

大規模シーンを対象とした形状復元の高精度化のための点選択

Point Selection for Accurate Shape Reconstruction of Large-scale Scenes

松崎 康平[†]

Kohei Matsuzaki

野中 敬介[†]

Keisuke Nonaka

概要

本稿では、大規模なシーンを表す点群から3次元形状を高精度に復元するために、深層学習に基づく点選択手法を提案する。最先端の形状復元手法は入力点群の点数に応じてメモリ使用量が増加するため、大規模なシーンへの拡張性が制限されている。提案手法は入力点群から一部の点を選択することにより、形状復元手法によるメモリ使用量を抑制する。また、提案手法はタスク損失を用いて点選択ネットワークを学習させることにより、形状復元タスクに最適化された点を選択する。さらに、点選択ネットワークへ入力される点の個数を抑制するための分割統合法を導入する。ScanNet データセットを用いた評価実験により、提案手法の有効性を確認した。

1 はじめに

3次元センシング技術の普及に伴い、点群から物体の3次元形状を復元する手法の検討が重要な研究課題の一つとなっている。古典的な手法は三角形分割 [1] や平滑性 [2] に基づく最適化によって点群から3次元形状を復元する。近年では、高精度な復元を実現できることから、ニューラル場に基づく形状復元手法が注目を集めている [3–7]。この技術では点の座標からその点の二値の占有確率 [3] や符号付き距離 [4] を予測するニューラル場を学習する。ニューラル場は連続的な表現形式であるため、離散的な表現形式に基づく手法 [8–10] に比べて高解像度な形状を復元することができる。

ニューラル場に基づく代表的な形状復元手法 [3,4] は3次元形状全体から大域特徴を抽出する。これらの手法は局所形状を高精度に表現することが困難であるため、復元対象が単一の物体に制限されている。この制限を克服し、復元対象を大規模なシーンへ拡張するために様々な手法が提案されている。例えば、3次元空間をボクセルを用いて分割し、ボクセルに基づく畳み込みによって局所領域ごとに特徴を抽出する手法が提案されている [5,11]。また、入力形状を複数の部分的な形状に分割し、個別に特徴を抽出する手法も提案されている [6,12]。より最近の手法は、点畳み込みを用いて点ごとに特徴を抽出する [7]。しかし、この手法は入力点群の点数に応じてメモリ使用量が増加するため、大量の点を処理する場合にはメモリ不足に陥る恐れがある。

前述の課題を解決するための有効な手段として、入力点群から一部の点を選択するサンプリングが挙げられる。最も一般的な手法は、ランダムサンプリング (Random Sampling, RS) と最遠方点サンプリング (Farthest Point Sampling) [13] である。これらの手法は空間的一様に点を選択するため、サンプリング後の点群では局所形状の詳細が失われる。一部の手法 [14–16] はエッジや輪郭、曲率のような幾何的な特徴に焦点を当てて局所形状

の詳細を保持した点の選択を実現するが、これらの手法はサンプリングの後段のタスクに適した点を選択するとは限らない。一方、学習に基づく手法 [17–19] はタスク損失を用いてニューラルネットワークを学習させることにより、後段のタスクに対して最適化された点の選択を実現する。しかし、これらの手法は主に推論時の計算効率を向上させることに焦点を当てており、学習時の拡張性の改善についてはほとんど検討されていない。

本稿では、ニューラル場に基づく形状復元手法の拡張性を改善するために、深層学習を用いた点選択手法を提案する。提案手法では、点の選択ネットワークを用いて入力点群から一部の点を選択することにより、メモリ使用量を抑制しながら形状復元ネットワークを学習させる。ここでは、選択点群を形状復元に対して最適化させるために、タスク損失を用いて選択ネットワークと復元ネットワークを同時に学習させる。また、選択ネットワークは復元ネットワークがシーンの一部の局所形状と局所形状の両方を学習することを促す点群を選択することを目指す。この目的のために、シーンの一部を中心とした点選択の起点となる中心点を導入する。提案手法は中心点の周辺から点群を選択するため、選択点群は必ずしも完全な入力形状を表さず、部分的な形状を表す可能性がある。しかし、ネットワークの学習時には空間的一様に選択された中心点により、入力点群のほとんどの点を選択される。提案手法のこの性質は、処理可能な点数に制限がある場合に、復元対象を大規模なシーンへ拡張するために有益である。さらに、点の選択によるメモリ使用量の増加を回避するために、点群の分割統合法も提案する。これは入力点群を分割することによって選択ネットワークへの入力点数を抑制し、個々の選択結果を統合することによって最終的な選択点群を構築する。結果として、従来の点選択手法と比べて、提案手法は形状復元精度を改善することができる。

本稿の主要な貢献は次のようにまとめられる。

- 学習可能な選択ネットワークから選択された点を用いて3次元形状を表現するニューラル場を学習するための新たな手法を提案する。
- シーンの一部の大域形状と局所形状の両方を表す点を選択するために、中心点を考慮した選択ネットワークを提案する。
- 点選択によるメモリ使用量の増加を防ぐために、選択ネットワークへ入力される点の個数を抑制するための分割統合法を導入する。

本稿の以降の構成は次の通りである。第2節では点群からの形状復元と点選択に関連する研究を概説する。第3節では学習可能な選択ネットワークを用いて形状復元に最適化された点を選択する点選択手法を提案する。第4節では大規模なシーンを表すデータセットを用いて提案手法の有効性を評価する。第5節でまとめを述べる。

[†] 株式会社 KDDI 総合研究所 KDDI Research, Inc.

