

Dynamic Graph CNN is a point cloud classification and segmentation method using deep learning. In the process, the nodes of the graph represent points and edges are established based on pairwise distances between points. This structure effectively transforms point cloud data into a graph representation. In particular, by defining convolution operations on dynamic graphs, local geometric features can be learned. The topology of the graph is used to adaptively determine the receptive field of each point, allowing the user to focus on local neighborhood information that is important for understanding the 3D shape. In addition, mathematical analysis to optimize graph convolutional networks is discussed [5].

979-8-3503-4017-4/23/$31.00 ©2023 IEEE

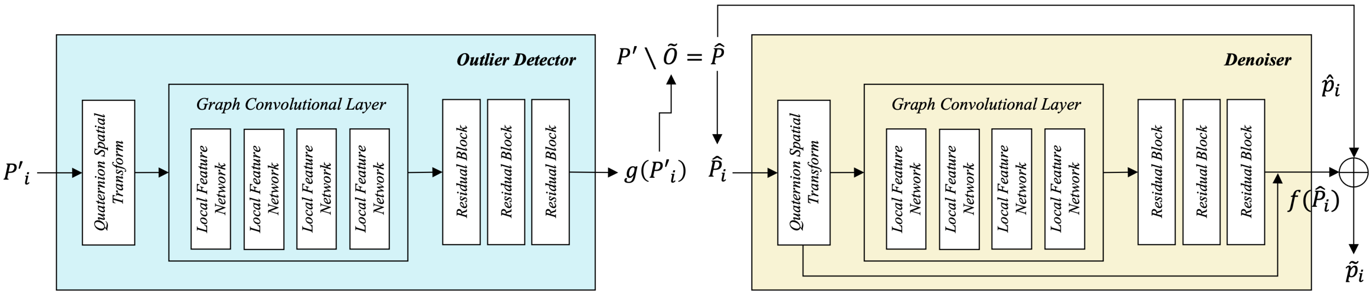


Fig. 1. The architecture of the proposed method: Outlier Detector and Denoiser.

III. Proposed Method

A. Architecture

The architecture consists of two stages: *1) Outlier Detector* for outlier removal and *2) Denoiser* for noise correction. It is shown in Fig. 1. It consists of quaternion spatial transform, graph convolutional layer, and residual block. In particular, the graph convolutional layer consists of four local feature networks. In this network, local neighborhood graphs are constructed by k-nearest neighbor search. Then, convolution and addition operations using them are used to generate edge features that can represent relationships between points. They allow the calculation to reflect the local geometric structure information while maintaining the global shape information of the patch.

Quaternion spatial transform is a network that outputs a rotation matrix using quaternions to rotate the patch. It is effective for translational invariance and implicitly learns the rotation transformation to be robust to outliers and noise. Residual block is a network that adds skip connections to the conventional structure. It can be effective in addressing the vanishing gradient problem.

B. Outlier Detector

The outlier detector takes a local patch P'_i of outlier points P as input and outputs the outlier estimation probability $g(P'_i)$ for each point in the patch. After that, an outlier is determined based on the set threshold value. Points determined to be outliers are added to the outlier set $\tilde{O}$. The outlier determination is given by

$$\mathit{If}\ \tilde{o}_i > \mathit{Threshold}\ :\ \tilde{o}_i \in \tilde{O}. \tag{1}$$

Then, point cloud $\hat{P}$ with outliers removed is given by

$$\hat{P} = P' \setminus \tilde{O}. \tag{2}$$

For the loss, we used the manhattan distance L_o between the estimated outlier label $\tilde{o}_i$ and the correct outlier label o_i. The loss function of outlier detector is given by

$$L_o(\tilde{p}_i, p_i) = \| \tilde{o}_i - o_i \|_1. \tag{3}$$

C. Denoiser

The denoiser takes as input the local patch $\hat{P}_i$ of noisy points $\hat{P}$ after removing outliers, and outputs the noise correction vector $f(\hat{P}_i)$ for each point. The denoised smooth point cloud $\tilde{P}$ is given by

$$\tilde{P} = \hat{P} + f(\hat{P}). \tag{4}$$

For the loss function, we used L_α, a combination of two types of losses L_s and L_r. L_s is the square of the L2 distance between each point after denoising and the nearest point in the neighborhood centered at the correct data point corresponding to that point. On the other hand, L_r is the square of the euclidean distance between the point and the farthest point. These two were weighted by α, L_α. These loss functions are represented by

