PU-MFA: Point Cloud Up-Sampling via Multi-Scale Features Attention

Hyungjun Lee 1 and Sejoon Lim 2,*

1 Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Republic of Korea
2 Department of Automobile and IT Convergence, Kookmin University, Seoul 02707, Republic of Korea
* Correspondence: lim@kookmin.ac.kr; Tel.: +82-2-910-5469

Abstract: Recently, research using point clouds has been increasing with the development of 3D scanner technology. According to this trend, the demand for high-quality point clouds is increasing, but there is still a problem with the high cost of obtaining high-quality point clouds. Therefore, with the recent remarkable development of deep learning, point cloud up-sampling research, which uses deep learning to generate high-quality point clouds from low-quality point clouds, is one of the fields attracting considerable attention. This paper proposes a new point cloud up-sampling method called Point cloud Up-sampling via Multi-scale Features Attention (PU-MFA). Inspired by prior studies that reported good performance at generating high-quality dense point set using the multi-scale features or attention mechanisms, PU-MFA merges the two through a U-Net structure. In addition, PU-MFA adaptively uses multi-scale features to refine the global features effectively. The PU-MFA was compared with other state-of-the-art methods in various evaluation metrics through various experiments using the PU-GAN dataset, which is a synthetic point cloud dataset, and the KITTI dataset, which is the real-scanned point cloud dataset. In various experimental results, PU-MFA showed superior performance of generating high-quality dense point set in quantitative and qualitative evaluation compared to other state-of-the-art methods, proving the effectiveness of the proposed method. The attention map of PU-MFA was also visualized to show the effect of multi-scale features.

Keywords: 3D vision; deep-learning; point cloud; attention mechanism; point cloud up-sampling

1. Introduction

A point cloud is one of the most popular formats for accurately representing 3D geometric information in robotics and autonomous vehicles. Recently, the number of studies using point clouds has been increasing with the development of 3D scanners, such as LiDAR [1,2]. Along with this trend, there is an increasing demand for high-quality point clouds that are low-noise, uniform, and dense. However, the high cost of collecting high-quality point clouds remains problematic. Therefore, point cloud up-sampling, which generates a low-noise, uniform, and dense point set from noisy, non-uniform, and sparse point sets, is an interesting study.

Similar to learning-based image super-resolution studies [3,4], various learning-based point cloud up-sampling studies [5–7] show better performance of generating high-quality dense point set than traditional point cloud up-sampling studies [8,9]. Intuitively, the image super-resolution tasks and the point cloud up-sampling tasks are similar. Unlike the image super-resolution tasks, which process regular format images, the point cloud up-sampling tasks, which process irregular formats, require additional consideration. First, the up-sampled point set should have a uniform distribution and a dense set of points. Next, the up-sampled point set should represent the details of the target 3D mesh surface well [10].
A traditional learning-based point cloud up-sampling study usually consists of a feature extractor and an up-sampler. In addition, most studies use multi-scale features or attention mechanisms. PU-Net [11], 3PU [12], and PU-GCN [7] extract multi-scale features from sparse point sets. These studies have reported that they are excellent for generating dense point sets, but the last feature extracted by the feature extractor has a limitation in that the details of the sparse point set are diluted features because the output of each layer is used as the input of the next layer. Dis-PU [13], PU-EVA [6], and PU-Transformer [5] showed successful performance of generating high-quality dense point set using the self-attention mechanism to learn long-range dependencies between points. However, there is a limit to applying the attention mechanism with limited information because the key, query, and value of the self-attention mechanism are generated from the same input.

Focusing on these limitations, this paper proposes PU-MFA, a novel method to fuse multi-scale features and attention mechanisms. PU-MFA solves point cloud up-sampling through an attention mechanism that uses an adaptive feature for each layer. The contributions of this research are as follows:

- This paper proposes a point cloud up-sampling method of U-Net structure using Multi-scale Features (MFs) adaptively to Global Features (GFs).
- Global Context Refining Attention (GCRA), a structure for effectively combining MFs and attention mechanisms, is proposed. To the best of the authors’ knowledge, this is the first MultiHead Cross-Attention (MCA) mechanism proposed in point cloud up-sampling.
- This study demonstrates the effect of MFs by visualizing the attention map of GCRA in ablation studies.

