Pattern Recognition 161 (2025) 111228

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

PyramidPCD: A novel pyramid network for point cloud denoising

Zheng Liu, Weijie Zhou, Chuchen Guo, Qinjun Qiu *, Zhong Xie

School of Computer Science, China University of Geosciences (Wuhan), Wuhan, 430074, China

ARTICLE INFO

Keywords:
Point cloud denoising
Feature pyramid
Transformer

ABSTRACT

Point cloud denoising, which aims to restore high-quality point clouds from noisy input, is an ingredient in various fields, including 3D mapping, 3D vision, and structured modeling. In this study, we present a feature pyramid network that can effectively remove noise while accurately preserving structural-to-detailed shape characteristics at varying scales. Our method is built upon the key insight that the coarser scale in the feature pyramid naturally contains more primary structures, while the finer scale provides more shape details. This discovery motivates us to progressively upgrade the current-scale feature with coarser-scale features to preserve structures while recovering details from the finer scale. To this end, we present a U-Net-shaped architecture that incorporates structure-aware and detail-preserving units at multiple scales for feature-preserving point cloud denoising. The structure-aware unit can enhance the current-scale feature by applying structural guidance from the coarse level of the feature pyramid, while the detail-preserving unit can learn a more comprehensive representation that incorporates the finer-scale feature in the pyramid. Extensive experiments conducted on publicly available benchmarks demonstrate that the proposed approach achieves state-of-the-art performances and outperforms the competing methods. Our source code and data are available at https://github.com/ForestNobear/PyramidPCD.

1. Introduction

Recent advancements in 3D scanning techniques have enabled us to digitize the real world more quickly, cheaply, and conveniently. This facilitates numerous fields, including 3D vision, computer graphics, and robotics [1]. Nevertheless, because of limitations imposed by physical and optical mechanism constraints, the raw data acquired by scanning devices is inevitably contaminated by various disturbances, such as physical sensors, optical imaging, natural environment interference, and other factors [2]. These disturbances induce noise to the point clouds, degrading the quality of the point clouds and causing difficulties in downstream geometry processing [3]. Thus, point cloud denoising has therefore long been an ingredient within structured modeling, digital twin, and 3D mapping [4,5].

In the last decades, a great deal of traditional point cloud denoising methods have been suggested, including moving least squares-related techniques [6], optimization-based approaches [7], locally optimal projection (LOP) and its variants [8–10], and low-rank and dictionary learning strategies [1,11–13]. However, most existing traditional methods typically rely on manual parameter tuning, which requires specialist knowledge for parameter selection. In addition, the time-consuming and laborious tuning procedure is crucial for producing promising results, resulting in the limited generalizability of the traditional methods.

In recent years, many learning-based methods using diverse neural network architectures have been presented as preferable alternatives to traditional methods. One key advantage of these learning-based methods is their parameter-free nature at inference time, which makes them particularly suitable for autonomously handling large volumes of noisy data, a task that would otherwise be laborious and time-consuming for humans. Despite the significant strides made by these learning-based methods in point cloud denoising, developing intricate networks to learn more discriminative feature representations for better preserving geometric characteristics remains challenging. In other words, the network's feature learning ability is not just a factor but a crucial determinant that considerably impacts denoising performance.

Since PointNet [14], a pioneering work that learns feature representations from point clouds directly, many point cloud denoising methods [15–17] use PointNet and its variants as feature encoders. Later, PCDNF [18], IterativePFN [19], and CL [20] adopted DGCNN [21] as encoders to learn feature representations for denoising. Unfortunately, the existing methods that rely on PointNet, its variants, and DGCNN

* Correspondence to: No. 388 Lumo Road, Wuhan, China.
E-mail addresses: liuzheng@cug.edu.cn (Z. Liu), Zhouzz0722@cug.edu.cn (W. Zhou), guocc5519@mails.jlu.edu.cn (C. Guo), qiuqinjun@cug.edu.cn (Q. Qiu), xiezhong@cug.edu.cn (Z. Xie).

https://doi.org/10.1016/j.patcog.2024.111228
Received 1 July 2024; Received in revised form 25 September 2024; Accepted 24 November 2024
Available online 30 November 2024

as encoders have limitations. These methods struggle to capture multiscale surface characteristics well. Recovering multiscale characteristics at high quality remains challenging since the underlying surfaces of point clouds may include highly complex information, ranging from structural to detailed. U-Net and its variants have demonstrated efficacy in learning multiscale and highly representative features for a variety of applications, including point cloud semantic segmentation [22], completion [23], and reconstruction [24]. However, previous research has not considered the complementary nature of multiscale representations in the feature pyramid, limiting the original U-Net's ability to recover structural-to-detailed characteristics across multiple scales.

In this paper, we introduce a feature pyramid network, dubbed PyramidPCD, that effectively removes noise while preserving structural-to-detailed shape characteristics at multiple scales. Our PyramidPCD is motivated by the insight that the coarse pyramid level reflects more main structures, while the finer level contains more shape details. PyramidPCD employs the U-Net architecture with five encoding and decoding layers to learn the multiscale feature pyramid. In each encoding layer, our encoder learns the geometry-supported encoder feature capable of geometric awareness. Then, we design a structure-aware Transformer to enhance the current encoder feature using the structural guidance from the coarser level of the pyramid. In each decoding layer, we utilize a detail-preserving connection to incorporate shape details into the decoder feature to learn a more comprehensive representation. Our method has been rigorously tested and compared with competing methods on various benchmarks. The numerical and visual results unequivocally present that our PyramidPCD not only matches but surpasses the performance of the competing methods, instilling confidence in its effectiveness.

Our main contributions are summarized below:

- We design a feature pyramid network for restoring noisy point clouds, which exploits the complementary nature of the feature pyramid to perceive structural and detailed shape characteristics in varying scales.
- We designed a structure-aware Transformer to enhance the encoder feature using the structural guidance from the coarser pyramid level. With such a design, multiscale structures from the feature pyramid can be fused into the encoder features hierarchically.
- We present a detail-preserving connection incorporating details from the finer pyramid level into the decoder feature. This connection operator can integrate shape details at varying scales into the decoder features to learn more comprehensive representations.

2. Related work

We only review learning-based denoising techniques on point clouds that are most closely associated with our method, even though numerous classical approaches are available. A thorough review of classical approaches can be found in [25], which interested readers could consult.

Many denoising methods directly infer the displacements of noisy points to the clean surface and then move these noisy points along the predicted displacements for denoising. Following this paradigm, PointCleanNet [15] first removes outliers and predicts displacements for remaining noisy points through a distance minimization loss. Pointfilter [16] shares a similar network structure to PointCleanNet, but it elaborately designs a bilateral loss function that considers both point and normal information to preserve sharp features well. Wang et al. [26] introduced FCNet as a way to reduce feature noise and train it using a teacher-student strategy that uses feature domain constraints to help the students identify clean features. PointFilterNet [27] generates denoised points by learning three coefficients and filtering point coordinates by these learned coefficients around noisy points. Chen et al. [17] presented a recurrent network architecture that learns multiscale feature representations well. Nevertheless, their technique struggles to strike a compromise between noise removal and geometric information preservation when dealing with high noise. Li et al. [2] proposes a novel single-stage point cloud cleaning network that simultaneously removes outliers and denoises within a single model. IterativePFN [19] was recently developed to simulate the progressive smoothing process inside the network. It incorporates a progressive loss that gradually moves denoised points closer to the underlying surface during each internal iteration. More recently, Wei et al. [28] proposed a reinforcement learning-based technique to restore noisy point clouds, enabling the selection of ideal paths for relocating noisy points onto a clean surface.

Some denoising approaches follow a two-phase paradigm: smoothing noisy normals and then updating point positions according to the smoothed normals. Lu et al. [29] divided the noisy point cloud into two point sets, only inferring multiple normals on the feature point set to preserve geometric features. Wei et al. [30] developed a dual neural network to filter noisy normals based on geometry information. Even though the proposed method can yield satisfactory results, its overall usefulness is limited by the need to compute the homogeneous neighbors for each point. To produce high-quality normals, Zhou et al. [31] and Zhang et al. [32] devised a two-step technique that treats normal filtering and refinement as two separate tasks. Recently, there have been several methods of exploring jointly denoising and normal filtering. Edirimun et al. [20] adopted a contrastive learning mechanism to train the decoder, and consequently, the decoder outputs a vector containing both predicted denoised points and normals. In contrast, PCDNF [18] and PN-Internet [33] take a multitask perspective, utilizing two interactive network branches to restore noisy points and normals simultaneously.

Some denoising techniques are explored to predict clean underlying surfaces from noisy point clouds for noise removal. TotalDenoising [34] shifts the noisy points projecting to the clean surface unsupervised with the spatial prior. However, this approach may be subject to a high level of noise. DMRDenoise [35] first identifies an underlying downsampled surface to reconstruct the clean surface, and subsequently upsamples points on the reconstructed clean surface for denoising. After estimating the noise-convolved distribution score from the noisy input, ScoreDenoise [36] employs gradient ascent to restore the noisy point cloud. Zhao et al. [37] perturbed the embedding features in the latent space and captured the commonality between the features to reconstruct the latent clean surface. Mao et al. [38] proposed PD-Flow to model the Euclidean-to-latent space mapping with normalizing flows for restoring noisy point clouds. Chen et al. [39] employed the gradient field to project the noisy points converges to the underlying clean surface.