$$L_s(\tilde{p}_i, P_{\tilde{p}_i}) = \min_{p_j \in P_{\tilde{p}_i}} \| \tilde{p}_i - p_j \|_2^2, \tag{5}$$

$$L_r(\tilde{p}_i, P_{\tilde{p}_i}) = \max_{p_j \in P_{\tilde{p}_i}} \| \tilde{p}_i - p_j \|_2^2, \tag{6}$$

$$L_\alpha(\tilde{p}_i, P_{\tilde{p}_i}) = \alpha L_s + (1-\alpha) L_r. \tag{7}$$

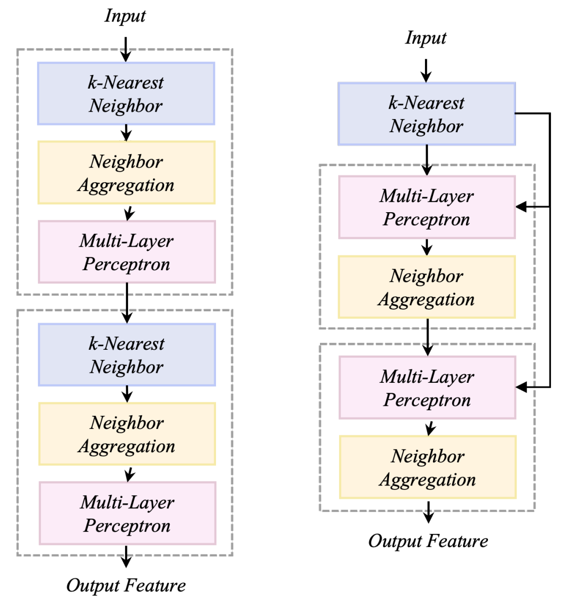
D. Difference of Two Graph Convolutional Layers

We introduced two types of graph convolution layers. They are shown in Fig. 2. In the first method, based on the normal *Dynamic Graph CNN*. The k-neighborhood graph is not fixed, and the graph is dynamically updated after each local feature network. This allows a detailed analysis of the local structure of the neighborhood. In the second method, the graph was designed based on the optimization of graph convolution. The graph is fixed, and the k-neighborhood graph in low-dimensional space is also reflected in each local feature network. This is expected to improve the computational speed compared to the usual graph convolution.

IV. Experiment

A. Datasets

For the outlier detector, the *PointCleanNet* outlier dataset consisting of 28 different point clouds with 140000 points for each shape is used. For the denoiser, the *PointCleanNet* noise dataset consisting of 28 different point clouds with 100000 points for each shape is used.



(a) Ours1: Normal Graph Convolutional Layers

(b) Ours2: Optimized Graph Convolutional Layers

Fig. 2. Comparison of two graph convolutional layers. (a) Nomal graph convolution of point clouds. (b) Optimized graph convolution of point clouds.

B. Experimental Parameters

For the outlier detector, the outlier threshold is set to 0.5 and the number of k-nearest neighbor searches to 16. The parameters used in the training were batch size 16, learning rate 10^{-4}, number of epochs 800, and initial values of network weights were initialized by He [6]. For the denoiser, The α in the loss function L_α is 0.99 and the k-nearest neighbor search is set to 16. The parameters used for training were a batch size of 16, a learning rate of 10^{-8}, an epoch count of 800, and a uniform random value of $(-0.001, 0.001)$ for the initial network weights.

C. Evaluation Method

The average AUPR was used for comparison of outlier detection accuracy. This is because the point cloud of the data set has a bias in the number of outlier and nonoutlier points; the AUPR is a curve representing the change in the reproducibility-fitness ratio of the test results and the study of the area of curvilinearization. The area value ranges from 0 to 1, with larger values indicating higher prediction accuracy. The average value of Chamfer Distance was used for comparison of denoising accuracy. The similarity of point clouds can be measured by evaluating them against each other from both predicted and ground truth, with lower values indicating higher denoising accuracy. The concept of Chamfer Distance is shown in Fig. 3. Chamfer Distance(CD) is given by

$$CD = \frac{1}{\tilde{P}}\sum_{p_i \in \tilde{P}} \min_{p_j \in P} \|p_i - p_j\|_2^2 + \frac{1}{P}\sum_{p_j \in P} \min_{p_i \in \tilde{P}} \|p_j - p_i\|_2^2. \quad (8)$$