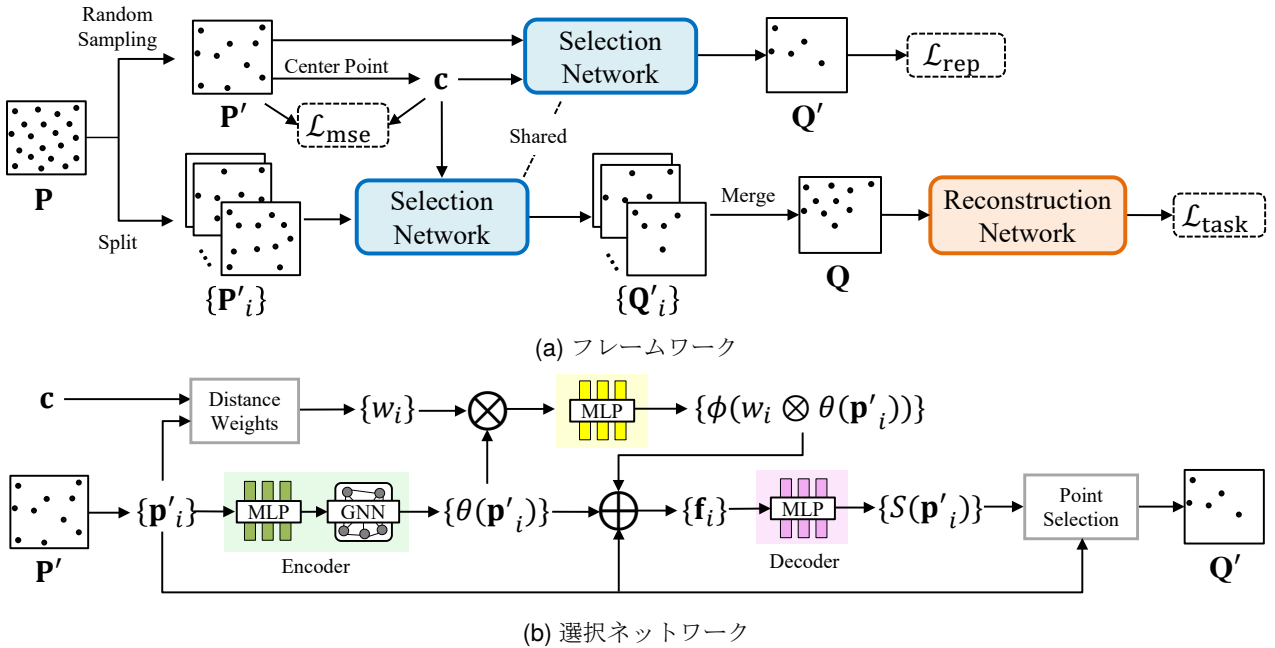


図1. 提案手法の概要. 提案手法は入力点群 \mathbf{P} から選択ネットワークを用いて選択された点群 \mathbf{Q} を復元ネットワークへ入力する. (a) における2つの選択ネットワークはモデルパラメータを共有する.

2 関連研究

2.1 形状復元

本稿では、ニューラル場に基づく点群からの形状復元に焦点を当てる. この技術では、物体形状を表す入力点群から抽出した特徴を用いて、クエリ点の占有確率や符号付き距離を予測することによって物体の形状を復元する. 入力点群から単一の特徴を抽出する手法 [3,4] は局所形状の詳細を捉える能力に制限があり、大規模なシーンを高精度に復元することが困難である. より詳細な形状を捉えるために、局所的な領域でニューラル場を学習させる手法が提案されている [5-7, 11, 12, 20, 21]. これらの手法はボクセルグリッドや局所的なパッチによって入力点群を分割し、分割された点群を用いてニューラル場を学習させる. しかし、これらの手法には物体表面と特徴位置の不整合や、方向付き法線の要求、推論速度の低下のような追加の制限が伴う.

これらの制限を克服し、拡張性を改善するために、POCO [7] は点畳み込みに基づいて入力点ごとに特徴を抽出した. この手法は入力点群から U-Net 型の点畳み込みニューラルネットワークを用いて大域的な特徴と局所的な特徴を捉える. そして、クエリ点の近傍に位置する入力点の特徴を用いて重み付き補間を行うことでクエリ点の特徴を取得する. この手法は大規模なシーンに対する高精度な形状復元を実現するが、入力点群の点数に応じてメモリ使用量が増加するため、依然として拡張性に制限がある.

