Point Cloud Sampling Preserving Local Geometry for Surface Reconstruction

Kohei Matsuzaki
ko-matsuzaki@kddi.com
Keisuke Nonaka
ki-nonaka@kddi.com

KDDI Research, Inc.
Saitama, Japan

Abstract

Surface reconstruction from point clouds is a fundamental task in computer vision and graphics. Recent methods learn neural fields as a surface representation from point clouds. However, these methods are difficult to scale to large scenes due to the limited size of the point clouds they can handle. In this paper, we propose a point cloud sampling method to improve scalability for training a surface reconstruction network. We train the surface reconstruction network with sampled point clouds obtained from a sampling network. In the sampling network, we introduce a seed point that serves as the origin to sample point clouds from partial regions. It encourages the surface reconstruction network to learn both the global structure and local geometry on a part of the scene. We also introduce a split-and-merge approach to avoid increasing the memory footprint by suppressing the input size to the sampling network. Experimental results on ScanNet dataset show that the proposed method significantly improves surface reconstruction performance compared with state-of-the-art methods.

1 Introduction

Reconstruction of object or scene surfaces from three-dimensional (3D) point clouds is a fundamental task in computer vision and graphics. It is essential to drive numerous practical applications such as virtual/augmented reality, autonomous navigation, and computer animation. Traditional methods have tackled the task through mathematical optimization with specific geometric priors based on triangulation [1, 7], smoothness [16, 17], and template [21, 29]. On the other hand, the learning-based methods [11, 23, 39] worked on explicitly reconstructing the surfaces from the point clouds. Recently, the representation of surfaces as neural fields has attracted a great deal of attention due to their fidelity and flexibility [2, 4, 15, 24, 25, 30, 33]. These methods learn the neural fields that can extract zero-level surfaces by mapping the positions of 3D points to binary occupancy values [25] or signed distance values [30]. The neural fields being continuous representations allow the reconstructed surfaces to represent high-resolution geometry of arbitrary topology.

The representative methods [25, 30] with neural fields encode the entire 3D shape into a global feature. These methods are limited to a single object since they cannot represent local geometry well. Aiming at better scalability for large scenes, many subsequent methods have been proposed. Several methods [33, 37] divide the 3D space into regular grids and encode the input shape into a feature grid with a grid-based convolution. Also, several methods [3, 15] divide the input shape into multiple partial shapes and encode them individually. A more recent method [2] encodes each input point with a point convolution to preserve direct connections between features and points. However, this method still has limited scalability since memory requirements increase in proportion to the size of the input point clouds.

Point cloud sampling before performing a target task is widely used in order to reduce the input size in the pre-processing step. The most common methods are random sampling and farthest point sampling [10, 27]. Point clouds sampled by these methods lose local geometry details since their points are sampled in a spatially uniform manner. Although other sampling methods [5, 12, 35] focus on specific properties such as edges, contours, and curvature to achieve sampling that preserves local geometry details, these methods do not consider the downstream tasks. Several learning-based methods [9, 19, 41] sample point clouds optimized for a downstream task by introducing a task loss, measured with the sampled point cloud. However, these methods mainly focused on improving computational efficiency for inference, with little exploration of better scalability for training.

In this paper, we propose a learning-based point cloud sampling method to improve scalability for training a surface reconstruction network based on neural fields. We train the surface reconstruction network with sampled point clouds obtained from a sampling network. The sampling network aims to sample point clouds that encourage the surface reconstruction network to learn both the global structure and local geometry on a part of the scene. To this end, we introduce a seed point that serves as the origin for sampling from partial regions. The sampling network samples points using features weighted by distances from the seed point. We also design loss functions regarding point sampling modified in a probabilistic manner to make the sampling operation differentiable. This makes it possible for the proposed method to sample both points around the seed point and points far from it during training. The sampled point cloud does not necessarily preserve the entire input shape and may represent a partial shape. However, uniformly selected seed points cover most regions of the input shape throughout the training. This partial sampling of the proposed method is beneficial for scaling the training to large scenes, even when the input size that can be handled is limited. Furthermore, we also propose a split-and-merge approach to avoid increasing the memory footprint for sampling. This approach suppresses the input size to the sampling network by splitting the input point cloud, and then merging the sampling results to construct the final sampled point cloud. As a result, the proposed method achieves better performance in terms of surface reconstruction compared to the previous sampling methods.

The main contributions of this paper can be summarized as follows:

- We propose a novel method to learn neural fields as a 3D surface representation using point clouds sampled with a learnable sampling network. To the best of our knowledge, our work is the first to propose learning-based sampling to improve the scalability of surface reconstruction with neural fields.
- We propose a sampling network considering a seed point to sample points that represent both global structure and local geometry on a part of the scene. With loss functions modified in a probabilistic manner, the sampling network can sample both points around the seed point and points far from it.
- We introduce a split-and-merge approach that suppresses the input size fed into the sampling network in order to avoid increasing the memory footprint.
- We experimentally show that the proposed method samples more effective point clouds to improve surface reconstruction performance compared to state-of-the-art methods.