3. Proposed method

We begin with a statement of point cloud denoising in this section. Then, we present our network design and elaborate on its core modules. Finally, we discuss our training loss.

3.1. Problem statement

Given a noisy point cloud P, we first apply farthest point sampling to obtain reference points $\{p_r\}$, and then use KNN to construct overlapping point cloud patches $\{\mathcal{P}\}$ as input to our network. For each noisy patch $\mathcal{P} = \{p_i | p_i \in \mathrm{KNN}(p_r, \mathrm{P}, m)\}$, where $\mathrm{KNN}(p_r, \mathrm{P}, m)$ refers to the m nearest neighbors of the reference point within P, our network is designed to denoise the corresponding patch supervised. Thus, our network can be formulated as:

$$\mathcal{F} : \mathcal{P} \rightarrow \mathcal{D}, \quad \widehat{\mathcal{P}} = \mathcal{P} + \mathcal{D}. \tag{1}$$

$\mathcal{D}$ denotes predicted displacement vectors of $\mathcal{P}$, while $\widehat{\mathcal{P}}$ denotes the denoised patch. As the denoised patches $\{\widehat{\mathcal{P}}\}$ have overlapping regions, we employ the stitching strategy outlined in [19] to generate the final denoised point cloud $\widehat{\mathrm{P}}$.

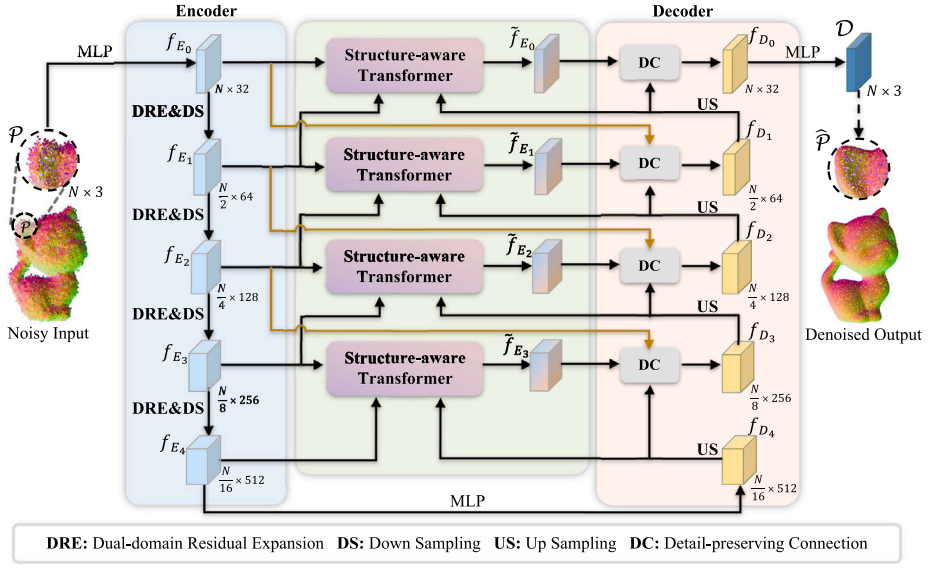

Fig. 1. An overview of our network, dubbed PyramidPCD. The noisy point cloud is partitioned into patches as inputs, and PyramidPCD outputs denoised point cloud patches. PyramidPCD comprises three core modules: an encoder, structure-aware Transformers, and a decoder. The encoder extracts multiscale encoder features using four cascaded encoding layers, each consisting of a dual-domain residual expansion (DRE) followed by a down-sampling (DS) operation. Then, structure-aware Transformers are employed to produce structural enhancing features at multiple scales. Following that, the decoder with four cascaded decoding layers, each containing up-sampling (US) interpolation and a detail-preserving connection (DC), is employed to produce decoder features.

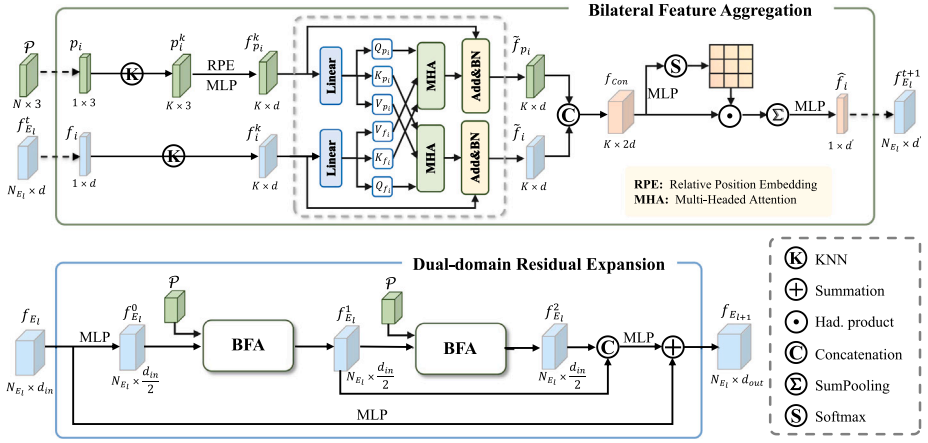

Fig. 2. Illustration of the encoder network's primary module, dual-domain residual expansion (DRE), which is made up of cascading bilateral feature aggregation (BFA) blocks. The top panel reveals the architecture of BFA. It utilizes the mutual guidance information from the bilateral feature space to compensate for correlations between geometry and feature domains. The bottom panel depicts the design of DRE, which integrates dense connections and residual learning to expand feature receptive fields.

3.2. Network architecture

Fig. 1 illustrates our network, leveraging the symmetric encoder–decoder architecture. Our network is composed of three core modules: an encoder for multiscale feature extraction, structure-aware transformers for feature enhancement, and a decoder for structural-to-detailed feature incorporation.

Considering a noisy patch $\mathcal{P}$ as input, our network begins with an encoder to learn a feature pyramid $\{f_{E_l}\}_{l=0}^{4}$, where $l = 0$ represents the finest scale and $l = 4$ corresponds to the coarsest scale. Specifically, the initial encoder feature f_{E_0} is derived from $\mathcal{P}$ through a multilayer perceptron (MLP). In each encoding layer, our encoder employs dual-domain residual expansion (DRE) to learn the higher-dimensional encoder feature, followed by a down-sampling operation to reduce point cloud resolution. For each scale except the coarsest, we feed the current encoder feature f_{E_l}, as well as the encoder feature $f_{E_{l+1}}$ and decoder feature $f_{D_{l+1}}$ from the coarse scale to our structure-aware Transformer to produce a structural-enhancement feature $\widetilde{f}_{E_l}$. Thus, we can obtain an enhanced feature pyramid $\{\widetilde{f}_{E_l}\}_{l=0}^{3}$, where $\widetilde{f}_{E_l}$ represents the enhanced feature at the l−th scale of the pyramid. In the decoder part, we perform the decoding progressively through four decoding layers to obtain the decoder features denoted as $\{f_{D_l}\}_{l=0}^{4}$. In each decoding layer, we feed the enhanced feature $\widetilde{f}_{E_l}$ and the decoder feature f_{D_l} of the current scale, and the finer-scale encoder feature $f_{E_{l-1}}$ to our detail-preserving connection (DC) to produce the decoder feature $f_{D_{l-1}}$. Finally, the displacement vectors $\mathcal{D}$ of the noisy patch $\mathcal{P}$ can be inferred from the uppermost decoder feature f_{D_0}, and the denoised patch $\widehat{\mathcal{P}}$ can be obtained directly according to the problem statement (1).

3.3. Encoder

Our encoder's main module is dual-domain residual expansion (DRE), consisting of cascaded bilateral feature aggregation (BFA) blocks, as shown in Fig. 2. Using the encoder feature f_{E_l} at the lth scale and geometric information $\mathcal{P}$ as input, our DRE can learn the encoder

feature $f_{E_{l+1}}$ for the coarser scale. The mutual guidance information calculated by cross-attention from the bilateral feature space allows each point to capture a richer local context, compensating correlations between geometric and feature domains. Furthermore, we employ a self-attention mask to extract salient information from the dual feature spaces, hence improving the final feature representation.

3.3.1. Bilateral feature aggregation

For each point $p_i \in \mathcal{P}$, its KNN neighborhood points are collected and represented as $p_i^k \in \mathbb{R}^{K\times 3}$. Concurrently, we extract each point's feature representation f_i and derive its neighboring feature f_i^k. Similarly to [40], we first encode the relative positions on p_i^k and then use MLP to learn the feature representation of p_i^k as

$$f_{p_i}^k = \mathrm{MLP}\left(\left[\overline{p_i}, p_i^k, \left(\overline{p_i} - p_i^k\right), \left\|\overline{p_i} - p_i^k\right\|\right]\right) \in \mathbb{R}^{K\times d}, \tag{2}$$

where $\overline{P_i}$ represents the replicated features of p_i, expanding its dimension to match the neighborhood. The operator $[\cdot]$ denotes concatenation, and $\|\cdot\|$ is a metric that calculates the Euclidean distance between positions.