2.2 サンプリングと単純化

点群から一部の点を選択するための代表的な方法として、サンプリングと単純化がある. これらは点群の外観品質や幾何学的特徴を保持しつつ点数を削減するため、データ量の削減や後段のタスクにおける計算効率の改善

に有益である. 多くの従来手法 [14-16, 22-24] はエッジや輪郭、曲率のような幾何学的特徴に焦点を当てて点を選択するが、選択された点は後段のタスクに対して最適化されていない.

近年、後段のタスクで最適な性能を得るために、深層学習を用いた点のサンプリング手法が提案された [17-19]. 多くの従来手法とは異なり、これらの手法は必ずしも外観品質や特定の幾何学的特徴を保持することを目的にしない. これらの手法ははじめにタスクネットワークを学習させ、その後学習済みのタスクネットワークを用いてサンプリングのためのネットワークを学習させる. そして、推論時にサンプリングのための学習済みネットワークを用いて点を選択することにより、タスクネットワークの計算効率を改善することができる. したがって、これらの手法は処理可能な点の個数に制限がある場合に、タスクネットワークを学習させる目的には適していない. また、これらの手法は入力点群から大域形状を保持する点を選択するが、局所形状の保持は考慮しない. 対照的に、提案手法はシーンの一部の大域形状と局所形状の両方の保持に焦点を当てながら、タスクと点選択のためのネットワークを同時に学習させる.

3 提案手法

提案手法はニューラル場に基づく形状復元ネットワークがシーンの一部の大域形状と局所形状の両方を学習するように点を選択する. 選択ネットワークと復元ネットワークはタスク損失を用いて同時に学習させる. これにより、復元タスクに最適化された点を選択される.

3.1 フレームワーク

図1(a)に提案手法のフレームワークを示す. 入力点群 $\mathbf{P} \in \mathbb{R}^{N \times 3}$ が与えられた場合、提案手法は学習可能な選択ネットワークを用いて選択点群 $\mathbf{Q} \in \mathbb{R}^{M \times 3}$ ($\mathbf{P} \supseteq \mathbf{Q}$)

を構築する。そして、選択点群 \mathbf{Q} を復元ネットワークへ入力する。

提案手法は図 1 (a) の上部と下部に描かれた 2 つのブランチで構成される。上部のブランチは、点数を抑制した点群を用いて選択ネットワークを学習させるためのものである。下部のブランチは、点数の抑制された点群から、復元ネットワークへ入力する選択点群を構築するためのものである。これらのブランチでは、選択ネットワークのモデルパラメータは共有されている。以下では、フレームワークの構成要素について説明する。

シーンの大きさに起因して入力点群の点数 N が多くなる場合、メモリ不足が発生する恐れがある。この問題を回避するために、はじめに \mathbf{RS} を用いて \mathbf{P} から一部の点を選択し、点群 $\mathbf{P}' \in \mathbb{R}^{N' \times 3}$ を構築する。そして、点群 \mathbf{P}' を選択ネットワークへ入力し、選択点群 $\mathbf{Q}' \in \mathbb{R}^{M' \times 3}$ を得る。この点群 \mathbf{Q}' は選択ネットワークに関する損失を計算するために使用する。 \mathbf{RS} と選択ネットワークの選択率をそれぞれ r_{rs} と r_{nw} とする。この場合、点群 \mathbf{P}' と \mathbf{Q}' の点数はそれぞれ $N' = \text{round}(r_{rs}N)$ と $M' = \text{round}(r_{nw}N')$ と表される。

復元ネットワークがシーンの一部の詳細な幾何形状を学習することを促すために、中心点を導入する。これは、選択ネットワークが中心点の周囲の局所領域から集中的に点を選択することによって実現される。具体的には、点群 \mathbf{P}' が与えられた場合、その中から無作為に 1 点を選択する。この点を中心点 $\mathbf{c} \in \mathbb{R}^3$ とする。学習時には、反復的な処理を通じて点群 \mathbf{P}' から空間的一様に中心点を選択される。

選択ネットワークへ入力する点群の点数を抑制しつつ、点群 \mathbf{P} から点群 \mathbf{Q} を構築するために、点群の分割統合法を提案する。はじめに点群 \mathbf{P} を空間的一様性を保ちつつ D 個に分割する。具体的には、点群 \mathbf{P} に対してこれまでに選択された点を除外しながら反復的に \mathbf{RS} を適用することによって D 個の点群を構築する。得られた点群の集合を $\{\mathbf{P}'_i\}_{i=1}^D$ と表記する。これらの点群は全て入力点群 \mathbf{P} の大域形状を保持しており、点の重複を持たない。ここで、 \mathbf{RS} の選択率としては点群 \mathbf{P}' を構築する場合と同値の r_{rs} を使用する。したがって、 $D = \text{round}(1/r_{rs})$ となる。