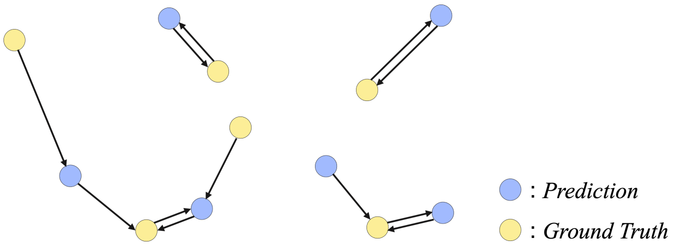


Fig. 3. The concept of Chamfer Distance.

D. Accuracy Evaluation

1) Outlier Detectoion: The results of outlier detection accuracy by AUPR are shown in Table I. It shows that ”Ours1” has the best performance. It outperforms the *PointCleanNet* score by 0.048 when the noise level is 1.0.

2) Denoising: The results of denoising accuracy by Chamfer Distance are shown in Table Ⅱ. It shows that ”Ours1” has the best performance. It outperforms the *PointCleanNet* score by 0.0035 when the noise level is 1.5.

E. Calculation Speed Evaluation

The comparison of computational speed of the proposed outlier detector and denoiser is also shown in Table Ⅲ. This shows that the performance of the ”Ours2” is superior. In particular, the outlier detector can compute each point in 0.36×10^{-4} seconds. Thus, it can be seen that $1,400,000$ points can be computed 8.4 minutes faster. Also, the denoiser can calculate each point in 0.31×10^{-4} seconds. Thus, it can compute $1,000,000$ points 5.2 minutes faster.

TABLE I
RESULTS OF THE OUTLIER DETECTION ACCURACY BY AUPR WITH OUTLIER DETECTOR

Model	*Gaussian Noise Level*			
	0%	1.0%	1.5%	2.5%
PointCleanNet	0.957	0.858	0.781	0.659
*Ours*1 (k=16)	**0.972**	**0.906**	**0.821**	**0.670**
*Ours*2 (k=16)	0.969	0.902	0.805	0.665

TABLE II
RESULTS OF THE DENOISING ACCURACY BY CHAMFER DISTANCE WITH DENOISER

Model	*Gaussian Noise Level*		
	1.0%	1.5%	2.5%
PointCleanNet	0.0123	0.0224	0.137
*Ours*1 (k=16)	**0.0109**	**0.0189**	**0.127**
*Ours*2 (k=16)	0.0111	0.0208	0.134