To utilize mutual guidance between low-level geometric features directly extracted from 3D coordinates and high-level features learned through several encoding layers, we employ cross-attention [41] mechanism to establish correlations between the geometric feature $f_{p_i}^k$ and the high-level feature f_i^k. Specifically, the features $f_{p_i}^k$ and f_i^k undergo linear projections, denoted by the operation $\varphi(\cdot)$, resulting in the generation of two distinct sets, each comprising query, key, and value elements as follows:

$$\left\{Q_{p_i}, K_{p_i}, V_{p_i}\right\} = \left\{\varphi\left(f_{p_i}^k\right), \varphi\left(f_{p_i}^k\right), \varphi\left(f_{p_i}^k\right)\right\},$$
$$\left\{Q_{f_i}, K_{f_i}, V_{f_i}\right\} = \left\{\varphi\left(f_i^k\right), \varphi\left(f_i^k\right), \varphi\left(f_i^k\right)\right\}.$$

Then, we implement a mutually guided feature integration through the multi-head attention mechanism (MHA) as

$$\widetilde{f}_{p_i} = \mathrm{BN}\left(\mathrm{MHA}\left(Q_{p_i}, K_{f_i}, V_{f_i}\right) + f_{p_i}^k\right), \tag{3}$$

$$\widetilde{f}_i = \mathrm{BN}\left(\mathrm{MHA}\left(Q_{f_i}, K_{p_i}, V_{p_i}\right) + f_i^k\right), \tag{4}$$

where BN represents normalization. By iteratively applying the attention function h times, MHA is able to yield many attention heads in various feature spaces. These attention heads are then concatenated as

$$\mathrm{MHA}(Q,K,V) = \left[\mathrm{HA}_1, \mathrm{HA}_2, \ldots, \mathrm{HA}_h\right],$$
$$\mathrm{HA}_h = \mathrm{Softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h,$$

where Q_h, K_h, V_h represent the query, key, and value of the input feature in the hth head, respectively.

Then, we concatenate the features $\widetilde{f}_{p_i}$ and $\widetilde{f}_i$ to obtain the raw bilateral feature f_{Con}. To enhance the discriminative properties of f_{Con}, we employ a mask function of self-attention $\gamma(\cdot)$ to adaptively identify crucial elements within f_{Con}, which can be formulated as

$$\hat{f}_i = \mathrm{SumPooling}\left(\gamma(f_{Con}) \odot f_{Con}\right) \in \mathbb{R}^{1\times d'}. \tag{5}$$

In our implementation, $\gamma(\cdot)$ consists of an MLP with a Softmax function, and $\odot$ denotes the Hadamard product. SumPooling$(\cdot)$ squeezes the input features of K points into a single feature by summation.

After performing the above operations on each point, the bilateral feature representation can be derived as $f_{E_l}^{t+1} = \left\{\hat{f}_i\right\}_{i=1}^{N_{E_l}} \in \mathbb{R}^{N_{E_l}\times d'}$, where N_{E_l} is the resolution of the current scale.

3.3.2. Dual-domain residual expansion

In the encoding phase, frequent aggregation and downsampling procedures may result in detailed and small-scale feature smoothing. To address the issue, we designed a dual-domain residual expansion (DRE) that leverages the aspects of residual learning to maintain information from expanding perception fields while compensating for detailed information loss.

The bottom panel of Fig. 2 illustrates the detailed architecture of DRE. Given $f_{E_l} \in \mathbb{R}^{N_{E_l}\times d_{in}}$ and $\mathcal{P} \in \mathbb{R}^{N\times 3}$ as input, we employ two cascaded BFA modules as feature extractors to learn the bilateral feature representations as

$$f_{E_l}^{t+1} = \mathrm{BFA}\left(f_{E_l}^t, \mathcal{P}\right), t = 0, 1, \tag{6}$$

$$f_{E_l}^0 = \mathrm{MLP}\left(f_{E_l}\right). \tag{7}$$

Then, the encoder feature $f_{E_{l+1}}$ in the coarser scale can be derived as

$$f_{E_{l+1}} = \mathrm{MLP}\left(\left[f_{E_l}^1, f_{E_l}^2\right]\right) + \mathrm{MLP}\left(f_{E_l}\right) \in \mathbb{R}^{N_{E_l}\times d_{out}}. \tag{8}$$

3.4. Structure-aware transformer

Since we observe that the coarse feature pyramid level reflecting large-scale structures frequently naturally eliminates details without smoothing structures, we argue that the scale is the greatest discriminative difference between details and structures. This insight prompted us to design our structure-aware Transformer, which performs under the guidance of coarser-scale features, efficiently preserving structural information without introducing extra details. Put simply, when the pyramid level's scale is coarser, small-scale details are smoothed out, while large-scale structures are well-preserved. Therefore, we rely on the coarse-scale features as guidance, which offer more reliable structure information, to enhance feature presentation at the current scale.

To modulate the current-scale encoder feature, we start by feeding the coarser-scale feature into an attention map as

$$\bar{f}_{E_l} = \mathrm{Softmax}\left(\mathrm{MLP}\left(\mathrm{US}\left(f_{E_{l+1}}\right)\right)\right) \odot f_{E_l}. \tag{9}$$

As shown in Fig. 3, for the encoder feature f_{E_l}, we exploit $f_{E_{l+1}}$ in the coarser scale to enhance f_{E_l}. $f_{E_{l+1}}$ is employed as the attention to modulate the structure representation of f_{E_l}. Next, in order to further refine structures and learn feature $\bar{\bar{f}}_{E_l}$ with the more discriminatory representation, we feed the structure-enhanced feature $\bar{f}_{E_l}$ into our structure-aware cross-attention unit (SCU).

Structure-aware cross-attention unit. As illustrated in the bottom left panel of Fig. 3, we employ our proposed SCU to refine the enhanced feature $\bar{f}_{E_l}$ under the guidance of coarser-scale feature $f_{D_{l+1}}$ Specifically, given $\bar{f}_{E_l}$ and $f_{D_{l+1}}$, we select $\bar{f}_{E_l}$ as query and $f_{D_{l+1}}$ as key–value pair, and then map them to the distinct set as follows:

$$\left\{Q_{E_l}, K_{D_{l+1}}, V_{D_{l+1}}\right\} = \left\{\varphi\left(\bar{f}_{E_l}\right), \varphi\left(\mathrm{US}\left(f_{D_{l+1}}\right)\right), \varphi\left(\mathrm{US}\left(f_{D_{l+1}}\right)\right)\right\}.$$

After that, we employ subtraction to acquire the correlation between $\bar{f}_{E_l}$ and $f_{D_{l+1}}$, and introduce $\bar{f}_{E_l}$ as information supplement. Thus, the correlation weight matrix W_{E_l} can be calculated as

$$W_{E_l} = \mu\left(\left(Q_{E_l} - K_{D_l}\right) + \bar{f}_{E_l}\right), \tag{10}$$

where $\mu(\cdot)$ is a mapping function implemented with a combination of an MLP followed by Softmax. With the correlation weight W_{E_l} in hand, we can learn the structure-refined feature $\bar{\bar{f}}_{E_l}$ as follows:

$$\bar{\bar{f}}_{E_l} = W_{E_l} \odot \left(V_{D_{l+1}} + \bar{f}_{E_l}\right). \tag{11}$$

Feature gate unit. After obtaining structure-enhanced and -refined features $\bar{f}_{E_l}$ and $\bar{\bar{f}}_{E_l}$, we employ a single update gate to fuse the features for adaptive learning the structure-aware feature. Specifically, we first compacts the information of $\bar{f}_{E_l}$ into vector $\mathbf{g}_l$ by $\mathbf{g}_l = \mathrm{Linear}\left(\bar{f}_{E_l}\right)$, then the refined gate $\mathbf{z}_l$ is given as

$$\mathbf{z}_l = \mathrm{MLP}\left(\bar{\bar{f}}_{E_l} + \mathbf{g}_l\right). \tag{12}$$

Finally, we obtain the structure-aware feature as follows:

$$\widetilde{f}_{E_l} = \mathbf{z}_l \odot \bar{\bar{f}}_{E_l} + \left(1 - \mathbf{z}_l\right) \odot \mathbf{g}_l \in \mathbb{R}^{N_{E_l}\times d}. \tag{13}$$