2 Related Work

2.1 Neural Surface Reconstruction

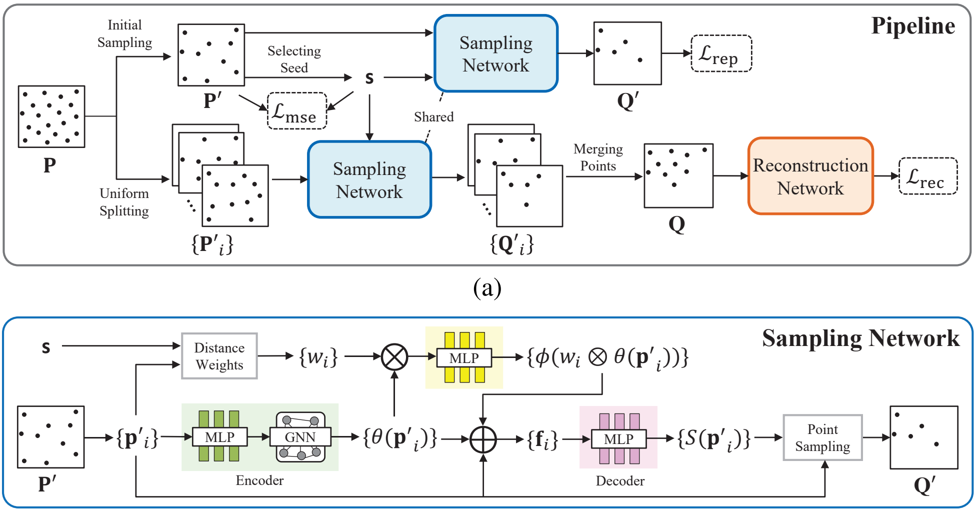
We focus on surface reconstruction from point clouds based on neural fields. Recently, neural fields have achieved remarkable success in the field of surface reconstruction from point clouds. Global methods [25, 26, 30] that encode the entire shape into a feature space are limited in their ability to capture local geometry details, making it difficult to scale to large scenes. Subsequent methods [3, 6, 15, 33, 37, 38] for training neural fields in local regions have been proposed in order to capture more detailed geometry. These methods divide the input point cloud with regular grids or patches, and train neural fields locally using the partial point clouds. However, these methods involve additional limitations such as the non-direct connections between features and points, the requirement for oriented normals, and slow inference speed. To overcome these limitations and achieve better scalability, a recent method [2] extracts features for each input point using a point convolution. This method extracts point-wise features from entire input point clouds using a U-Net-like architecture, and then performs weighted interpolation with neighbors of query points to obtain local features. Although this method achieves faithful reconstruction of large scenes, its scalability is still limited since the memory requirements increase with the number of input points.

2.2 Point Cloud Sampling and Simplification

Point cloud sampling and simplification have been studied to reduce the size of point clouds while preserving appearance quality and geometric properties. They are beneficial for improving the efficiency of storage, transmission, and computational processing associated with downstream tasks. Although many methods [5, 12, 20, 32, 34, 35, 44] reduce the size of point clouds while focusing on specific properties such as edges, contours, and curvature, they do not take the downstream tasks into account. Recently, learning-based point cloud sampling methods [9, 19, 41] have been proposed to achieve optimal performance on downstream tasks. Unlike previous methods, they do not necessarily aim to preserve the appearance quality and geometric properties. In these methods, a task network is first trained, then a sampling network is trained with the frozen task network. The computational efficiency of the task network is improved at inference with sampled point clouds. Therefore, these methods are not suitable for training task networks when computational resources are limited. Furthermore, although these methods sample points that preserve the global structure of the input point clouds, they cannot guarantee that the local geometry will be preserved. In contrast, the proposed method trains the sampling network for surface reconstruction while focusing on preserving both the global structure and local geometry on a part of the scene.

3 Proposed Method

The proposed method aims to sample points to encourage the surface reconstruction network based on neural fields to learn both the global structure and local geometry on a part of the scene. We train neural networks for sampling and surface reconstruction while feeding point clouds sampled by the sampling network to the reconstruction network. In the following, we explain the pipeline of the proposed method in Section 3.1, the sampling network in Section 3.2, and the training details in Section 3.3.



(b)

Figure 1: Overview of the proposed method: (a) pipeline; (b) sampling network. The proposed method feeds a point cloud $\mathbf{Q}$ sampled from the input point cloud $\mathbf{P}$ to the reconstruction network. The weights of two sampling networks are shared in (a). The sampling network samples a point cloud $\mathbf{Q}'$ from a size-suppressed point cloud $\mathbf{P}'$ with a seed point $\mathbf{s}$.