点数を抑制するために、入力点群 \mathbf{P} の代わりに各 \mathbf{P}'_i を選択ネットワークへ入力する。この処理は選択ネットワークのモデルパラメータを固定しながら、各 \mathbf{P}'_i に対して順次的に実行する。ここでは、 $\{\mathbf{P}'_i\}_{i=1}^D$ のそれぞれに対して共通の中心点を用いる。そして、選択点群の集合 $\{\mathbf{Q}'_i \in \mathbb{R}^{M' \times 3}\}_{i=1}^D$ を得る。最後に、これらを統合することで選択点群 $\mathbf{Q} \in \mathbb{R}^{M \times 3}$ を構築する。これは入力点群 \mathbf{P} からの選択結果の近似とみなすことができる。

選択点群 \mathbf{Q} をニューラル場に基づく復元ネットワークへ入力する。このネットワークは一般に対象物体のバウンディングボックス内 [3] や入力点群の周辺 [25] からサンプリングされたクエリ点群を必要とする。提案手法では選択点群が対象物体の部分的な形状を表す可能性があり、結果として選択点群とクエリ点群の分布に差異が生じる恐れがある。この問題に対処するために、学習の前処理として入力点群 \mathbf{P} 内の点と、その近傍に位置するクエリ点に関連付ける。そして、学習時には \mathbf{P} から選択された点に関連付けられているクエリ点を使用する。

3.2 選択ネットワーク

中心点を考慮した点の選択ネットワークを提案する。図 1 (b) に選択ネットワークの構成を示す。このネットワークは点群 $\mathbf{P}' = \{\mathbf{p}'_i \in \mathbb{R}^3\}_{i=1}^{N'}$ と中心点 $\mathbf{c} \in \mathbb{R}^3$ を入力とし、選択点群 $\mathbf{Q}' = \{\mathbf{q}'_i \in \mathbb{R}^3\}_{i=1}^{M'}$ を出力する。

点群 \mathbf{P}' が与えられた場合、 θ で表されるエンコーダネットワークを用いて特徴を抽出する。エンコーダネットワークは MLP (Multi-Layer Perceptrons) と GNN (Graph Neural Networks) で構成する。MLP の全ての全結合層と GNN の後には ReLU 活性化を導入する。GNN は各点の k 近傍グラフから局所形状情報を取得する。点群 \mathbf{P}' の i 番目の点 \mathbf{p}'_i に対する特徴を $\theta(\mathbf{p}'_i)$ と表記する。

提案手法では、特徴空間において中心点と入力点群の各点の関連性を表すための距離重み付けを導入する。距離尺度としてユークリッド距離を使用する。 i 番目の特徴に対する重み w_i は、次式のようにガウス関数を用いて中心点からの距離に応じて計算する。

$$w_i = e^{-\frac{\|\mathbf{c}-\mathbf{p}'_i\|^2}{\sigma^2}} / \sum_{i=1}^{N'} e^{-\frac{\|\mathbf{c}-\mathbf{p}'_i\|^2}{\sigma^2}}, \quad (1)$$

ここで \mathbf{c} は中心点、 \mathbf{p}'_i は i 番目の入力点、 $\|\cdot\|$ は L2 ノルム、 σ は調整可能なパラメータである。分母は正規化係数を表す。重み w_i を特徴 $\theta(\mathbf{p}'_i)$ に乗算することにより、重み付き特徴を得る。そして、 ϕ で表される MLP を用いて重み付き特徴を洗練する。

特徴 \mathbf{f}_i からデコーダネットワークを用いて点 \mathbf{p}'_i に対する選択確率 $S(\mathbf{p}'_i) \in \mathbb{R}$ を予測する。特徴 \mathbf{f}_i は、次式のように点の座標 \mathbf{p}'_i 、特徴 $\theta(\mathbf{p}'_i)$ 、重み付き特徴を結合することで生成する。

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

ここで \oplus は結合操作、 \otimes は要素ごとの積を表す。デコーダネットワークは MLP で構成し、最後の層を除く全ての全結合層の後にバッチ正規化と ReLU 活性化を導入する。

選択確率に対応する点在选择点群に存在する確率を表す。予測された選択確率 $S(\mathbf{p}'_i)$ の上位 M' 件に対応する点の集合を選択し、選択点群 $\mathbf{Q}' = \{\mathbf{q}'_i\}_{i=1}^{M'}$ とする。したがって、選択点群 \mathbf{Q}' は点群 \mathbf{P}' の部分集合である。最終的に、選択ネットワークは選択点群 \mathbf{Q}' を出力する。

3.3 損失関数

提案手法では、選択ネットワークと復元ネットワークを同時に学習させる。点の選択に関する損失関数として、選択確率を用いて修正した MSE (Mean Squared Error) 損失 \mathcal{L}_{mse} と反発損失 \mathcal{L}_{rep} を導入する。さらに、形状復元タスクに対してネットワークを最適化するために、タスク損失 \mathcal{L}_{task} を使用する。したがって、次式の損失 \mathcal{L} を最小化することによってネットワークを学習させる。

$$\mathcal{L} = \mathcal{L}_{mse} + \alpha \mathcal{L}_{rep} + \beta \mathcal{L}_{task}, \quad (3)$$