This method was compared with various state-of-the-art methods using the Chamfer Distance (CD), Hausdorff Distance (HD), and Point-to-Surface (P2F) evaluation metrics for the PU-GAN [10] and the KITTI [14] dataset. As a result, the effectiveness of this method was confirmed by showing better performance at generating dense point set.

2. Related Work

2.1. Optimization-Based Point Cloud Up-Sampling

Various optimization-based studies have been performed to generate a dense set of points from a sparse set. Alexa et al. solved up-sampling by inserting new points into the Voronoi diagram of the local tangential space computed based on the moving-least-squares error [8]. Lipman et al. explained the up-sampling using the Locally Optimal Projection (LOP) operator [9]. In this study, the points were re-sampled by using $L_1$ norm. Huang et al. up-sampled a noisy and non-uniform set of points using an improved LOP that is a weighted LOP [15]. Later, Huang et al. proposed an advanced method called Edge-Aware Re-sampling of a set of points (EAR). The EAR first re-samples the edges and then uses edge-aware up-sampling to resolve the up-sampling [16].

2.2. Learning-Based Point Cloud Up-Sampling

With the successful performance of learning-based image super-resolution, many studies have proposed a learning-based point cloud up-sampling method.

As with image analysis, many studies have used MFs in point cloud up-sampling. PU-Net [11], the first attempt at deep learning for point cloud up-sampling, showed good performance at generating high-quality point set by extracting MFs through hierarchical feature learning and interpolation based on the framework of PointNet++ [17]. 3PU [12] performed well using MFs via an Intra-Level Dense connection and Inter-Level Skip connection. PU-GCN [7] uses MFs extracted by Inception DenseGCN. In this study, Inception DenseGCN could effectively extract MFs with an InceptionNet-inspired structure [18].
Because of the advantages of learning the long-range dependency of the self-attention mechanism, it is used in various point cloud up-sampling studies. In PU-GAN [10], generators are trained using discriminators that apply a self-attention mechanism. PugeoNet [19] showed good performance at generating high-quality dense point set by using it in Feature Recalibration. PU-EVA [6] showed successfully generating up-sampled point set using an EVA Expansion Unit with the mechanism. Dis-PU [13] performed well using the Local Refinement Unit with self-attention applied to the generated point set. PU-Transformer [5], which applied the transformer structure for the first time in point cloud up-sampling, uses Shifted Channel MultiHead Self-Attention to show the state-of-the-art performance of generating high-quality point set.

3. Problem Description

Given an unordered sparse point set \( S = \{s_i\}_{i=1}^{N} \) of \( N \) samples, we aim to generate low-noise, uniform, and dense point set \( Q = \{q_i\}_{i=1}^{rN} \) using \( D = \{d_i\}_{i=1}^{rN} \) as Ground Truth (GT), where \( N \) is the input patch size and \( r \) is the up-sampling ratio. Figure 1 shows the problem description of this study. Also, Table 1 summarizes the definitions of symbols to be used.

Table 1. Description of symbols.

| Symbol | Description |
|--------|-------------|
| \( S \) | Sparse point set |
| \( s_i \) | Element of \( S \) |
| \( S' \) | Offset of \( S \) |
| \( s'_i \) | Element of \( S' \) |
| \( D \) | Ground truth point set |
| \( d_i \) | Element of \( D \) |
| \( Q' \) | Coarse point set |
| \( q'_i \) | Element of \( Q' \) |
| \( Q'_{:\Delta} \) | Offset of \( Q' \) |
| \( q'_{i\Delta} \) | Element of \( Q'_{:\Delta} \) |
| \( Q \) | Dense point set |
| \( q_i \) | Element of \( Q \) |
| \( N \) | Input patch size |
| \( r \) | Up-sampling ratio |
| \( H \) | Depth of layer |
| \( F_h \) | Set of point wise feature extracted from \( h^{th} \) Point Transformer |
| \( f_h^i \) | Point-wise feature extracted from \( h^{th} \) Point Transformer |
| \( C \) | Channel |
| \( K, K' \) | Expansion rate |
| \( patch_i \) | Patch created through KNN based on \( s_i \) |
| \( patch\_size \) | Neighbor size of KNN |
4. Method