3.1 Pipeline

Figure 1 (a) illustrates a pipeline of the proposed method. Given an input point cloud $\mathbf{P} \in \mathbb{R}^{N \times 3}$, the proposed method constructs a sampled point cloud $\mathbf{Q} \in \mathbb{R}^{M \times 3}$ ($\mathbf{Q} \subseteq \mathbf{P}$) using a learnable sampling network. Then, $\mathbf{Q}$ is fed into a reconstruction network.

The pipeline consists of the two branches drawn at the upper and lower of Figure 1 (a). The upper one is for training the sampling network while suppressing the size of inputs. The lower one is for constructing a sampled point cloud from size-suppressed inputs. This point cloud is constructed to be fed into the reconstruction network. The weights of the sampling networks in these branches are shared. We describe the components of the pipeline below.

Initial Sampling. If the number of input points N increases due to the scale of the scenes, the computer may run out of memory. To avoid this, we first sample a point cloud $\mathbf{P}' \in \mathbb{R}^{N' \times 3}$ from the $\mathbf{P}$ using a random sampling method. Then we feed the point cloud $\mathbf{P}'$ into the sampling network and obtain further sampled point cloud $\mathbf{Q}' \in \mathbb{R}^{M' \times 3}$. The point cloud $\mathbf{Q}'$ is used to measure the loss values with respect to the sampling network. Let r_{init} and r_{nw} be the sampling rates of the initial sampling and sampling network, respectively. The size of $\mathbf{P}'$ and $\mathbf{Q}'$ are represented as $N' = \text{round}(r_{\text{init}} N)$ and $M' = \text{round}(r_{\text{nw}} N')$, respectively.

Selecting Seed. We introduce a seed point to encourage the reconstruction network to capture the detailed geometry on a part of the scene. This is accomplished by our sampling network intensively sampling points from a local region around the seed point. In practice, given the input points $\mathbf{P}'$, we randomly select one of them. Let it be the seed point $\mathbf{s} \in \mathbb{R}^3$. During training, spatially uniform seed points are selected through an iterative process.

Uniform Splitting. We propose a split-and-merge approach to sample a point cloud $\mathbf{Q}$ from a point cloud $\mathbf{P}$ while suppressing the input size to the sampling network. We first split the $\mathbf{P}$ into D point clouds while preserving spatial uniformity. Specifically, we generate D point clouds by iteratively applying a random sampling method to the $\mathbf{P}$ while excluding points sampled so far from the $\mathbf{P}$. The obtained set of the point clouds is denoted as $\{\mathbf{P}'_i \in \mathbb{R}^{N' \times 3}\}_{i=1}^{D}$. Here, we use a sampling rate r_{init} common to that of the initial sampling. Therefore, it becomes $D = \text{round}(1/r_{\text{init}})$. All of these point clouds retain the global structure of the input point cloud $\mathbf{P}$, with no point overlap.

Merging Points. We feed $\mathbf{P}'_i$ into the sampling network instead of $\mathbf{P}$ to suppress the input size. This process is performed sequentially for each $\mathbf{P}'_i$ while freezing model parameters of the sampling network. Here, a common seed point $\mathbf{s}$ is fed into the sampling network along with each $\mathbf{P}'_i$. Then, we obtain a set of sampled point clouds $\{\mathbf{Q}'_i \in \mathbb{R}^{M' \times 3}\}_{i=1}^{D}$. Finally, we merge them to construct a sampled point cloud $\mathbf{Q} \in \mathbb{R}^{M \times 3}$. This can be considered as an approximation of the sampling results from the original input $\mathbf{P}$.

Reconstruction Network. We feed the sampled point cloud $\mathbf{Q}$ to the reconstruction network based on neural fields. The network requires a set of query points, which are usually sampled uniformly within the object's bounding box [25] or near the object's surfaces [24]. In the proposed method, the input shape may become partial due to the sampling, which causes differences in distribution from the query points. To address this, we pre-associate points in the input point cloud $\mathbf{P}$ with their neighboring query points. Then, we use the query points associated with the sampled input points $\mathbf{Q}$ during training.

3.2 Sampling Network

We propose a sampling network for point clouds considering the seed point. Figure 1 (b) shows the architecture of the network. The network takes a point cloud $\mathbf{P}' = \{\mathbf{p}'_i \in \mathbb{R}^3\}_{i=1}^{N'}$ and a seed point $\mathbf{s} \in \mathbb{R}^3$ as inputs, and outputs a sampled point cloud $\mathbf{Q}' = \{\mathbf{q}'_i \in \mathbb{R}^3\}_{i=1}^{M'}$.