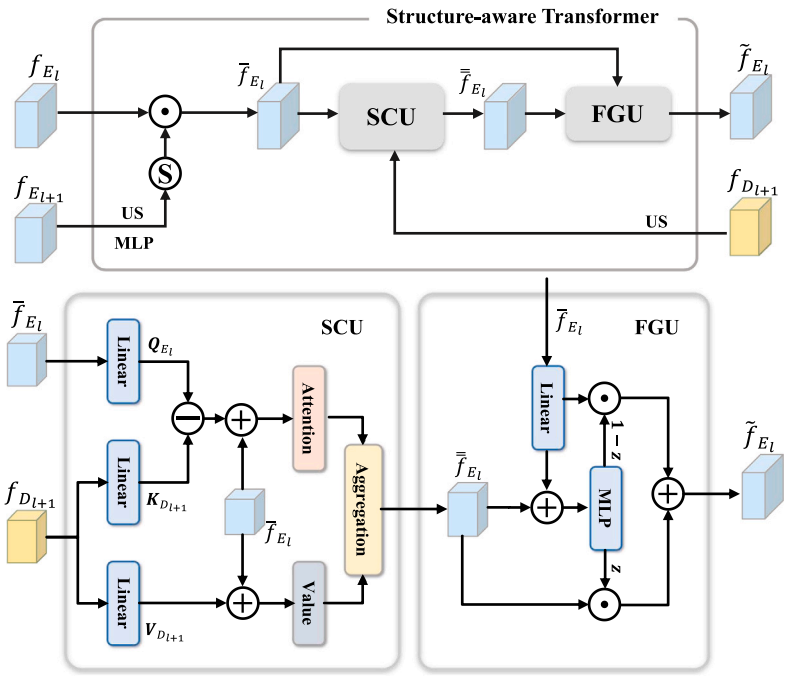

Fig. 3. The top panel reveals the architecture of our Structure-aware Transformer (ST). The bottom panel depicts the network design of Structure-aware Cross-attention Unit (SCU) and Feature Gate Unit (FGU).

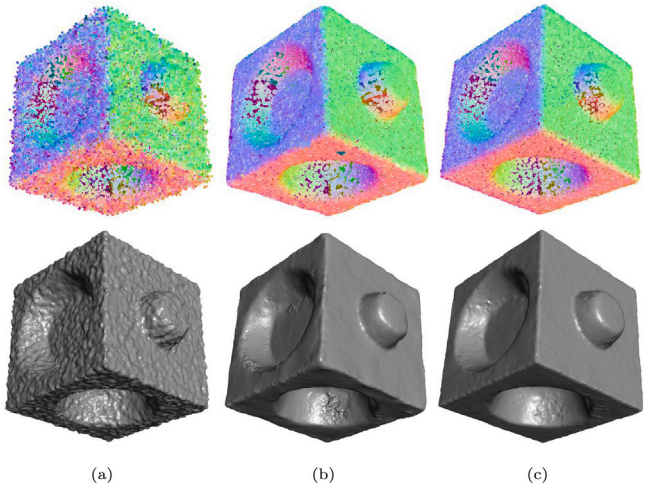

Fig. 4. Illustration of the structure-aware Transformer module. (a) Noisy input and its reconstruction result. (b) Denoising and reconstruction result without the structure-aware Transformer module. (c) Denoising and reconstruction result with the structure-aware Transformer module.

The above process is illustrated in the bottom right panel of Fig. 3. Fig. 4 illustrates the capabilities of the structure-aware Transformer module. The results presented in Fig. 4(c) demonstrate that the proposed module is crucial in preserving structural features and smooth regions. In its absence, as shown in Fig. 4(b), the denoising results may suffer from structural degradation.

3.5. Decoder

As mentioned before, the finer scale in the feature pyramid contains more details. Given the structure-aware feature $\widetilde{f}_{E_l}$, aligned encoder feature in finer scale providing details, and aligned decoder feature as input, our detail-preserving connection (DC) integrates these features to yield the decoder feature at the lth scale as

$$f_{D_l} = \begin{cases} \mathrm{MLP}\left(\widetilde{f}_{E_l}\right), & \text{if } l = 4 \\ \mathrm{MLP}\left(\left[f'_{D_{l+1}}, \widetilde{f}_{E_l}\right]\right), & \text{if } l = 0 \\ \mathrm{MLP}\left(\left[f'_{E_{l-1}}, f'_{D_{l+1}}, \widetilde{f}_{E_l}\right]\right), & \text{otherwise} \end{cases} \tag{14}$$

where $f'_{E_{l-1}}$ and $f'_{D_{l+1}}$ are the aligned features as follows:

$$f'_{E_{l-1}} = \mathrm{DS}\left(\mathrm{Conv}\left(f_{E_{l-1}}\right)\right), \tag{15}$$

$$f'_{D_{l+1}} = \mathrm{US}\left(\mathrm{TransConv}\left(f_{D_{l+1}}\right)\right). \tag{16}$$

Conv and TransConv represent convolution and transposed convolution, respectively.

Thanks to the detail-preserving connection (14), our decoder can learn a more comprehensive representation that incorporates information from various perceptual fields and retains structural and detailed information well without noticeable information loss and redundant information.

3.6. Training loss

We develop a loss function that determines the L_2 distance between each point in the restored patch and its closest point in the ground-truth (GT) patch to facilitate the proposed network's end-to-end training. Formally, the loss is defined in the following form:

$$\mathcal{L} = \sum_{\hat{p}_i \in \hat{\mathcal{P}}} w_i \min_{p_i^{\mathrm{GT}} \in \mathcal{P}^{\mathrm{GT}}} \|\hat{p}_i - p_i^{\mathrm{GT}}\|^2, \tag{17}$$

where $\hat{\mathcal{P}}$ denotes the denoised patch and $\mathcal{P}^{\mathrm{GT}}$ denotes the corresponding GT patch. The weight w_i of p_i is calculated as

$$w_i = \frac{\exp\left(-\|p_i - c\|^2 / 2r_s^2\right)}{\sum_i \exp\left(-\|p_i - c\|^2 / 2r_s^2\right)}, \tag{18}$$

where c and r_s denotes the reference point and support radius of patch $\mathcal{P}$, respectively. Referring to [19], the support radius can be calculated as $r_s = r/3\sqrt{2}$, where r is the radius of $\mathcal{P}$. The function (18) weights

Table 1
Quantitative comparison on PUNet [42] and Kinect [43] datasets. We bold and underline the denoising results that rank first and second, respectively. The metrics are CD ($\times 10^{-5}$) and P2M ($\times 10^{-5}$). The average value is abbreviated as AVE.

Metric	Methods	PUNet										Kinect			
		20K points			50K points			100K points			AVE	Fusion	v1	v2	AVE
		1%	2%	3%	1%	2%	3%	1%	2%	3%					
CD	PCN	16.33	23.02	35.42	7.13	10.01	22.08	4.17	6.96	25.08	16.69	7.4	13.73	22.48	14.54
	GPD	20.04	38.4	56.26	10.5	32.74	53.81	8.23	28.09	42.9	32.33	7.92	14.88	23.11	15.3
	DMR	24.23	19.18	38.43	14.66	18.73	27.4	12.86	16.68	25.5	21.96	13.82	–	–	–
	PF	14.83	18.31	27.62	7.41	9.23	22.12	5.91	7.78	**18.28**	15.28	7.87	13.82	18.96	13.55
	Score	15.03	22.89	30.95	7.16	12.88	19.28	4.32	10.97	18.77	15.81	8.43	13.23	19.66	13.77
	PDFlow	13.13	21.77	30.78	6.5	11.82	19.3	–	–	–	–	9.03	13.35	20.01	14.13
	IterativePFN	13.28	**17.1**	27.32	6.01	7.99	21.5	3.37	5.34	23.05	13.88	7.3	13.25	18.75	13.1
	Ours	**13.24**	17.28	**24.73**	**5.89**	**7.78**	**17.85**	**3.23**	**4.91**	18.41	**12.59**	**7.25**	**13.1**	**18.29**	**12.88**
P2M	PCN	6.21	9.62	17.34	3.19	5.56	14.76	3.33	5.32	20.05	9.49	6.49	8.75	13.29	9.51
	GPD	9.71	16.39	28.9	6.29	24.91	27.06	5.01	11.87	28.47	17.62	6.81	8.73	11.8	9.11
	DMR	11.95	16.07	23.95	7.35	10.81	18.4	7.23	10.66	18.75	13.91	9.55	–	–	–
	PF	4.96	6.89	14.3	4.74	5.77	15.07	4.68	6.2	**14.5**	8.57	6.56	**8.09**	10.31	8.32
	Score	5.39	10.78	17.45	4.0	8.32	13.27	3.6	8.94	14.73	9.61	6.74	8.17	11.08	8.66
	PDFlow	5.25	11.8	19.84	4.15	8.56	15.37	–	–	–	–	6.71	8.67	11.71	9.03
	IterativePFN	**3.86**	**6.01**	13.48	2.97	4.31	14.45	2.78	4.15	17.52	7.73	6.39	8.45	10.88	8.57
	Ours	**3.86**	6.11	**11.64**	**2.89**	**4.17**	**11.78**	**2.69**	**3.91**	14.53	**6.84**	**6.36**	8.32	**10.21**	**8.3**

input points with a Gaussian distribution according to their distance from the reference point.