This method consists of a Multi-scale Feature Extractor (MFE), Global Context Refiner (GCR), Coarse Point Generator (CPG), and Self-Attention Block (SAB). As shown in Figure 2, MFE extracts MFs, an adaptive feature for use in GCR. GCR uses MFs to refine GFs adaptively and finally produce $Q^A$, where $Q^A$ is defined as $Q^A = \{q^A_i\}_{i=1}^N$. CPG generates $Q'$ from $S$ and SAB extracts GFs from $Q'$, where $Q'$ is defined as $Q' = \{q'_i\}_{i=1}^N$. Based on the definitions of $Q'$ and $Q^A$, $Q$ is formulated as Equation (1), where $\oplus$ is an element-wise sum.

$$Q = Q' \oplus Q^A \quad (\oplus : \text{element-wise sum}) \quad (1)$$
4.1. Multi-Scale Feature Extractor

Because the GFs extracted from $Q'$ via SAB is a feature extracted from a set of points in which geometric information about the original input $S$ is diluted, MFE using Point Transformer (PT) [20], an advanced point cloud analysis technique, extracts MFs from $S$. As shown in Figure 2, the MFE consists of $H$ PT, and the set of point-wise features extracted from the $h^{th}$ PT is $F_h \in \mathbb{R}^{N \times K^{h-1} \times C}$. MFs are the set of $F_h$ extracted from all layers of the MFE. The extracted MFs and $F_h$ are formulated as in Equation (2), where $f_i^h$ is a point-wise feature extracted from the $h^{th}$ PT.

$$F_h = \{f_i^h\}_{i=1}^N, \text{MFs} = \{F_h\}_{h=1}^H$$

Point Transformer

PT consists of two elements. The first is the K-Nearest Neighbor (KNN), and the second is the Vector Self-Attention (VSA) mechanism. At the $h^{th}$ PT, the point-wise feature $f_i^{h-1}$ of point $s_i \in S$ is updated to $f_i^h$ through VSA, which uses $s_i$ and patch$_i$ as the inputs. The patch$_i$ is generated through KNN using $s_i$ as the input. This operation works on all points in $S$, updating the point-wise feature of all points [20]. This is formulated in Equation (3), where $\text{patch}_{\text{size}}$ is the size of KNN’s neighbor size.

$$\text{patch}_i = \text{KNN}(s_i, \text{patch}_{\text{size}})$$
$$f_i^h = \sum_{p_k \in \text{patch}_i} \text{VectorSelfAttention}(s_i, p_k)$$

(3)

Inspired by this operation, this considered patch$_i$ is equivalent to the CNN’s kernel. In CNN, even if the kernel of the CNN is fixed, a deeper layer, means a wider receptive field. Therefore, even if the patch size of the KNN in PT is fixed, the deeper the layer, the more $s_i$ can interact with a wider range of points. Figure 3 is an example with a KNN patch size of four. In Figure 3a, when the $h^{th}$ PT updates $f_i^{h-1}$ to $f_i^h$, VSA is performed on patch$_i$, which is composed of $s_i$ and incidental points, to update $f_i^{h-1}$. In Figure 3b, the $h^{th}$ PT updates each feature by performing a VSA for each patch in all incidental points. In Figure 3c, the $(h+1)^{th}$ PT updates $f_i^h$ to $f_i^{h+1}$ by performing VSA using patch$_i$ similar to the $h^{th}$ PT. However, the $(h+1)^{th}$ PT updates $f_i^h$ to $f_i^{h+1}$ using a wider receptive field than the receptive field of the $h^{th}$ PT because the features of the incidental points of the $(h+1)^{th}$ PT are updated by the $h^{th}$ PT. This operation allows the MFE to extract the MFs effectively.