Encoder. Given a point cloud $\mathbf{P}'$, we extract features with an encoder network parameterized by θ. The encoder network consists of multi-layer perceptrons (MLP) and graph neural networks (GNN). The MLP is composed of eight fully-connected layers followed by the ReLU activation function [28]. The GNN following the MLP is also activated with the ReLU function. The GNN captures local geometry information for each point from its k-nearest neighbor (NN) graph. Let $\theta(\mathbf{p}'_i)$ denote a feature for i-th point $\mathbf{p}'_i$.

Distance Weights. We introduce distance weights to represent the relevance between the seed point and each input point in the feature space. We adopt the Euclidean distance as the distance metric. The weight w_i for the i-th feature is calculated according to the distance of the point from the seed point using the Gaussian function as follows:

$$w_i = e^{-\frac{\|\mathbf{s}-\mathbf{p}'_i\|^2}{\sigma^2}} \tag{1}$$

where $\mathbf{s}$ is the seed point, $\mathbf{p}'_i$ is the i-th input point, $\|\cdot\|$ is the L2-norm, and σ is a smoothing parameter. We obtain a weighted feature by multiplying the feature $\theta(\mathbf{p}'_i)$ by the weight w_i. Then, we refine the weighted features using an MLP with parameters ϕ.

Decoder. We predict sampling scores $S(\mathbf{p}'_i) \in \mathbb{R}$ for points $\mathbf{p}'_i$ with a decoder network from features $\mathbf{f}_i$. We generate the features $\mathbf{f}_i$ by concatenating the point coordinates $\mathbf{p}'_i$, the extracted features from the encoder, and the weighted features as follows:

$$\mathbf{f}_i = \mathbf{p}'_i \oplus \theta(\mathbf{p}'_i) \oplus \phi(w_i \otimes \theta(\mathbf{p}'_i)), \tag{2}$$

where $\oplus$ and $\otimes$ represent the concatenate operation and the element-wise product, respectively. The architecture of the decoder network is a shallow MLP. All fully connected layers are followed by batch normalization [14] and the ReLU activation function [28], except for the last prediction layer.

Point Sampling. The sampling score indicates the probability that the point is present in the sampled point cloud. We select a set of points $\{\mathbf{q}'_i\}_{i=1}^{M'}$ corresponding to the top M' values of the predicted sampling scores $S(\mathbf{p}'_i)$, denoting the sampled point cloud $\mathbf{Q}'$. Therefore, the point cloud $\mathbf{Q}'$ is a subset of $\mathbf{P}'$. Lastly, the sampling network outputs the $\mathbf{Q}'$.

3.3 Training

We train the sampling network and the reconstruction network while feeding the sampled point clouds to the reconstruction network. We introduce a mean squared error (MSE) loss $\mathcal{L}_{\text{mse}}$ and a repulsion loss $\mathcal{L}_{\text{rep}}$ as loss functions modified in a probabilistic manner regarding point sampling. Therefore, we train the sampling network by minimizing the following total loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \alpha \mathcal{L}_{\text{rep}}, \tag{3}$$

where α is a parameter to balance each term. We describe these loss functions below.

MSE Loss. We introduce a loss function to encourage the sampling network to intensively sample points from a local region around the seed point. To this end, we define the MSE loss between the seed point $\mathbf{s}$ and the input points to the sampling network $\mathbf{P}'$. We modify the MSE loss in a probabilistic manner to make the sampling operation differentiable as follows:

$$\mathcal{L}_{\text{mse}} = \frac{1}{N'} \sum_{i=1}^{N'} g(S(\mathbf{p}'_i)) \|\mathbf{s} - \mathbf{p}'_i\|^2, \tag{4}$$

where $S(\mathbf{p}'_i)$ is the predicted sampling score for the point $\mathbf{p}'_i$, $g(\cdot)$ is the sigmoid function applied to stabilize the training.

Repulsion Loss. We encourage the reconstruction network to learn not only the local geometry, but also the global structure on a part of the scene. Therefore, we also introduce a repulsion loss [42, 43] to make sure that points far away from the seed point are sampled. The loss function penalizes each point in the sampled point cloud $\mathbf{Q}'$ when its neighbors are too close. As with the MSE loss, we modify the loss in a probabilistic manner as follows:

$$\mathcal{L}_{\text{rep}} = \frac{1}{M' \cdot K} \sum_{i=1}^{M'} \sum_{j \in \mathcal{N}(\mathbf{q}'_i)} g(S(\mathbf{q}'_j)) \eta(\|\mathbf{q}'_i - \mathbf{q}'_j\|) \omega(\|\mathbf{q}'_i - \mathbf{q}'_j\|), \tag{5}$$

where $\mathcal{N}(\mathbf{q}'_i)$ is the index set of the K-NN of the point $\mathbf{q}'_i \in \mathbf{Q}'$ excluding i itself, $\eta(a) = -a$, $\omega(a) = e^{-a^2/b^2}$, and b is a parameter that determines the range of influence.