4. Experiments

4.1. Datasets and implementation details

Training dataset. Following existing works [19,36,38], we train with data from the PUNet training set [42], which includes 40 meshes. Specifically, we sample point clouds from their corresponding meshes at resolutions of 50K and 100K using Poisson disk sampling, and then normalize the data into a unit sphere. We generate noisy data by adding Gaussian noise with standard deviations ranging from 0.5% to 2% of the radius of the unit sphere. During the training phase, we divided each point cloud into 1K-sized patches. This strategy saves memory while assuring that our model can effectively handle point clouds of arbitrary resolutions.

Testing datasets. We compare our method with competing approaches on synthetic testing sets of PUNet [42] with resolutions of 20K, 50K, 100K, and PCNet [15] with resolutions of 50K and 100K, respectively. Each tested point cloud is corrupted by Gaussian noise with standard deviations of 1%, 2%, and 3%, respectively. We also use real-scanned Kinect data [43] to validate our method's generalization capability. The Kinect data is separated into three subsets, each containing point clouds from different scanning equipment. Kinect V1 collects the first subset, containing 73 point clouds, using a single view, while Kinect V2 captures the second one, which has 72 point clouds. The third subset consists of four point clouds produced by integrating scans from various viewpoints obtained by Kinect V1. Ground truths for all tested shapes are derived from the corresponding high-resolution meshes. The Paris-rue-Madame dataset [44], which contains authentic street scenes recorded in Paris using a 3D mobile laser scanner, is also evaluated. This dataset contains real-world noisy artifacts caused by scanning technology constraints, yet it is a great foundation for comparing performance on real-world data.

Baselines. We numerically and visually compare our method to existing state-of-the-art approaches, including PCN [15], GPD [45], DMR [35], PF [16], Score [36], PDFlow [38], and IterativePFN [19]. We use the publicly available codes of the tested approaches mentioned above to retrain them on the tested datasets in order to provide fair comparisons.

Implementation details. We employ a single NVIDIA RTX4090 GPU with PyTorch 2.0 and CUDA 11.8 to train and infer our approach. The training process employs the Adam optimizer with default parameters. The initial learning rate is $8e^{-4}$. The maximum number of training epochs is 200, with a batch size of 32. The number of KNN neighbors is set at 16. To ensure a fair comparison of computational costs, all tested methods are evaluated on the same GPU.

4.2. Quantitative comparisons

Error metrics. To compare our method quantitatively, we follow the previous works and choose Chamfer distance (CD) and Point-to-Mesh distance (P2M) as our primary evaluation metrics. These metrics are extensively used in the fields of computer graphics and computer vision due to their effectiveness in measuring the accuracy and completeness of denoising results. Both P2M and CD are calculated using the latest implementations in PyTorch3D [19]. Better denoising performance is indicated by lower CD and P2M values.

Numerical evaluation. We first numerically assess our method on the PUNet dataset [42] based on the CD and P2M metrics and record the results in the left part of Table 1. We can observe that our approach outperforms the competing approaches in the majority of cases, yielding better-quality results with lower CD and P2M values, indicating that our results on the synthetic dataset with varying noise levels are more representative of the truth.

Next, we conduct comparison experiments between our method and those competitors on the Kinect dataset [43] and show the numerical results in the right part of Table 1. We demonstrate the superiority of our approach by achieving state-of-the-art results on both measures. In addition, when our method is applied to real-world scanned data, the evaluation outcomes show significant denoising capabilities and robust generalization performance.

After that, we present the comparative results for synthetic point clouds with varying noise levels from the PCNet [15] dataset in Table 2. Compared to the PUNet dataset, the tested shapes in PCNet are more complex, featuring greater structural complexity and more detailed information. Again, our PyramidPCD produces state-of-the-art performance.

Iteration mechanisms. To emphasize the efficiency of our method in noise reduction, we produced all denoised results within a single iteration, even for instances with 3% noise levels. In contrast, several methods, such as PF [16], PCN [15], and Score [36], typically require multiple iterations, especially for high noise data. These approaches utilize the denoised points from earlier iterations for further refinement.

Noise pattern evaluation. We also test our method against common noise patterns on synthetic data, including anisotropic Gaussian, Laplace, uniform distribution, and discrete noise. The robustness test

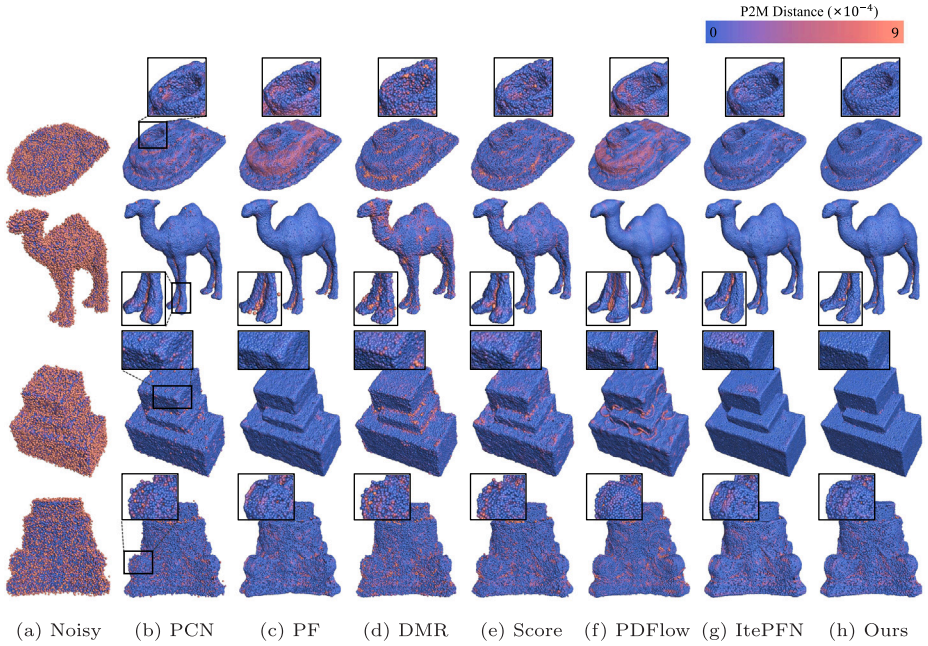

Fig. 5. Comparison of tested approaches of P2M distance for 50K resolution shapes with 2% Gaussian noise on synthetic datasets (PUNet and PCNet). Darker points (i.e., more blue) indicate the smaller distance from the ground truth.

Table 2
Quantitative comparison on the PCNet [15] dataset. We emphasize the denoising results that rank first and second by highlighting them in bold and underlining them, respectively. The metrics are CD ($\times 10^{-5}$) and P2M ($\times 10^{-5}$). The average value is abbreviated as AVE.

Metric	Methods	PCNet						
		50K points			100K points			AVE
		1%	2%	3%	1%	2%	3%	
CD	PCN	10.42	14.25	25.25	6.22	8.9	22.79	14.64
	GPD	19.71	48.41	89.51	12.43	23.07	46.2	39.89
	DMR	15.65	20.09	29.92	11.56	15.39	24.69	19.55
	PF	11.18	16.08	**22.62**	11.18	11.18	20.11	15.39
	Score	10.66	16.59	24.93	6.48	11.97	20.31	15.16
	PDFlow	9.69	16.66	24.83	–	–	–	–
	IterativePFN	9.27	**12.9**	25.27	5.35	7.99	22.88	13.94
	Ours	**9.22**	13.1	24.21	**5.15**	**7.93**	**18.72**	**13.06**
P2M	PCN	5.15	7.86	17.14	3.27	5.02	16.84	9.21
	GPD	12.52	22.09	43.92	8.12	12.98	29.44	21.51
	DMR	9.49	13.4	22.09	7.57	11.03	19.55	13.91
	PF	5.63	9.89	**15.29**	5.62	5.62	13.99	9.34
	Score	5.35	9.95	17.61	3.49	7.85	15.26	9.92
	PDFlow	4.66	11.64	18.73	–	–	–	–
	IterativePFN	4.12	7.09	16.97	2.53	4.58	16.75	8.67
	Ours	**3.83**	**6.88**	16.42	**2.14**	**4.37**	**12.92**	**7.76**

Table 3
Numerical evaluation of different noise patterns, including anisotropic Gaussian, Laplace, uniform distribution, and discrete noise. The unit of CD and P2M is 10^{-5}.