Figure 3. Illustration of KNN and VSA in PT. (a) $h^{th}$ Point Transformer layer with $s_i$. (b) $h^{th}$ Point Transformer layer with incidental points. (c) $(h+1)^{th}$ Point Transformer layer with $s_i$. 
4.2. Global Context Refiner

Because GCR and MFE are U-Net [21] structures, they are composed of H GCRA.
GCRA effectively refines GFs by querying MFs, which is adaptive geometric information
applied to each layer. As shown in Figure 2, the \((H − h + 1)^{th}\) GCRA generates
\(RGC_{H−h+1} \in \mathbb{R}^{N \times k^{h−2} C}\) by using \(F_p \in \mathbb{R}^{N \times k^{h−1} C}\) as a query and
\(RGC_{H−h}\) as a pool. However, \(\mathbb{R}^{N \times r^3}\) was used instead to prevent \(RGC_H\) from becoming \(\mathbb{R}^{N \times \tilde{F}}\). After refining the
GFs, the linear layer was used to perform the transformation. PixelShuffle was then used
to generate the \(Q^\Delta \in \mathbb{R}^{N \times 3}\).

Global Context Refining Attention

Inspired by Skip-Attention [22], which acts as a communicator between the encoder
and decoder, GCRA uses MFs and GFs to apply the MCA mechanism. In various studies,
self-attention mechanisms are used to extract the features of point sets or to generate up-
sampling point sets [5,10]. However, the self-attention mechanism is limited because it uses
only limited information due to the structure in which key, query, and value are generated
from the same input. With these limitations in mind, GCRA in the \(H\) hierarchy uses
GFs \(\in \mathbb{R}^{N \times K'C'}\) as the pool (key, values) and MFs as the queries, progressively refining the
GFs through MCA. GCRA consists of MCA [23], Batch Normalization (BN) [24], and Feed
Forward. As shown in Figure 4, the output shape of applying MCA using the query and
pool is \(\mathbb{R}^{N \times F_p}\). The pool was then refined by adding the pool and the MCA output. The BN
was used for stable training after addition. Feed Forward transforms the output of the BN
and produces a Refined Global Context (RGC) \(\in \mathbb{R}^{N \times F_o}\).

![Figure 4. Illustration of Global Context Refining Attention (GCRA). \(F_p\) is the pool input channel, \(F_q\)
is the query input channel, and \(F_o\) is the output channel.](image)

4.3. Coarse Point Generator

CPG generates \(Q'\). In CPG, PT [20] and PixelShuffle [5,25] generate \(S'^\Delta\) from \(S\),
where, \(S'^\Delta\) is defined as \(S'^\Delta = \{s'^\Delta\}_{j=1}^{N}\). The structure of CPG consists of four layers,
such as the structure of the 3PU’s Feature Extraction Unit [12]. As shown in Figure 2,
to make the final output into 3D coordinates, first, PT was first used to expand the features,
and then gradually reduce them. Subsequently, PixelShuffle generates 3D coordinates
using those features. \(Q'\) is generated through the element-wise sum of the generated \(S'^\Delta\)
and \(\text{duplicate}(S,r) \in \mathbb{R}^{N \times 3}\). This process is formulated as Equation (4).

\[
duplicate(S,r) = \left\{ r \times \begin{array}{c} s_j \end{array} \right\}_{j=1}^{N} \\
Q' = \duplicate(S,r) \oplus S'^\Delta
\]

4.4. Self-Attention Block

Inspired by self-attention, which learns long-range dependency [23], we use Multi-
Head Self-Attention (MSA) was used to extract the GFs from \(Q'\). As shown in Figure 2,
the shape of \(Q'\) was changed from \(\mathbb{R}^{N \times 3}\) to \(\mathbb{R}^{N \times 3r}\), and the coordinates of \(Q'\) were used
as features of the original point set $S$. The GFs $\mathbb{R}^{N \times K'C'}$ was then extracted using the changed shape $Q'$ as the input to the MSA.

5. Experimental Settings

5.1. Datasets

All methods were trained using the most popular PU-GAN [10] dataset in these experiments and evaluated using the PU-GAN dataset and the KITTI [14] dataset. The PU-GAN dataset was a synthetic point cloud dataset produced from 147 3D meshes, and the KITTI dataset was a real-scanned point cloud dataset collected using real LiDAR.