We train the reconstruction network with the reconstruction loss $\mathcal{L}_{\text{rec}}$ defined in a reconstruction method applied to the proposed pipeline. Here, we input the sampled point cloud into the reconstruction network and measure the reconstruction loss. Therefore, the reconstruction network can be optimized for the point clouds sampled by the sampling network.

4 Experiments

We experimentally demonstrate that the point clouds sampled with the proposed method are beneficial for the training of the reconstruction network. We adopt POCO [2] for the reconstruction network, the state-of-the-art surface reconstruction method based on neural fields. At testing, surfaces are reconstructed with the marching cube algorithm [22].

4.1 Experimental Setups

Dataset. We use ScanNet-v2 [8] as the main dataset to evaluate the proposed method by performing experiments on a scene-level surface reconstruction. The ScanNet-v2 dataset consists of large-scale 3D models acquired with an RGB-D sensor from real-world scenes. We split the dataset into 1201/312/100 models for training/validating/testing according to the provided lists. We generated watertight models from the original models with a manifold surface generation method [13] since the dataset does not provide them. We use uniformly sampled points on the surfaces of each model as input point clouds. For training and validating, query points are sampled uniformly from near the surfaces. Occupancy values that represent inside/outside information of the surface are computed for the query points.

Evaluation Metric. We evaluate the surface reconstruction performance to validate the effectiveness of the proposed method. Following [25], we adopt intersection over union (IoU), Chamfer distance (CD), and normal consistency (NC) as evaluation metrics. We use the L1-norm as the point-to-point distance measure when computing the CD. These metrics are measured between randomly sampled points from the reconstructed surface and ground truth surface as approximations of deviations between two surfaces.

Implementation Detail. We implemented the proposed method using PyTorch [31]. To train our model, we used the Adam optimizer [18] with a learning rate of 0.001. We set the batch size to 8 and train the model for 100k iterations. We empirically use $\alpha = 0.1$ in Eq. (3). The reconstruction loss $\mathcal{L}_{\text{rec}}$ is a cross entropy loss [2] for occupancy prediction. We set the sampling rate for the initial sampling and sampling network to $r_{\text{init}} = 0.1$ and $r_{\text{nw}} = 0.1$. Thus, the number of splits is $D = 10$. The size of point clouds $\mathbf{P}$, $\mathbf{P}'$, $\mathbf{Q}'$ and $\mathbf{Q}$ are $N =$100k, $N' =$10k, $M' =$1k, and $M =$10k, respectively. The number of query points used to train the reconstruction network is 10k. We set $\sigma = 1.0$ in Eq. (1), and $K = 4$ as in [42] and $b = 1.0$ in Eq. (5). The dimensions of the features $\theta(\mathbf{p}'_i)$, $\phi(w_i \otimes \theta(\mathbf{p}'_i))$ and $\mathbf{f}_i$ are 64, 64, and 131, respectively. The neighborhood size used in the GNN is set to $k = 7$ as in [34]. To achieve better reconstruction performance, we use unsampled input point clouds $\mathbf{P}$ of size N for testing[1]. All experiments are conducted on a computer equipped with a NVIDIA RTX A6000 GPU, Intel Core i9-13900K CPU (3.0 GHz), and 128 GB of RAM.

4.2 Comparison with State-of-the-art Methods

Comparison Methods. We conduct comparisons with previous point cloud sampling and simplification methods. For the comparison, we choose the random sampling method as a Baseline. We also compare the proposed method to the state-of-the-art learning-based methods, SampleNet [19] and RPCS [34]. For training SampleNet, we first trained the reconstruction network with input point clouds of size N since SampleNet requires a pre-trained task network. We then trained SampleNet to sample M points from point clouds of size N using the frozen pre-trained reconstruction network. RPCS is trained independently to generate a simplified point cloud of size M since it does not require a task network. We train the reconstruction network using a point cloud of size M constructed by each method as inputs.

Quantitative Evaluation. Table 1 summarizes the surface reconstruction performance with POCO. SampleNet tends to sample spatially uniform points since it does not consider the preservation of local geometry. As a result, a similar performance to the Baseline is observed. RPCS constructs point clouds that are beneficial for learning the local geometry

[1] We do not apply scene scaling [2] since it degraded reconstruction performance in our preliminary experiments.

Table 1: Surface reconstruction performance. CD is scaled by 10^2. ↑ (↓) denotes higher (lower) is better.