Noise type	Methods	50K points					
		1%		2%		3%	
		CD	P2M	CD	P2M	CD	P2M
Anisotropic	PF	7.65	4.39	9.77	5.67	**20.28**	**14.02**
	Score	7.11	4.01	13.19	8.62	20.87	14.89
	PDFlow	6.53	4.19	11.97	9.6	20.8	14.99
	IterativePFN	5.98	2.99	8.48	4.71	25.56	18.0
	Ours	**5.92**	**2.94**	**8.26**	**4.57**	22.16	15.59
Laplace	PF	8.31	4.87	12.61	7.66	29.5	21.61
	Score	8.25	4.89	16.77	11.67	**26.69**	**18.47**
	PDFlow	8.21	5.52	15.07	17.89	45.32	31.91
	IterativePFN	6.59	**3.41**	**10.03**	**5.77**	32.54	23.75
	Ours	**6.57**	**3.41**	10.25	6.09	31.84	23.43
Uniform	PF	6.52	4.26	7.66	4.29	7.79	3.72
	Score	5.05	2.88	6.91	3.78	9.0	4.21
	PDFlow	4.56	3.03	8.54	4.54	9.53	4.8
	IterativePFN	4.69	2.57	5.97	2.91	6.86	3.51
	Ours	**4.61**	**2.54**	**5.8**	**2.82**	**6.82**	**3.44**
Discrete	PF	5.51	3.98	6.32	4.25	7.22	3.49
	Score	4.48	2.91	6.16	3.87	8.01	3.71
	PDFlow	4.33	3.04	6.72	4.53	8.41	3.98
	IterativePFN	3.69	2.57	4.32	**2.78**	**4.97**	**3.18**
	Ours	**3.6**	**2.55**	**4.29**	2.8	5.07	3.3

about noise patterns is conducted on the PUNet dataset with a resolution of 50K. The corresponding parameters of these noise patterns are set in reference to [39]. The numerical results of our method and the competing ones are recorded in Table 3. As we can see, our method consistently produces superior results, demonstrating the robustness of our method against noise types and its potential for real-world applications.

4.3. Visual comparisons

In addition to the numerical evaluation, we offer qualitative comparisons to verify the effectiveness of PyramidPCD.

The denoising evaluation between our method and the competing approaches on the synthetic datasets is demonstrated in Fig. 5. As can be seen in the first and third rows of Fig. 5, all the compared methods oversmooth the piecewise transition regions and blur sharp edges and corners to some extent for CAD shapes. On the contrary, our PyramidPCD reconstructs transition regions with greater accuracy while preserving sharp edges and corners. Compared to the competing approaches, our PyramidPCD yields more appealing results that preserve clearer geometric information for non-CAD shapes. Thus, our PyramidPCD outperforms the competing ones in terms of visual quality, which can reduce unnecessary noise and artificial effects. See the second and fourth rows in Fig. 5 for example. We show the denoised Kinect data results in Fig. 6. As we can see, PCN [15] and PF [16] tend to maintain additional noise on the denoised surfaces, while Score [36], PDFlow [38], and IterativePFN [19] smooth the geometric features to some extent. In contrast, our method produces clearer denoising results

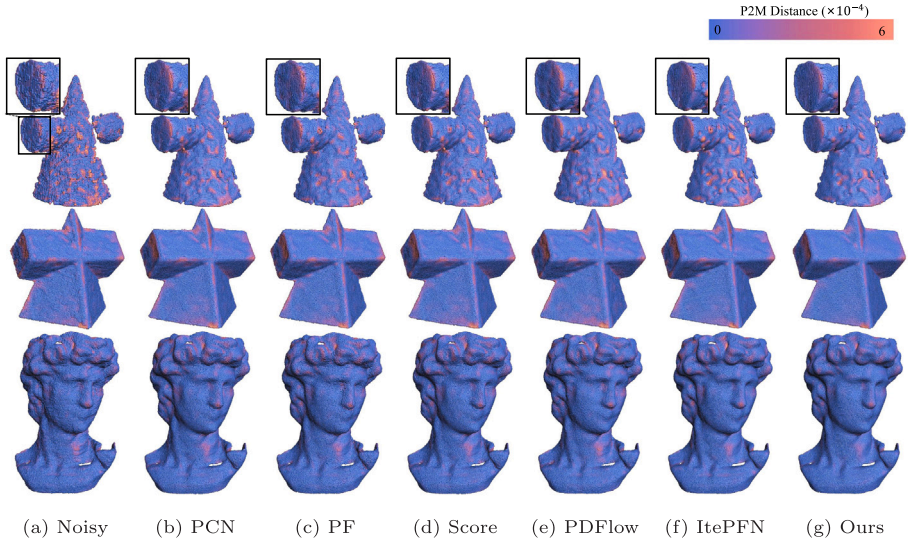

Fig. 6. Comparison of tested approaches of P2M real-scanned Kinect data. The top row shows denoised results of a single-view point cloud, and the bottom two rows demonstrate the results of multiple-view point clouds. Our method produces clearer results while preserving geometric details better.

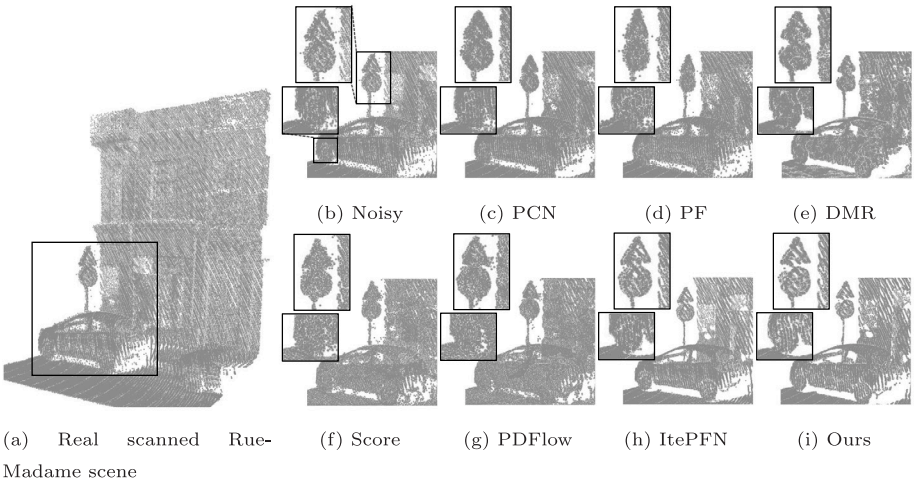

Fig. 7. Comparison results of the LiDAR data. The zoomed-in views highlight that our method removes heavy noise well and preserves shape structures better than the competing approaches.

while preserving geometric details better. We also tested our method on the outdoor LiDAR dataset [44], shown in Fig. 7. Unlike object shapes, city streets are corrupted by heavy noise and obtain more infrastructure structures. As we can see, our method removes heavy noise well and preserves shape structures better than those competing approaches.

The above quantitative comparisons show that PyramidPCD surpasses other competing approaches in maintaining multiscale characteristics while avoiding apparent artifacts.

4.4. Ablation stuides

We conduct ablated experiments on several variants of our network, as indicated in Table 4, to validate each module choice and the effectiveness of the internal design within modules. All ablated experiments are conducted on the PUNet dataset with 50K resolution shapes. Our ablation studies have four parts.

Encoder selection. To illustrate the effectiveness of our proposed bilateral feature aggregation (BFA) unit design, we removed the ST and DC modules and replaced our encoder with the popular PointNet++ [46] for comparison. As seen in Table 4 (a), the variant shows lower performance, indicating the effectiveness of our encoder design. Compared to the encoder in PointNet++ [46], our BFA effectively propagates information between the two feature spaces, enhancing the distinctiveness of encoder features.

Effect of ST unit. As seen in Table 4 (b), we conduct independent ablated tests to assess each design's contribution to our structure-aware Transformer (ST). When we remove the entire ST unit, we can observe a noticeable performance drop in CD and P2M values. Then, we remove the input features of ST from the coarse-scale encoder and decoder. As we can see, lacking any of these coarse-scale features reduces overall efficiency to some extent, demonstrating the importance of structural information for producing better results. We also validate the effect of our feature gate unit (FGU) in ST for feature fusion, as shown in the bottom row of Table 4 (b).

Effect of DC unit. To test the effect of our detail-preserving connection (DC) in the decoder, we remove DC and replace it with the skip connection. As shown in Table 4 (c), without DC, the values of CD and P2M drop directly. These results show the importance of our DC unit.

Depth choice. Table 4 (d) illustrates the impact of varying network depths on denoising results. As we can see, our analysis indicates optimal performance with a depth of five layers. A network with insufficient layers cannot adequately explore the geometric information

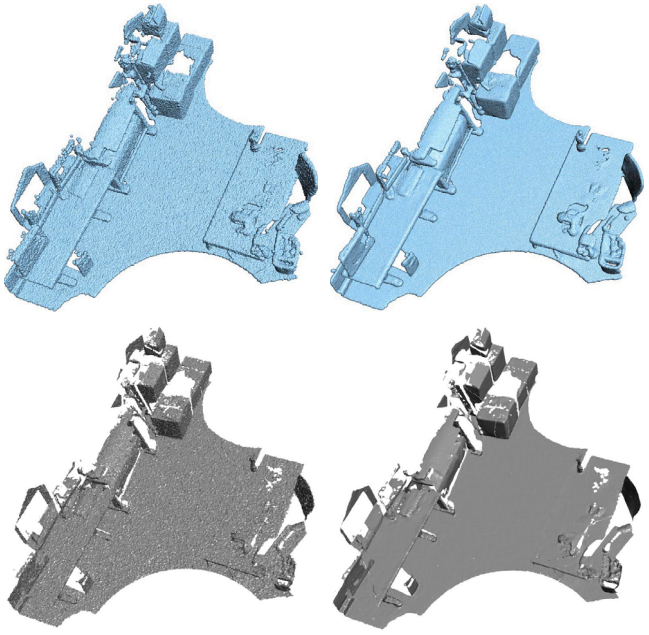

Fig. 8. Top: the LiDAR indoor scene (left) and its denoised result (right). Bottom: the reconstructed meshes of the raw data and our denoised result. The quality of the reconstructed surface is raised considerably after applying our denoising method.