The training phase used 120 3D meshes from the PU-GAN dataset. All patches were generated via the Poisson disk sampling after converting the original mesh to a point cloud, just like the patch-based up-sampling approach. The sampling resulted in 24,000 input-output pairs.

In the evaluation phase, 27 3D meshes from the PU-GAN dataset were converted into point clouds to test the synthetic point up-sampling, and the real-scanned point up-sampling test was performed using the KITTI dataset. The generated patches should cover all point sets when evaluating the synthetic point cloud and real-scanned point cloud up-sampling. After merging each up-sampled patch, the up-sampled point set was reconstructed by farthest point sampling. More details can be found at study in PU-GAN [10]. This dataset was downloaded and used from https://github.com/liruihui/PU-GAN (accessed date: 12 July 2022).

5.2. Loss Function

In most point cloud reconstruction methods, CD is used as the loss function [22,26,27]. However, it was confirmed empirically that the Using Density-Aware Chamfer Distance as loss function showed good performance at point cloud reconstruction, considering the uniformity of the points set on the CD [28]. Therefore, the total loss was formulated as Equation (5), where $\alpha$ is linearly interpolated from 0.1 to 1 during training and $\| \cdot \|_2$ is $L_2$ norm.

$$\text{Loss}(Q', Q, D) = L_{CD}(Q', D) + \alpha \times L_{DCD}(Q, D)$$

$$L_{CD}(Q', D) = \frac{1}{|Q'|} \sum_{x \in Q'} \min_{y \in D} \| x - y \|_2 + \frac{1}{|D|} \sum_{y \in D} \min_{x \in Q'} \| y - x \|_2$$

$$L_{DCD}(Q, D) = \frac{1}{|Q|} \sum_{x \in Q} \min_{y \in D} (1 - e^{-\| x - y \|_2}) + \frac{1}{|D|} \sum_{y \in D} \min_{x \in Q} (1 - e^{-\| y - x \|_2})$$

(5)

5.3. Metric

This study evaluated the method using CD, HD, and P2F metrics, as in prior studies [5,6,13]. CD is a metric that measures the similarity between a set of GT points and a set of predicted points for each point, and HD is an evaluation metric that measures the outliers in a set of predicted points based on a set of GT points. P2F is an index that measures the similarity between the original mesh and the predicted point set and measures the quality of the predicted point set. The parameter complexity was also measured by measuring the number of parameters. For all metrics, a lower the number, meant better performance.

5.4. Comparison Methods

The proposed method was compared with three state-of-the-art methods: Dis-pu [13], PU-EVA [6], and PU-Transformer [5] to validate the method. For an exact comparison, all methods were implemented using pytorch [29] version 1.7.0 on Ubuntu 20.04 and trained on the same Intel i9-10980XE CPU and NVIDIA TITAN RTX environment.
5.5. Implementation Details

All methods for the experiment were trained with a batch size of 64 for 100 epochs, and the Adam [30] optimizer with a learning rate of 0.0001 was used. The patch size of KNN used in PT is set to 20 as in PU-Transformer [5]. Rotation, scaling, random perturbation, and regularization were applied to the training dataset, as in prior studies [10,11]. The upsampling ratio \( r \) was four and the input patch size \( N \) was 256. The CPG’s \( C' \) and \( K' \) were 32 and 8, respectively. For MFE and GCR, \( C \) and \( K \) were 16 and 4, respectively. The layer depth of MFE and GCR, \( H \), was four. The head number of MCA and MSA was set to eight, as in the prior study [23]. Here, the head is used to learn different perspectives in Multihead Attention.

6. Experimental Results

Dis-PU [13], PU-EVA [6], and PU-Transformer [5], and the present method were compared using the PU-GAN [10] and the KITTI [14] datasets.

6.1. Results on 3D Synthetic Datasets

Table 2 lists the quantitative performance comparisons for \( \times 4 \) and \( \times 16 \) up-sampling. \( \times 4 \) up-sampling sampled 2048 points to 8192 points. \( \times 16 \) up-sampling sampled 512 points to 8192 points by repeating the \( \times 4 \) up-sampling twice. As shown in Table 2, the present method showed good performance of generating high-quality point set compared to the other state-of-the-art methods. Presented method has the best value in the evaluation metric compared to other methods with similar parameter complexity. As shown in Table 3, the time complexity of the proposed method is similar to that of other methods.