	POCO			ALTO		
Method	IoU ↑	CD ↓	NC ↑	IoU ↑	CD ↓	NC ↑
Baseline	0.810	0.310	0.923	0.800	0.323	0.918
SampleNet [19]	0.811	0.308	0.925	0.805	0.319	0.919
RPCS [34]	0.840	0.306	0.936	0.801	0.320	0.915
Ours	**0.924**	**0.291**	**0.948**	**0.828**	**0.309**	**0.928**

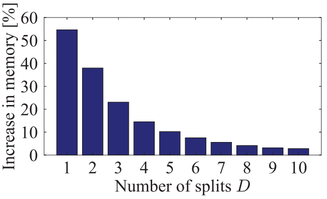


Figure 2: Increase in memory footprint as a function of the number of splits D.

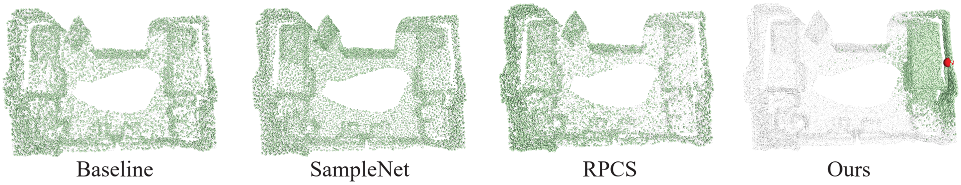


Figure 3: Sampled point clouds. Gray and green points represent the input and sampled point clouds, respectively. The red point represents the seed point in the proposed method.

since it intensively samples points with high curvature. Therefore, a reasonable performance improvement is confirmed. However, it still tends to sample points with high spatial uniformity, limiting the amount of improvement. The proposed method achieves the best performance in all metrics. The proposed method preserves local details around the seed point while sampling points far from it. This encourages the reconstruction network to learn both global structure and local geometry on a part of the scene, resulting in a significant improvement. Table 1 also shows the performance when ALTO [40] is adopted as the reconstruction network in order to verify the effectiveness of the proposed method for different surface reconstruction methods. Even in this case, it can be seen that the proposed method achieves superior surface reconstruction performance compared to the other methods.

We also evaluate the memory efficiency of the proposed method. Figure 2 shows the increase in the memory footprint of the proposed method for training compared to the Baseline. Here, we draw it as a function of the number of splits D to verify the effectiveness of our split-and-merge approach. In this experiment, the memory footprint of the Baseline is 20.5 GB. It can be seen that the increase in memory footprint decreases rapidly as D becomes larger. The input size to the reconstruction network is the same across all D. On the other hand, the input size to the sampling network is effectively suppressed in proportion to the value of D. As a result, memory efficiency improves as D becomes larger. Therefore, the proposed method significantly improves surface reconstruction performance when using $D = 10$, with only a 2.8% additional memory footprint. Although the size of sampled point cloud $\mathbf{Q}$ increases with the size of input point cloud $\mathbf{P}$, the proposed method can suppress the size of $\mathbf{Q}$ to the acceptable range of memory by adjusting the sampling rates r_{init} and r_{nw}.

Qualitative Evaluation. We qualitatively evaluate the sampled point clouds and the reconstructed surfaces. Figure 3 illustrates the sampled point cloud from the input point cloud. Gray and green points represent the input and sampled point clouds, respectively. A red point in the proposed method represents the seed point. Baseline samples points randomly. It is observed that SampleNet samples point clouds with higher spatial uniformity than Baseline. RPCS samples to preserve points with high curvature and does not sample much from flat areas. It can be seen that the proposed method samples points intensively around the seed

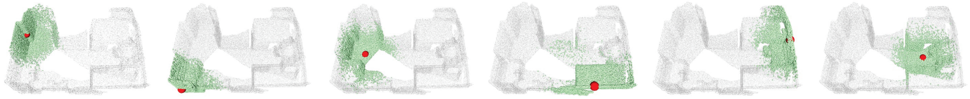


Figure 4: Sampled point clouds for various seed points in the proposed method.

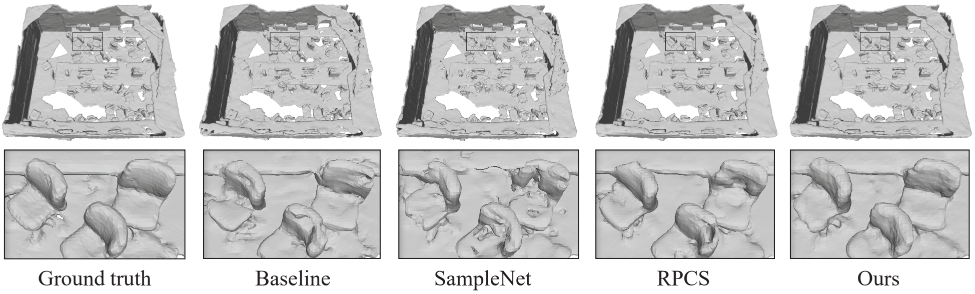


Figure 5: Reconstructed surfaces. The upper row represents the overall view, and the lower represents a partially enlarged view. White spaces represent areas with no data.