ここで α および β は各項のバランスを調整するためのパラメータである。これらの損失関数について以下で説明する。

MSE 損失は、選択ネットワークが中心点の周辺の局所領域から点を選択することを促すために導入する。この目的のために、中心点と選択ネットワークへの入力点

群 \mathbf{P}' の間で MSE を計算する。点の選択操作を微分可能にするために、次式のように選択確率 $S(\mathbf{p}_i)$ を用いて MSE を修正する。

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

ここで $g(\cdot)$ は学習を安定させるために適用されるシグモイド関数である。

選択ネットワークはシーンの一部の局所形状だけでなく、大域形状も表す点を選択することを目指す。そのため、中心点から離れた点を選択するために反発損失 [26] を導入する。この損失関数により、選択された点 \mathbf{q}'_i とその近傍点との距離が遠くなるように促される。MSE 損失と同様に、次式のように選択確率 $S(\mathbf{p}_i)$ を用いて反発損失を修正する。

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

ここで、 $\mathcal{N}(\mathbf{q}'_i)$ は点 \mathbf{q}'_i の自身を除いた K 近傍点のインデックス、 $\eta(a) = -a$ 、 $\omega(a) = e^{-a^2/b^2}$ 、 b は調整可能なパラメータである。

タスク損失は、選択ネットワークが形状復元タスクに適した点を選択することを促す。タスクネットワークを固定する従来手法 [17–19] とは対照的に、提案手法は選択ネットワークと復元ネットワークを同時に学習させる。これにより、選択点群を用いて復元ネットワークの性能を改善することが可能になる。

4 評価実験

提案手法で選択した点群が形状復元ネットワークの学習に有効であることを実験的に示す。形状復元ネットワークには、最先端手法である POCO [7] を使用する。

4.1 実験設定

シーンの形状復元に対する提案手法の有効性を評価するために、ScanNet [27] データセットを使用する。このデータセットは実世界の屋内環境から取得された大規模な 3D 形状モデルで構成される。データセットに定義されたリストを用いて、学習、検証、評価のためにデータセットを分割する。前処理として多様体形状生成手法 [28] を用いて 3D 形状モデルを水密なものに変換する。入力点群は 3 次元形状モデルの表面から一様に点をサンプリングすることで生成する。クエリ点群は入力点群の周辺からサンプリングし、占有確率として 3 次元形状モデルに対する内外判定結果を計算する。

提案手法の有効性を評価するために、3 次元形状モデルに対する形状復元精度を測定する。評価尺度として、IoU (Intersection over Union)、CD (Chamfer Distance)、NC (Normal Consistency) [3] を使用する。CD を計算する際には、点と点の距離尺度として L1 ノルムを用いる。

提案手法は PyTorch を用いて実装する。ネットワークの学習には学習率 10^{-3} を持つ Adam optimizer [29] を使用する。バッチサイズを 8 に設定し、NVIDIA RTX A6000 GPU を用いて 100k iteration の学習を行う。式 (3) においては経験的に $\alpha = 0.1$ 、 $\beta = 0.001$ に設定する。式 (3) におけるタスク損失 $\mathcal{L}_{\text{task}}$ は、占有予測に関する

表 1. 形状復元精度の比較.

Method	IoU \uparrow	CD \downarrow	NC \uparrow
RS	0.810	0.00310	0.923
SampleNet [18]	0.811	0.00308	0.925
RPCS [24]	0.840	0.00306	0.936
Proposed	0.929	0.00291	0.949

表 2. Scene scaling の効果.

Method	IoU \uparrow	CD \downarrow	NC \uparrow
RS	0.408	0.00435	0.760
SampleNet [18]	0.318	0.00546	0.808
RPCS [24]	0.474	0.00408	0.800
Proposed	0.876	0.00330	0.918

クロスエントロピー損失 [7] である。RS と選択ネットワークの選択率はそれぞれ $r_{\text{rs}} = 0.1$ 、 $r_{\text{nw}} = 0.1$ に設定する。したがって、点群 \mathbf{P} の分割数は $D = 10$ となる。点群 \mathbf{P} 、 \mathbf{P}' 、 \mathbf{Q}' 、 \mathbf{Q} の点数はそれぞれ $N = 100\text{k}$ 、 $N' = 10\text{k}$ 、 $M' = 1\text{k}$ 、 $M = 10\text{k}$ に設定する。復元ネットワークの学習に使用するクエリ点の数は 10k とする。式 (1) では $\sigma = 1$ 、式 (5) では $K = 4$ 、 $b = 1$ に設定する。特徴 $\theta(\mathbf{p}'_i)$ 、 $\phi(\mathbf{w}_i \otimes \theta(\mathbf{p}'_i))$ 、 \mathbf{f}_i の次元はそれぞれ 64、64、131 とする。GNN で使用される近傍点の数は $k = 7$ に設定する。より高い形状復元精度を達成するために、推論時には点選択を行わずに点数 N の入力点群 \mathbf{P} を使用する。