Figures 5 and 6 present the visualization result of \( \times 4 \) up-sampling, and Figure 7 is the visualization result of \( \times 16 \) up-sampling. Figures 5b–d, show a set of points representing a tubular object, such as a bird’s leg, the space between the kitten’s body and tail, a statue’s leg, and a camel’s hoof with unclear boundaries. However, Figure 5e shows low-noise and clear boundaries. Also, Figures 6b–d, show the set of points representing non-tubular objects with noise the LP rear control cover and star. However, Figure 6e shows low noise in non-tubular objects.

In Figure 7b,d, the chair back does not represent the original shape well, and Figure 7c maintains the shape to some extent, but there is considerable noise. On the other hand, Figure 7e has relatively little noise and represents the original shape well.

Table 2. Comparing the quantitative evaluation of \( \times 4 \) and \( \times 16 \) up-sampling with the state-of-the-art methods.

| Method            | \( \times 4 \) (2048 → 8192) | \( \times 16 \) (512 → 8192) |
|-------------------|-----------------------------|------------------------------|
|                   | CD (10^{-3}) | HD (10^{-3}) | P2F (10^{-3}) | #Params (M) | CD (10^{-3}) | HD (10^{-3}) | P2F (10^{-3}) | #Params (M) |
| Dis-PU            | 0.2703      | 5.501        | 4.346         | 2.115       | 1.341       | 28.47        | 20.68         | 2.115       |
| PU-EVA            | 0.2969      | 4.839        | 5.103         | 2.198       | 0.8662      | 14.54        | 15.54         | 2.198       |
| PU-Transformer    | 0.2671      | 3.112        | 4.202         | 2.202       | 1.034       | 21.61        | 17.56         | 2.202       |
| PU-MFA (Ours)     | 0.2326      | 1.094        | 2.545         | 2.172       | 0.5010      | 5.414        | 9.111         | 2.172       |

Table 3. Measure average time complexity after 50 measurements on \( \times 4 \) up-sampling.

| Method            | Time per Batch (sec/batch) |
|-------------------|---------------------------|
| Dis-PU            | 0.02659                   |
| PU-EVA            | 0.02360                   |
| PU-Transformer    | 0.02244                   |
| PU-MFA (Ours)     | 0.02331                   |
Figure 5. Visualization result of $\times 4$ up-sampling on PU-GAN dataset (tubular objects).
6.2. Results on Real-Scanned Datasets

Dis-PU, PU-EVA, PU-Transformer, and the present method were evaluated using the KITTI dataset for ×4 up-sampling. Figure 8 shows ×4 up-sampling. In Figure 8b–d, the boundary between the window and the door of the vehicle was unclear. However, Figure 8e generated by the present method, showed that the boundary was clearer.
6.3. Ablation Study

This method was evaluated by performing various ablation studies using the PU-GAN dataset.

6.3.1. Effect of Components

To demonstrate the effectiveness of the contribution, four cases were divided into ablation studies. The cases were as follows: Case 1 was a structure using GCR, CPG, and SAB, with the MultiHead Attention (MHA) of GCR and SAB consisting of self-attention with one head. Case 2 was a structure changed from Case 1 to eight heads. Case 3 was a structure using GCRA composed of MCA by adding MFE to Case 2, where the query of all GCRA becomes \( F_q \), the final output of MFE. Case 4 was PU-MFA. As shown in Table 4, all contributions affected the method performance of generating point set.