Table 2: Ablation study. CD is scaled by 10^2. ↑ (↓) denotes higher (lower) is better.

Method	IoU ↑	CD ↓	NC ↑
k-NN sampling	0.647	2.182	0.896
Random query	0.852	0.306	0.930
Remove $\mathcal{L}_{\text{mse}}$	0.804	0.313	0.919
Remove $\mathcal{L}_{\text{rep}}$	0.918	0.294	0.944
Complete	**0.924**	**0.291**	**0.948**

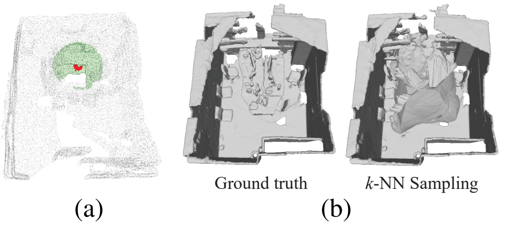


Figure 6: Results of k-NN sampling: (a) sampled point cloud; (b) reconstructed surface.

point, while also sampling points far from it. In the proposed method, various seed points are selected from the input points during training, as shown in the examples in Figure 4.

Figure 5 visualizes the reconstructed surfaces. The upper row represents the overall view, and the lower represents the partially enlarged view. Comparison methods have low reconstruction accuracy for local geometry details, resulting in distortion and rough surfaces. On the other hand, it can be seen that the proposed method reconstructs surfaces more faithfully compared with the other methods. This is because the proposed method makes it possible for the surface reconstruction network to learn the local geometry on a part of the scene.

4.3 Ablation Study

We provide ablation studies to verify how the design choices made in the proposed method impact the final results.

***k*-NN Sampling.** Row 1 in Table 2 shows the results of constructing a sampled point cloud $\mathbf{Q}$ with k-NN search from the seed point instead of our sampling network. This constructs a locally concentrated point cloud as shown in Figure 6 (a), and the reconstruction network can hardly learn the global structure. This often leads to large distortions in the reconstructed surfaces as shown in Figure 6 (b).

Query Points. Row 2 in Table 2 shows the results of randomly selecting query points without

Table 3: Generalization ability with POCO. CD is scaled by 10^2. ↑ (↓) denotes higher (lower) is better.

	Scene3D			Replica		
Method	IoU ↑	CD ↓	NC ↑	IoU ↑	CD ↓	NC ↑
Baseline	0.721	0.250	0.938	0.650	0.258	0.949
SampleNet [19]	0.726	0.244	0.939	0.668	0.255	0.951
RPCS [34]	0.718	0.256	0.934	0.634	0.273	0.942
Ours	**0.848**	**0.238**	**0.947**	**0.850**	**0.235**	**0.958**

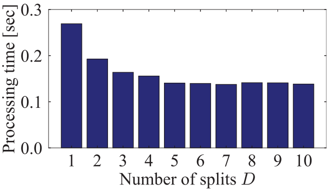


Figure 7: Processing time of the sampling network as a function of the number of splits D.

associating them with input points. In this case, the reconstruction performance is degraded due to the difference in the distribution of sampled points and query points.

Loss Functions. Rows 3 and 4 in Table 2 show the results when each loss term in Eq. (3) is removed. If these losses are removed, it can also be seen that the reconstruction performance is degraded. When removing the MSE loss, local geometry details are lost due to spatially uniform sampling. If the repulsion loss is removed, it becomes difficult to capture the global structure since the sampled points are concentrated around the seed point. We have explored a parameter α that achieves the proper balance of these losses. Therefore, the best performance is obtained from the complete proposed method.

Generalization. We evaluate the generalization ability of all methods on the scene-level datasets Scene3D [45] and Replica [36]. Table 3 shows the surface reconstruction performance tested on these datasets with POCO trained on the ScanNet-v2 dataset. As can be seen, the proposed method achieved better generalization to both datasets.

Sampling Efficiency. We provide processing time of the sampling network as a function of the number of splits D in Figure 7. Note that these are the total sampling times for all split point clouds. The processing time tends to become shorter as D becomes larger. This is because the sampling network involves processes such as k-NN graph construction, where computational complexity increases non-linearly with the size of input point clouds.

Sampling Rate. We investigate the effect of the sampling rate of the sampling network r_{nw} on the surface reconstruction performance. Figure 8 shows the CD as a function of the sampling rate. The performance tends to degrade with lower sampling rates, while the performance remains almost the same when the sampling rate is above 0.1. This indicates that a performance close to that of a sampling rate of 1 can be achieved at much lower sampling rates.