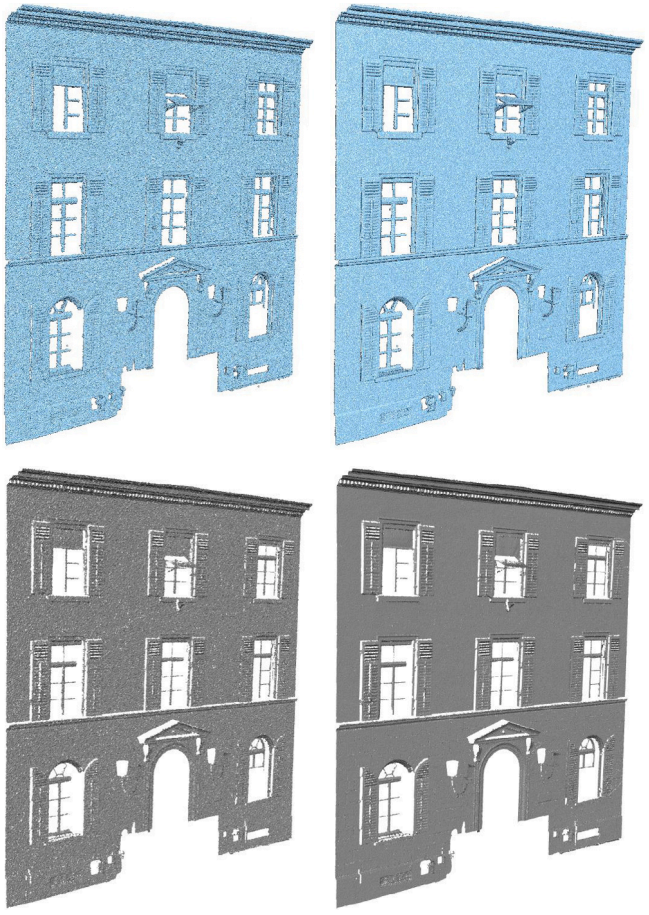

Fig. 9. Top: the LiDAR outdoor scene (left) and its denoised result (right). Bottom: the corresponding reconstructed meshes.

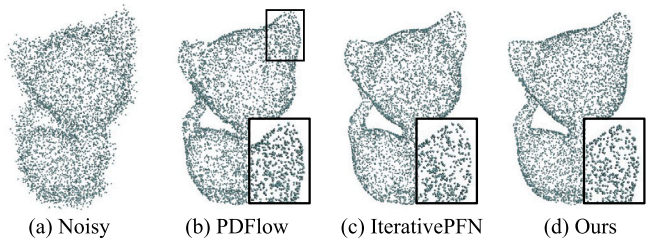

Fig. 10. Visual effects of Kitten shape with 5K resolution and 3% Gaussian noise. Filtering sparse point clouds under high noise conditions poses a significant challenge to our approach.

of the point clouds, failing to perceive multiscale characteristics. Conversely, an overly deep network produces oversized receptive fields, blurring important details and increasing unnecessary computational consumption.

4.5. Model complexity and efficiency

We evaluate the model complexity, including the total number of parameters, and the efficiency, considering inference time, of the tested methods on synthetic point clouds at 50K resolution. Results are detailed in Table 5. Although DMR [35] has the fewest parameters and highest efficiency, it produces results with higher error metrics. To enhance smoothing, Score [36], and PDFlow [38] might need several iterations, increasing their runtime. PF [16] and PCN [15] demonstrate poor efficiency due to their per-point smoothing strategy. In contrast, our method yields superior denoising performance and is highly competitive in model complexity and efficiency.

4.6. Generalization

We employ the LiDAR indoor and outdoor scenes to evaluate our method's performance further and demonstrate its generalization on unseen data. As we can see in Figs. 8 and 9, our method consistently produces satisfactory denoised results and significantly raises the reconstructed surface's quality. Intuitively, our method's denoised accuracy remains nearly impeccable, even in scenarios not encountered during training, proving its superiority in dealing with scene data. In addition, our denoising method considerably raises the quality of the reconstructed surface.

4.7. Limitations

Our approach has been shown to surpass typical existing denoising methods, yet it has limitations. Like most methods, it struggles with high sparse and noisy point clouds. To our knowledge, no current methods yield satisfactory results for such low-quality cases. Fig. 10 presents the denoising results of Kitten shape with 5K resolution and 3% Gaussian noise. It can be observed that under such high sparsity and high noise conditions, recovering geometric details is challenging. Additionally, our results appear superior and recover a nicer distribution to others.

5. Conclusion

We have presented PyramidPCD, a feature pyramid learning network for point cloud denoising. The proposed PyramidPCD has an updated U-Net architecture, based on the insight that more structures are kept in a feature pyramid when the scale is coarser and vice versa. Thanks to our improved U-Net architecture, our method can learn feature pyramid representations of varying scales with structure augmentation and detail preservation. To some extent, extremely

Table 4
Numerical results of ablation studies on PUNet dataset with 50K resolution shapes. The ablated experiments include core module design and depth choice of the feature pyramid.

Ablation		CD				P2M			
		1%	2%	3%	AVE	1%	2%	3%	AVE
(a)	Encoder of PointNet++	6.11	8.25	24.21	12.86	3.04	4.58	16.23	7.95
	Our encoder with BFA	6.03	8.09	22.01	12.04	2.97	4.41	14.92	7.43
(b)	w/o ST	6.02	8.08	21.49	11.86	3.0	4.38	14.55	7.30
	ST w/o coarse encoder	5.94	7.86	20.42	11.41	2.93	4.23	13.73	6.96
	ST w/o coarse decoder	5.98	8.03	21.84	11.95	2.95	4.34	14.61	7.30
	ST w/o FGU	5.93	7.83	17.97	10.58	2.93	4.25	11.9	6.36
(c)	w/o DC	5.98	7.89	18.01	10.63	2.97	4.28	11.96	6.41
(d)	4 layers	5.99	8.02	26.36	13.46	2.95	4.32	18.23	8.50
	6 layers	5.94	7.83	**17.81**	10.53	2.91	4.2	**11.76**	6.29
Default	BFA, w/ ST, DC, 5 layers	**5.89**	**7.78**	17.85	**10.51**	**2.89**	**4.17**	11.78	**6.28**

Table 5
Model complexity and efficiency comparison of our approach and the competing methods.

Methods	Params. (M)	Time (s)
PCN	29.7	158.7
PF	1.4	84.4
DMR	0.23	13.2
Score	0.21	21.6
PDFlow	0.49	59.0
IterativePFN	3.2	26.5
Ours	3.0	16.1

low quality point clouds will affect the final denoising results of our method. We thoroughly evaluate our method and compare it to competing approaches. Comprehensive studies show its benefits, and our method yields outstanding results on three benchmarks.

There is a wealth of opportunity for exploring future directions. For instance, the network architecture of PyramidPCD is expected to perform well in real-world environments with complex noise patterns. Moreover, our method is flexible for extension and effective for other point cloud processing tasks, such as semantic segmentation, object detection, and registration. Our method is potential to adapt to other 3D data types (e.g., triangular meshes) is noteworthy, which leads to improved noise reduction in mesh data while maintaining surface details.

CRediT authorship contribution statement

Zheng Liu: Writing – original draft, Project administration, Methodology, Conceptualization. **Weijie Zhou:** Visualization, Software, Data curation. **Chuchen Guo:** Visualization, Software, Investigation, Conceptualization. **Qinjun Qiu:** Writing – review & editing, Methodology, Conceptualization. **Zhong Xie:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this manuscript and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs.

Acknowledgments

This work was supported by National Key R&D Program of China (2022YFB3904100).

Data availability

No data was used for the research described in the article.