Table 4. Ablation study results to analyze the effect of the present contribution.

| Case | Contribution | Metric |
|------|--------------|--------|
|      | MHA | MFE | MFs | CD (10^{-3}) | HD (10^{-3}) | P2F (10^{-3}) |
| 1    |     |     |     | 0.3349      | 4.461       | 4.926        |
| 2    | √   |     |     | 0.2473      | 1.101       | 2.829        |
| 3    | √   | √   |     | 0.2500      | 2.735       | 2.737        |
| 4    | √   | √   | √   | 0.2362      | 1.094       | 2.545        |

6.3.2. Multi-Scale Features Attention Analysis

By visualizing the attention maps of all GCRAs, it was confirmed that the GCRAs of GCR with \( H = 4 \) refined the GFs by adaptively using the MFs extracted from receptive fields of various sizes. Figure 9 shows the results visualized by choosing three attention heads in the GCRA and selecting 30 points, which had the highest attention score in \( S \), from each head. The attention map was visualized using Case 3 in Table 4 without MFs in Figure 9b to compare that MFs operated adaptively. As shown in Figure 9a, in the low-layer GCRA, an attention map was formed for a wide range of points in a point set, and in high-layer GCRA, an attention map was formed for a relatively narrow range of points. On the other hand, in Figure 9b, a wide range of attention maps was formed regardless of the high and low levels of the hierarchy. This phenomenon confirmed that PU-MFA uses the adaptive point feature for each layer of the GCRA.
Figure 9. Visualization of attention map generated using MFs as a query in GCR with $H = 4$.

Figure 10. Visualization result of the effect of noise.

6.3.3. Effect of Noise

Table 5 lists the $\times 4$ up-sampling results of Dis-PU [13], PU-EVA [6], PU-Transformer [5], and the present method using the PU-GAN dataset with various noises added. The noise effect evaluated the result obtained by adding different levels of Gaussian noise $\mathcal{N}(0, \text{noise level})$ to a set of input points. As shown in Table 5, the proposed method showed the most robustness to various noise levels. As shown in Figure 10, it can be seen that the boundary between the fingers blurred in the dense set of points generated by the state-of-the-art methods as the noise level was increased. On the other hand, the proposed method showed that the boundary between the fingers was maintained in the dense set of points generated by the present method.
Table 5. Quantitative evaluation results of the noise effects using the PU-GAN dataset.

| Method         | Various Noise Levels Test at $\times 4$ Up-Sampling (CD with $10^{-3}$) |
|----------------|-------------------------------------------------------------------------|
|                | 0            | 0.001         | 0.005         | 0.01          | 0.015         | 0.02          |
| Dis-PU         | 0.2703       | 0.2751        | 0.2975        | 0.3257        | 0.3466        | 0.3706        |
| PU-EVA         | 0.2969       | 0.2991        | 0.3084        | 0.3167        | 0.3203        | 0.3268        |
| PU-Transformer | 0.2671       | 0.2717        | 0.2905        | 0.3134        | 0.3331        | 0.3585        |
| PU-MFA (Ours)  | 0.2326       | 0.2376        | 0.2547        | 0.2764        | 0.2989        | 0.3195        |

7. Conclusions

In this paper, we proposed PU-MFA, a point cloud up-sampling method of U-Net structure that combines multi-scale features and attention mechanism. One of the most significant differences from the prior point cloud up-sampling methods was that PU-MFA used multi-scale features adaptively and effectively through fusion with the cross-attention mechanism. Also, the PU-MFA is the first method to apply the cross-attention mechanism to point cloud up-sampling to the best of the authors’ knowledge. Various experiments were performed on PU-MFA and other state-of-the-art methods using the PU-GAN and the KITTI dataset. As a result, PU-MFA showed better performance of generating high-quality dense point set than other state-of-the-art methods in various experiments. In addition, ablation study showed that multi-scale features are very useful in PU-MFA for generating high-quality point sets by choosing receptive field size adaptively for each layer.

Despite the successful performance at generating high-quality dense point set of PU-MFA, PU-MFA cannot cope with an arbitrary up-sampling ratio. Because PU-MFA is a patch-based up-sampling of $\times 4$, up-sampling is only possible for 4 to the $M$ power. A method that can respond to an arbitrary up-sampling ratio is planned in the future to overcome this limitation.