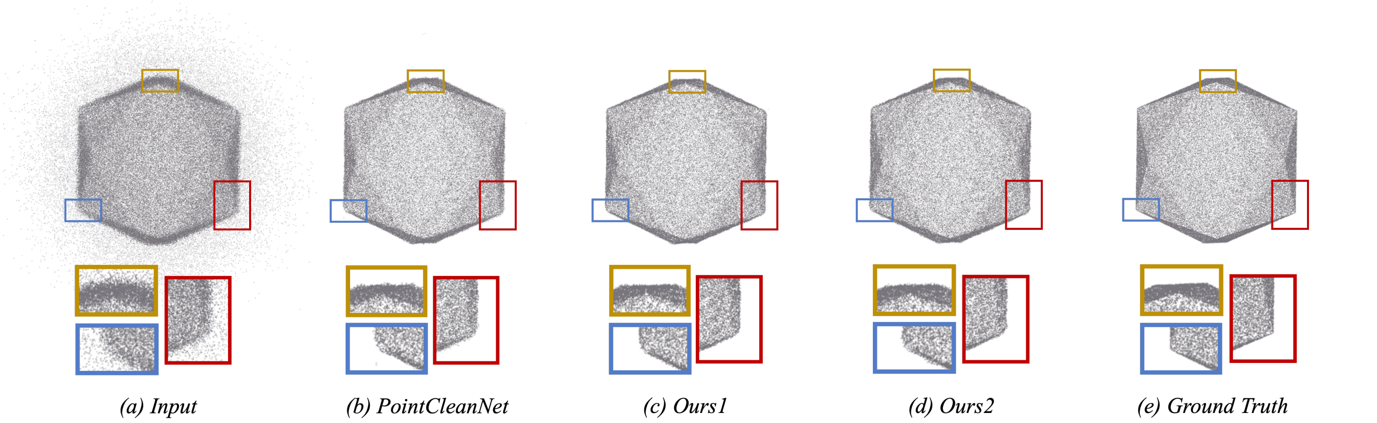


Fig. 4. Experiments of qualitative results of the overall model for the icosahedron data. (a) Point clouds contaminated by outliers and noise. (b) Processing results of conventional method. (c) Processing results of proposed method1 (Ours1). (d) Processing results of proposed method2 (Ours2). (e) Ground truth.

TABLE III
Comparison of Computation Time Per Point for Outlier Detector and Denoiser

Model	*Outlier Detector* (s)	*Denoiser* (s)
*Ours*1 (k=16)	2.57×10^{-4}	2.38×10^{-4}
*Ours*2 (k=16)	$\mathbf{2.21 \times 10^{-4}}$	$\mathbf{2.07 \times 10^{-4}}$

TABLE IV
Results of The Overall Model Performance

Model	*Chamfer Distance*	*Time* (s)
PointCleanNet	1.56×10^{-4}	–
*Ours*1 (k=16)	$\mathbf{1.03 \times 10^{-4}}$	5.02×10^{-4}
*Ours*2 (k=16)	1.04×10^{-4}	$\mathbf{4.35 \times 10^{-4}}$

F. Overall Model Evaluation

To evaluate the performance of the overall model, which consists of *Outlier Detector* and *Denoiser*. The data is applied to a contaminated icosahedron. It is subjected to gaussian noise with a standard deviation of 20% on the diagonal of the shape bounding box. In addition, 30% of the points farther from the surface than 1.5% standard deviation are converted as outliers.

Quantitative results are shown in TABLE IV. It shows that the "Ours1" is the best. In particular, it can be seen that for *PointCleanNet*, the chamfer distance was reduced by 0.53×10^{-4}. It can also be seen that the "Ours2" is slightly inferior to "Ours1", but performs equally well. In addition, it can reduce the computation time by 13.3%, showing that two proposed methods are efficient.

Qualitative results are also shown in Fig. 4. It can be seen that the two proposed methods detect outliers at a higher gaussian level than the conventional methods. It shows that they can reduce the unnatural distortion of the plane and smoothing of the edge areas. These results confirm the effectiveness of the two proposed methods. From the above, "Ours1 outperforms the conventional method in terms of performance. In addition, "Ours2 shows that it is more efficient than the method that computes the graph dynamically, in addition to having the same performance as the method that computes the graph dynamically.

V. Conclusion

In this paper, we proposed new method, point cloud denoising and outlier detection. Specifically, we introduced two types of graph convolutional layer based on *Dynamic Graph CNN* to *PointCleanNet*. Experiments show that the two proposed models are superior in AUPR and chamfer distance. Thus, the effectiveness of the two models was confirmed. In particular, we found that the "Ours1" is accuracy specific, while the proposed "Ours2" is balance between accuracy and computation speed. Depending on the issue to be applied, it is necessary to be flexible in pursuit of accuracy or in consideration of calculation speed.

References

[1] J. Digne and C. Franchis, "The Bilateral Filter for Point Clouds," Image Processing On Line (IPOL), pp.278-287, Jul. 2017.

[2] M. Alexa, J Behr, D. Cohen-Or S. Fleishman, D. Levin, C.T. Silva, "Point Set Sur faces," VIS '01: Proceedings of the conference on Visualization '01, pp.21-28, Oct. 2001.

[3] M.J. Rakotosaona, V .L. Barbera, P . Guerrero, N.J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," In Computer Graphics Forum, vol.39, pp.185-203, Wiley Online Library, 2020.

[4] 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 Transactions on Graphics (TOG), vol.38, Jun. 2019.

[5] Y. Li, H. Chen, Z. Cui, R. Timofte, M. Pollefeys, G. Chirikjian, and L.V. Gool, "Towards Efficient Graph Convolutional Networks for Point Cloud Handling," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp.3732-3742, Apr. 2021.

[6] K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," 2015 IEEE International Conference on Computer Vision (ICCV), pp.1026-1034, Dec. 2015.