References

[1] Y. Zhou, R. Chen, Y. Zhao, X. Ai, G. Zhou, Point cloud denoising using non-local collaborative projections, Pattern Recognit. 120 (2021) 108128.
[2] Y. Li, H. Sheng, A single-stage point cloud cleaning network for outlier removal and denoising, Pattern Recognit. 138 (2023) 109366.
[3] J. Zhang, Z. Jiang, Q. Qiu, Z. Liu, TCFAP-Net: Transformer-based Cross-feature Fusion and Adaptive Perception Network for large-scale point cloud semantic segmentation, Pattern Recognit. 154 (2024) 110630.
[4] Y. Zhang, S. Yan, L. Zhang, B. Du, Fast projected fuzzy clustering with anchor guidance for multimodal remote sensing imagery, IEEE Trans. Image Process. 33 (2024) 4640–4653.
[5] Y. Zhang, G. Jiang, Z. Cai, Y. Zhou, Bipartite graph-based projected clustering with local region guidance for hyperspectral imagery, IEEE Trans. Multimedia 26 (2024) 9551–9563.
[6] A.C. Öztireli, G. Guennebaud, M. Gross, Feature preserving point set surfaces based on non-linear kernel regression, Comput. Graph. Forum 28 (2) (2009) 493–501.
[7] Z. Liu, X. Xiao, S. Zhong, W. Wang, Y. Li, L. Zhang, Z. Xie, A feature-preserving framework for point cloud denoising, Comput. Aided Des. 127 (2020) 102857.
[8] Y. Lipman, D. Cohen-Or, D. Levin, H. Tal-Ezer, Parameterization-free projection for geometry reconstruction, ACM Trans. Graph. 26 (3) (2007) 22.
[9] H. Huang, D. Li, H. Zhang, U. Ascher, D. Cohen-Or, Consolidation of unorganized point clouds for surface reconstruction, ACM Trans. Graph. 28 (5) (2009) 176:1–7.
[10] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, H.R. Zhang, Edge-aware point set resampling, ACM Trans. Graph. 32 (1) (2013) 9:1–9:12.
[11] H. Chen, M. Wei, Y. Sun, X. Xie, J. Wang, Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint, IEEE Trans. Vis. Comput. Graphics 26 (11) (2019) 3255–3270.
[12] X. Lu, S. Schaefer, J. Luo, L. Ma, Y. He, Low rank matrix approximation for 3D geometry filtering, IEEE Trans. Vis. Comput. Graphics 28 (4) (2020) 1835–1847.
[13] J. Wang, J. Jiang, X. Lu, M. Wang, Rethinking point cloud filtering: A non-local position based approach, Comput. Aided Des. 144 (2022) 103162.
[14] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 652–660.
[15] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N.J. Mitra, M. Ovsjanikov, PointCleanNet: Learning to denoise and remove outliers from dense point clouds, Comput. Graph. Forum 39 (1) (2020) 185–203.

[16] D. Zhang, X. Lu, H. Qin, Y. He, Pointfilter: Point cloud filtering via encoder-decoder modeling, IEEE Trans. Vis. Comput. Graphics 27 (3) (2021) 2015–2027.
[17] H. Chen, Z. Wei, X. Li, Y. Xu, M. Wei, J. Wang, RePCD-Net: Feature-aware recurrent point cloud denoising network, Int. J. Comput. Vis. 130 (3) (2022) 615–629.
[18] Z. Liu, Y. Zhao, S. Zhan, Y. Liu, R. Chen, Y. He, PCDNF: Revisiting learning-based point cloud denoising via joint normal filtering, IEEE Trans. Vis. Comput. Graphics (2023).
[19] D.d. Edirimuni, X. Lu, Z. Shao, G. Li, A. Robles-Kelly, Y. He, IterativePFN: True iterative point cloud filtering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 13530–13539.
[20] D.d. Edirimuni, X. Lu, G. Li, A. Robles-Kelly, Contrastive learning for joint normal estimation and point cloud filtering, IEEE Trans. Vis. Comput. Graphics (2023).
[21] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Trans. Graph. 38 (5) (2019) 1–12.
[22] G. Yang, F. Xue, Q. Zhang, K. Xie, C.-W. Fu, H. Huang, UrbanBIS: a large-scale benchmark for fine-grained urban building instance segmentation, ACM Trans. Graph. (SIGGRAPH) (2023) 1–11.
[23] Z. Wang, F. Luo, X. Long, W. Zhang, C. Xiao, Learning long-range information with dual-scale transformers for indoor scene completion, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 18523–18533.
[24] S. Li, G. Gao, Y. Liu, M.G. Yu-Shen Liu, GridFormer: Point-grid transformer for surface reconstruction, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2024, pp. 3163–3171.
[25] L. Zhou, G. Sun, Y. Li, W. Li, Z. Su, Point cloud denoising review: from classical to deep learning-based approaches, Graph. Models 121 (2022) 101140.
[26] X. Wang, W. Cui, R. Xiong, X. Fan, D. Zhao, FCNet: Learning noise-free features for point cloud denoising, IEEE Trans. Circuits Syst. Video Technol. 33 (11) (2023) 6288–6301.
[27] X. Wang, X. Fan, D. Zhao, PointFilterNet: A filtering network for point cloud denoising, IEEE Trans. Circuits Syst. Video Technol. 33 (3) (2023) 1276–1290.
[28] Z. Wei, H. Chen, L. Nan, J. Wang, J. Qin, M. Wei, PathNet: Path-selective point cloud denoising, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[29] D. Lu, X. Lu, Y. Sun, J. Wang, Deep feature-preserving normal estimation for point cloud filtering, Comput. Aided Des. 125 (2020) 102860.
[30] M. Wei, H. Chen, Y. Zhang, H. Xie, Y. Guo, J. Wang, GeoDualCNN: Geometry-supporting dual convolutional neural network for noisy point clouds, IEEE Trans. Vis. Comput. Graphics (2021).
[31] H. Zhou, H. Chen, Y. Zhang, M. Wei, H. Xie, J. Wang, T. Lu, J. Qin, X.-P. Zhang, Refine-net: Normal refinement neural network for noisy point clouds, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[32] Y. Zhang, M. Wei, L. Zhu, G. Shen, F.L. Wang, J. Qin, Q. Wang, Norest-net: Normal estimation neural network for 3-D noisy point clouds, IEEE Trans. Neural Netw. Learn. Syst. (2024).
[33] C. Yi, Z. Wei, J. Qiu, H. Chen, J. Wang, M. Wei, PN-Internet: Point-and-normal interactive network for noisy point clouds, IEEE Trans. Geosci. Remote Sens. (2024) 1.
[34] P. Hermosilla, T. Ritschel, T. Ropinski, Total denoising: Unsupervised learning of 3d point cloud cleaning, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2019, pp. 52–60.
[35] S. Luo, W. Hu, Differentiable manifold reconstruction for point cloud denoising, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2020, pp. 1330–1338.
[36] S. Luo, W. Hu, Score-based point cloud denoising, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2021, pp. 4583–4592.
[37] T. Zhao, P. Gao, T. Tian, J. Ma, J. Tian, From noise addition to denoising: A self-variation capture network for point cloud optimization, IEEE Trans. Vis. Comput. Graphics (2022).
[38] A. Mao, Z. Du, Y.-H. Wen, J. Xuan, Y.-J. Liu, PD-Flow: A point cloud denoising framework with normalizing flows, in: Proceedings of the European Conference on Computer Vision, ECCV, 2022, p. 13663.
[39] H. Chen, B. Du, S. Luo, W. Hu, Deep point set resampling via gradient fields, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2023) 2913–2930.
[40] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Randla-net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 11108–11117.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 30, 2017, pp. 6000–6010.
[42] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, P.-A. Heng, Pu-net: Point cloud upsampling network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 2790–2799.
[43] P.-S. Wang, Y. Liu, X. Tong, Mesh denoising via cascaded normal regression, ACM Trans. Graph. 35 (6) (2016) 1–12.
[44] A. Serna, B. Marcotegui, F. Goulette, J.-E. Deschaud, Paris-rue-madame database: a 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods, in: International Conference on Pattern Recognition Applications and Methods, ICPRAM, 2014, pp. 819–824.
[45] F. Pistilli, G. Fracastoro, D. Valsesia, E. Magli, Learning graph-convolutional representations for point cloud denoising, in: Proceedings of the European Conference on Computer Vision, ECCV, 2020, pp. 103–118.
[46] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in Neural Information Processing Systems (NeurIPS), Vol. 30, 2017, pp. 5105–5114.

Zheng Liu is currently an associate professor at China University of Geosciences (Wuhan). He received the Ph.D. degree from Central China Normal University in 2012. From 2013 to 2014, he held a post-doctoral position with School of Mathematical Sciences, University of Science and Technology of China. His research interests include geometry processing, computer graphics, 3D vision, and deep learning.

Weijie Zhou received the B.S. degree in Information and computing science from Chengdu University of Technology, Chengdu, China, in 2022 and is currently a M.S. candidate at China University of Geosciences, Wuhan, China. His research interests include 3D vision, 3D deep learning and computer graphics.

Chuchen Guo received the B.S. degree in Information and computing science from Chengdu University of Technology, Chengdu, China, in 2022 and is currently a M.S. candidate at China University of Geosciences, Wuhan, China. His research interests include 3D vision, 3D deep learning and computer graphics.

Qinjun Qiu received the B.S. degree in computer science and technology and the M.S. degree in computer applicated technology from the China Three Gorges University, Yichang, China, in 2011 and 2014, respectively, and the Ph.D. degree in geographic information engineering from the China University of Geosciences, Wuhan, China, in 2020. He is currently an Associate Professor at China University of Geosciences. His research interests include deep learning, text mining, and knowledge graph.

Zhong Xie received the B.S., M.S., and Ph.D. degrees in cartography and geographic information engineering from the China University of Geosciences (Wuhan), Wuhan, China, in 1990, 1998, and 2002, respectively. He is currently a Professor of China University of Geosciences (Wuhan). His research interests include deep learning, 3D rebuilding and spatial analysis, and digital twins.