Author Contributions: Conceptualization, S.L.; methodology, H.L.; software, H.L.; validation, S.L.; formal analysis, H.L.; investigation, H.L. and S.L.; resources, H.L. and S.L.; data curation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, S.L.; visualization, H.L.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Institute of Police Technology (KIPoT) grant funded by the Korea government (KNFA)(No.92021C26503000), Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2022R1F1A1072626), and the BK21 Program(5199990814084) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/liruihui/PU-GAN (accessed date: 12 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Koide, K.; Miura, J.; Menegatti, E. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *Int. J. Adv. Robot. Syst.* 2019, 16. [CrossRef]
2. Lim, H.; Yeon, S.; Ryu, S.; Lee, Y.; Kim, Y.; Yun, J.; Jung, E.; Lee, D.; Myung, H. A Single Correspondence Is Enough: Robust Global Registration to Avoid Degeneracy in Urban Environments. *arXiv* 2022, arXiv:2203.06612.
3. Niu, B.; Wen, W.; Ren, W.; Zhang, X.; Yang, L.; Wang, S.; Zhang, K.; Cao, X.; Shen, H. Single image super-resolution via a holistic attention network. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 191–207.
4. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. Swinir: Image restoration using swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 1833–1844.
5. Qiu, S.; Anwar, S.; Barnes, N. PU-Transformer: Point Cloud Upsampling Transformer. *arXiv* 2021, arXiv:2111.12242.
6. Luo, L.; Tang, L.; Zhou, W.; Wang, S.; Yang, Z.X. PU-EVA: An Edge-Vector Based Approximation Solution for Flexible-Scale Point Cloud Upsampling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 16208–16217.

7. Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; Ghanem, B. Pu-gcn: Point cloud upsampling using graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11683–11692.

8. Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T. Computing and rendering point set surfaces. IEEE Trans. Vis. Comput. Graph. 2003, 9, 3–15. [CrossRef]

9. Lipman, Y.; Cohen-Or, D.; Levin, D.; Tal-Ezer, H. Parameterization-free projection for geometry reconstruction. ACM Trans. Graph. (TOG) 2007, 26, 22-es. [CrossRef]

10. Li, R.; Li, X.;Fu, C.W.; Cohen-Or, D.; Heng, P.A. Pu-gan: A point cloud upsampling adversarial network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 7203–7212.

11. Yu, L.; Li, X.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. Pu-net: Point cloud upsampling network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2790–2799.

12. Yifan, W.; Wu, S.; Huang, H.; Cohen-Or, D.; Sorkine-Hornung, O. Patch-based progressive 3d point set upsampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5958–5967.

13. Li, R.; Li, X.; Heng, P.A.; Fu, C.W. Point cloud upsampling via disentangled refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 344–353.

14. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. Int. J. Robot. Res. 2013, 32, 1231–1237. [CrossRef]

15. Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of unorganized point clouds for surface reconstruction. ACM Trans. Graph. (TOG) 2009, 28, 1–7. [CrossRef]

16. Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H. Edge-aware point set resampling. ACM Trans. Graph. (TOG) 2013, 32, 1–12. [CrossRef]

17. Qi, C.R.; Li, Y.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 2017, 30; pp. 5105–5114.

18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

19. Qian, Y.; Hou, J.; Kwong, S.; He, Y. PUGeo-Net: A geometry-centric network for 3D point cloud upsampling. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2020; pp. 752–769.

20. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 16259–16268.

21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.

22. Wen, X.; Li, T.; Han, Z.; Liu, Y.S. Point cloud completion by skip-attention network with hierarchical folding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1939–1948.

23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. Adv. Neural Inf. Process. Syst. 2017, 30; pp. 1936–1945.

24. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning; PMLR: Lille, France, 2015; pp. 448–456.

25. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.

26. Nguyen, A.D.; Choi, S.; Kim, W.; Lee, S. Graphx-convolution for point cloud deformation in 2d-to-3d conversion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8628–8637.

27. Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. Pointir: Diverse point cloud completion with geometry-aware transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 12498–12507.

28. Wu, T.; Pan, L.; Zhang, J.; Wang, T.; Liu, Z.; Lin, D. Density-aware chamfer distance as a comprehensive metric for point cloud completion. arXiv 2021, arXiv:2112.12702.

29. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2019; pp. 8024–8035.

30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.