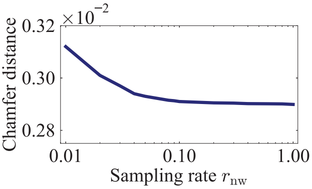


Figure 8: Chamfer distance as a function of sampling rate r_{nw}.

5 Conclusions

In this paper, we proposed a point cloud sampling method to improve the scalability of surface reconstruction with neural fields. We introduced a seed point that serves as the origin to encourage the sampling network to sample point clouds from partial regions of the scene. Our split-and-merge approach allowed the construction of sampled point clouds while suppressing the input size to the sampling network. Experimental results show that the proposed method improves the performance of the surface reconstruction while avoiding an increase in memory footprint for training. In the future, we will explore ways to extend the proposed method to train the reconstruction network without ground truth.

Acknowledgement

These research results were obtained from the commissioned research (No. 06801) by National Institute of Information and Communications Technology (NICT) , Japan.

References

[1] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[2] Alexandre Boulch and Renaud Marlet. POCO: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022.

[3] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *Proceedings of the European Conference on Computer Vision*, pages 608–625. Springer, 2020.

[4] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Latent partition implicit with surface codes for 3D representation. In *Proceedings of the European Conference on Computer Vision*, pages 322–343. Springer, 2022.

[5] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovačević. Fast resampling of three-dimensional point clouds via graphs. *IEEE Transactions on Signal Processing*, 66(3):666–681, 2017.

[6] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Häne, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3D shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13087–13096, 2021.

[7] David Cohen-Steiner and Frank Da. A greedy Delaunay-based surface reconstruction algorithm. *The Visual Computer*, 20:4–16, 2004.

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[9] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2019.

[10] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.

[11] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018.

[12] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM Transactions on Graphics*, 32(1):1–12, 2013.

[13] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for ShapeNet models. *arXiv preprint arXiv:1802.01698*, 2018.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[15] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[16] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.

[17] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Itai Lang, Asaf Manor, and Shai Avidan. SampleNet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[20] Nallig Leal, Esmeide Leal, and Sanchez-Torres German. A linear programming approach for 3D point cloud simplification. *IAENG International Journal of Computer Science*, 44(1):60–67, 2017.

[21] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3D reconstruction. In *Proceedings of the Computer Graphics Forum*, volume 34, pages 435–446. Wiley Online Library, 2015.

[22] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987.

[23] Yiming Luo, Zhenxing Mi, and Wenbing Tao. DeepDT: Learning geometry from Delaunay triangulation for surface reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2277–2285, 2021.

[24] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance function from point clouds by learning to pull space onto surface. In *Proceedings of the International Conference on Machine Learning*, pages 7246–7257. PMLR, 2021.

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[26] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019.

[27] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003.

[28] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, pages 807–814, 2010.

[29] Liangliang Nan and Peter Wonka. Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2353–2361, 2017.

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Mark Pauly, Markus Gross, and Leif P Kobbelt. Efficient simplification of point-sampled surfaces. In *Proceedings of the IEEE Visualization Conference*, pages 163–170. IEEE, 2002.

[33] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision*, pages 523–540. Springer, 2020.

[34] Rolandos Alexandros Potamias, Giorgos Bouritsas, and Stefanos Zafeiriou. Revisiting point cloud simplification: A learnable feature preserving approach. In *Proceedings of the European Conference on Computer Vision*, pages 586–603. Springer, 2022.

[35] Junkun Qi, Wei Hu, and Zongming Guo. Feature preserving and uniformity-controllable point cloud simplification on graph. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 284–289. IEEE, 2019.

[36] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[37] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. SA-ConvOnet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6504–6513, 2021.

[38] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-based generalizable deep implicit 3D shape representations. In *Proceedings of the European Conference on Computer Vision*, pages 293–309. Springer, 2020.

[39] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics*, 37(6):1–11, 2018.

[40] Zhen Wang, Shijie Zhou, Jeong Joon Park, Despoina Paschalidou, Suya You, Gordon Wetzstein, Leonidas Guibas, and Achuta Kadambi. ALTO: Alternating latent topologies for implicit 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 259–270, 2023.

[41] Yang Ye, Xiulong Yang, and Shihao Ji. APSNet: Attention based point cloud sampling. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2022.

[42] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. EC-Net: An edge-aware point set consolidation network. In *Proceedings of the European Conference on Computer Vision*, pages 386–402, 2018.

[43] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-Net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.

[44] Kun Zhang, Shiquan Qiao, Xiaohong Wang, Yongtao Yang, and Yongqiang Zhang. Feature-preserved point cloud simplification based on natural quadric shape models. *Applied Sciences*, 9(10):2130, 2019.

[45] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4):1–8, 2013.