4.2 従来手法との比較

提案手法と従来の点選択手法との比較を行う。本実験では、広く利用される RS を比較手法とする。さらに、学習に基づく点選択の最先端手法である SampleNet [18] と RPCS [24] とも比較する。SampleNet の学習には事前学習されたタスクネットワークが必要であるため、はじめに点数 N の入力点群を用いて復元ネットワークを学習させる。その後、学習済みの復元ネットワークを用いて点数 N の点群から M 点を選択するように SampleNet を学習させる。RPCS はタスクネットワークを必要としないため、点数 N の点群から M 点を選択するように独立に学習させる。そして、各手法で選択された M 個の点を入力として復元ネットワークを学習させる。

表 1 に全ての手法の形状復元精度を示す。RS や SampleNet は空間的に一様に点を選択するため、局所形状の学習が困難である。RPCS は曲率の高い点を優先的に選択するため、局所形状の学習に有益な点を選択する。そのため、RPCS では RS や SampleNet に比べて全ての評価尺度で改善が見られた。しかし、RPCS は依然として空間的一様性の高い点を選択する傾向があるため、改善は限定的である。提案手法は全ての評価尺度において最も高い精度を達成した。提案手法は中心点の周囲の局所形状の詳細を保持する一方、中心点から離れた点も選択する。そのため、復元ネットワークがシーンの一部の大域形状と局所形状の両方を学習することを促す。結果として、従来手法に比べて大きな改善を実現した。

POCO には学習時と推論時の点密度の変化に対処するために、scene scaling と呼ばれる点の密度調整手法がある [7]。表 2 に scene scaling を適用した場合の形状復元

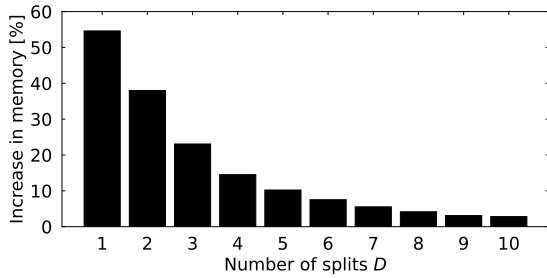


図 2. 分割数ごとのメモリ使用量の増加率.

精度を示す. いずれの手法においても, 表 1 に示される精度と比べて改善が見られないことがわかる. これは, 学習時には点の選択を行うため, 推論時と比べて点の分布に変化が生じていることに起因すると考えられる. したがって, scene scaling によって点密度の調整を行うだけでは, 形状復元精度を改善することは困難である.

また, メモリ使用量の観点から提案手法を評価する. 図 2 に RS を基準とした提案手法の学習時のメモリ使用量の増加率を示す. ここでは, 提案した分割統合法の有効性を評価するために分割数 D の値を変化させ, D ごとに増加率を示す. RS を使用する場合のメモリ使用量は 20.5 GB である. D の増加に伴い, メモリ使用量の増加率が急速に低減することが分かる. 全ての D にわたって復元ネットワークへの入力点数は共通である一方, D が大きいほど選択ネットワークへの入力点数が抑制される. 結果として, D の増加に伴いメモリ効率が改善される. したがって, 提案手法は $D = 10$ を用いた場合, メモリ使用量の 2.8% の増加と引き換えに, 表 1 に示されるように形状復元精度の大幅な改善を実現する.

図 3 に入力点群から選択された点を示す. 灰色と緑色の点はそれぞれ入力点群と選択点群を表す. 提案手法における赤色の点は中心点を表す. RS は無作為に点を選択する. SampleNet は RS に比べて空間的一様性の高い点を選択することが観察される. RPCS は高い曲率を持つ点を優先的に選択し, 平坦な領域からはあまり点を選択しない. 提案手法は中心点の周囲から集中的に点を選択する一方, 中心点から遠い点も選択することが分かる. この図における中心点は一例であり, 学習を通じて空間的一様に中心点を選択される.

図 4 に復元された形状を示す. この図では, 上段はシーンの全体図, 下段は一部の拡大図を表す. 従来手法は局所形状の復元精度が低く, 細部に歪みが発生している. 一方, 提案手法は従来手法に比べて高精度に局所形状を復元できることが分かる. これは提案手法によって, 復元ネットワークがシーンの詳細な局所形状を学習したためである.

4.3 構成要素の影響

表 3 に提案手法の構成要素の影響を評価するための復元精度の比較を示す. 一行目は選択ネットワークの代わりに中心点からの k 近傍探索によって選択点群 Q を構築する場合の結果を示している. これは図 5 (a) に示すような局所的に密集した選択点群を構築するため, 復元ネットワークはシーンの一部の領域形状を学習することが困難となる. 結果として, 復元された形状にはしばしば図 5 (b) に示すような大きな歪が生じる. 二行目は入

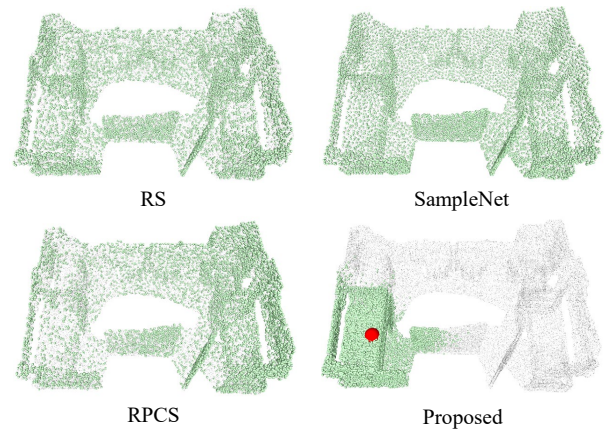


図 3. 選択された点の可視化. 灰色, 緑色, 赤色の点はそれぞれ入力点群, 選択された点群, 提案手法における中心点を表す.

表 3. 提案手法の構成要素の影響.

Variant	IoU \uparrow	CD \downarrow	NC \uparrow
k -NN search	0.647	0.02182	0.896
Uniform query	0.852	0.00306	0.930
Remove \mathcal{L}_{mse}	0.804	0.00313	0.919
Remove \mathcal{L}_{rep}	0.918	0.00294	0.944
Remove \mathcal{L}_{task}	0.923	0.00293	0.947
Complete	0.929	0.00291	0.949

力点とクエリ点を関連付けず, 無作為にクエリ点を選択する場合の結果を示している. この場合には選択された点とクエリ点の分布の差異に起因して, 復元精度が低下する. 三行目から五行目は各損失項を除外した場合の結果を示している. タスク損失を除外することは, 選択ネットワークの学習にはタスク損失を使用せず, 復元ネットワークの学習には使用することを意味する. これらの損失を除外した場合にも, 復元性能の低下が見られた. MSE 損失を除外すると空間的一様に点の選択を行うようになるため, 局所形状の詳細が失われる. 一方, 反発損失を除外する場合, 選択された点が中心点の周辺に集中するために, 大域形状を捉えにくくなる. また, タスク損失を除外する場合, 選択ネットワークが形状復元に対して最適化された点選択を行わなくなる. したがって, 提案手法が最高の精度を達成する.

5 まとめ

本稿では, ニューラル場に基づく形状復元手法の拡張性を改善するための点選択手法を提案した. 提案手法はタスク損失を用いて選択ネットワークを学習させることにより, 形状復元に最適化された点を選択する. また, 選択ネットワークがシーンの一部の領域から点を選択することを促すために, 起点となる中心点を導入した. さらに, 選択ネットワークへの入力点数を抑制しながら点を選択するために, 点群の分割統合法も導入した. 実験結果は, 提案手法が学習時のメモリ使用量の増加を回避しながら形状復元手法の性能を改善することを示した. 今後は, ground truth なしで復元ネットワークを学習させるために, 提案手法を拡張する予定である.

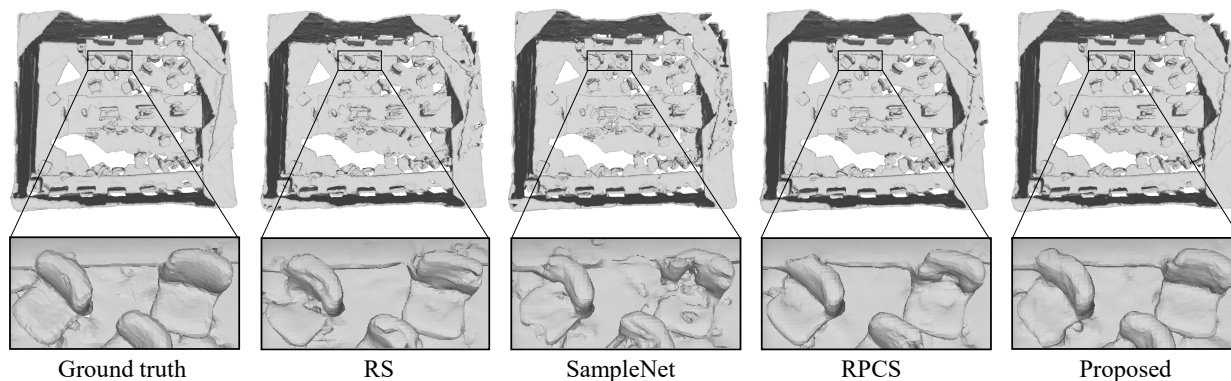
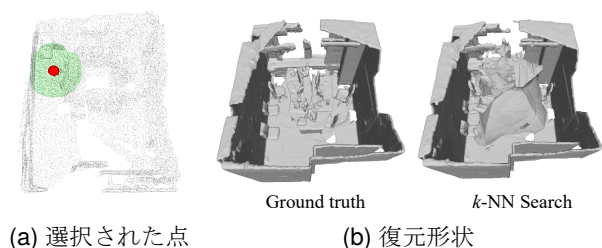


図4. 復元形状. 上段はシーンの全体図, 下段は一部の拡大図を表す.



(a) 選択された点

(b) 復元形状

図5. k 近傍探索によって点を選択する場合の結果.

参考文献

- [1] D. Cohen-Steiner and F. Da, "A greedy Delaunay-based surface reconstruction algorithm," *The Visual Computer*, vol.20, pp.4–16, 2004.
- [2] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM TOG*, vol.32, no.3, pp.1–13, 2013.
- [3] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," *Proc. of CVPR*, pp.4460–4470, 2019.
- [4] J.J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," *Proc. of CVPR*, pp.165–174, 2019.
- [5] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," *Proc. of ECCV*, pp.523–540, Springer, 2020.
- [6] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser, *et al.*, "Local implicit grid representations for 3D scenes," *Proc. of CVPR*, pp.6001–6010, 2020.
- [7] A. Boulch and R. Marlet, "POCO: Point convolution for surface reconstruction," *Proc. of CVPR*, pp.6302–6314, 2022.
- [8] T. Groueix, M. Fisher, V.G. Kim, B.C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," *Proc. of CVPR*, pp.216–224, 2018.
- [9] P.S. Wang, C.Y. Sun, Y. Liu, and X. Tong, "Adaptive O-CNN: A patch-based deep representation of 3D shapes," *ACM TOG*, vol.37, no.6, pp.1–11, 2018.
- [10] Y. Luo, Z. Mi, and W. Tao, "DeepDT: Learning geometry from Delaunay triangulation for surface reconstruction," *Proc. of AAAI*, pp.2277–2285, 2021.
- [11] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "SA-ConvOnet: Sign-agnostic optimization of convolutional occupancy networks," *Proc. of ICCV*, pp.6504–6513, 2021.
- [12] R. Chabra, J.E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, "Deep local shapes: Learning local SDF priors for detailed 3D reconstruction," *Proc. of ECCV*, pp.608–625, Springer, 2020.
- [13] C. Moenning and N.A. Dodgson, "Fast marching farthest point sampling," tech. rep., University of Cambridge, Computer Laboratory, 2003.
- [14] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM TOG*, vol.32, no.1, pp.1–12, 2013.
- [15] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE TSP*, vol.66, no.3, pp.666–681, 2017.
- [16] J. Qi, W. Hu, and Z. Guo, "Feature preserving and uniformity-controllable point cloud simplification on graph," *Proc. of ICME*, pp.284–289, IEEE, 2019.
- [17] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," *Proc. of CVPR*, pp.2760–2769, 2019.
- [18] I. Lang, A. Manor, and S. Avidan, "SampleNet: Differentiable point cloud sampling," *Proc. of CVPR*, pp.7578–7588, 2020.
- [19] Y. Ye, X. Yang, and S. Ji, "APNet: Attention based point cloud sampling," *Proc. of BMVC*, BMVA Press, 2022.
- [20] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Stoll, and C. Theobalt, "PatchNets: Patch-based generalizable deep implicit 3D shape representations," *Proc. of ECCV*, pp.293–309, Springer, 2020.
- [21] Z. Chen, Y. Zhang, K. Genova, S. Fanello, S. Bouaziz, C. Häne, R. Du, C. Keskin, T. Funkhouser, and D. Tang, "Multiresolution deep implicit functions for 3D shape representation," *Proc. of ICCV*, pp.13087–13096, 2021.
- [22] M. Pauly, M. Gross, and L.P. Kobbelt, "Efficient simplification of point-sampled surfaces," *IEEE VIS*, pp.163–170, IEEE, 2002.
- [23] N. Leal, E. Leal, and S.T. German, "A linear programming approach for 3D point cloud simplification," *IAENG IJCS*, vol.44, no.1, pp.60–67, 2017.
- [24] R.A. Potamias, G. Bouritsas, and S. Zafeiriou, "Revisiting point cloud simplification: A learnable feature preserving approach," *Proc. of ECCV*, pp.586–603, Springer, 2022.
- [25] B. Ma, Z. Han, Y.S. Liu, and M. Zwicker, "Neural-Pull: Learning signed distance function from point clouds by learning to pull space onto surface," *Proc. of ICML*, pp.7246–7257, PMLR, 2021.
- [26] L. Yu, X. Li, C.W. Fu, D. Cohen-Or, and P.A. Heng, "PU-Net: Point cloud upsampling network," *Proc. of CVPR*, pp.2790–2799, 2018.
- [27] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," *Proc. of CVPR*, pp.5828–5839, 2017.
- [28] J. Huang, H. Su, and L. Guibas, "Robust watertight manifold surface generation method for ShapeNet models," *arXiv preprint arXiv:1802.01698*, 2018.
- [29] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.