

サンプリングに基づく点群からの3次元形状復元の高精度化

High Accuracy 3D Shape Reconstruction from Point Clouds based on Sampling

松崎 康平[†]
Kohei Matsuzaki

野中 敬介[†]
Keisuke Nonaka

概要

本稿では、点群から3次元形状を高精度に復元することを目的とした点のサンプリング手法を提案する。深層学習に基づく形状復元手法ではメモリ容量の制限に対処するために空間的に一様なサンプリングを行うため、詳細な形状を学習することが困難である。この問題を解決するために、提案手法はニューラルネットワークを用いて部分的な領域から詳細な形状を表す点をサンプリングする。また、サンプリングの際にニューラルネットワークへの入力点数を抑制することにより、メモリ使用量の増加を防ぐための学習パイプラインも導入する。屋内シーンを表す実環境データセットを用いて定量および定性評価実験を行い、提案手法の有効性を確認した。

1 はじめに

LiDAR センサや深度センサなどの3次元センサを用いた点群取得技術の普及に伴い、点群からの3次元形状復元手法の検討が重要な研究課題の一つとなっている。形状復元に対しては、数理最適化に基づく手法 [1-6] や離散的な表現形式を用いた深層学習に基づく手法 [7-9] が数多く提案されている。近年では、ニューラル場に基づく手法 [10-16] が高精度に形状を復元できることから多くの注目を集めている。これらの手法は3次元点の座標から二値の占有確率 [10] や符号付き距離 [11] を予測し、マーチングキューブ法 [17] を用いて物体の形状を復元する。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手法は3次元形状全体から単一の特徴を抽出する [10, 11]。これらの手法は局所的な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。復元対象をより大きなシーンに拡張することを目指し、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 [12, 18]。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 [12, 18]。近年、特徴と点の直接的な接続を保持するために、点畳み込みを用いて点ごとに特徴を抽出する手法が提案された [15, 16]。しかし、これらの手法は入力点群の点数に伴ってメモリ使用量が増加するため、利用可能なメモリ容量によって拡張性が制限される。

本稿ではニューラル場に基づく形状復元ネットワークの学習における拡張性を改善するための点群サンプリング手法を提案する。提案手法では、シーンの部分領域の大域形状と局所形状の両方を表す点群を用いて復元ネットワークを学習させることを目指す。このために、部分

領域からのサンプリングの起点となるシード点を考慮したサンプリングネットワークを提案する。サンプリングネットワークはシード点からの距離によって重み付けられた特徴を用いて点をサンプリングする。さらに、サンプリングにおけるメモリ使用量を抑制するための学習パイプラインも提案する。提案手法は、タスク損失を用いてサンプリングネットワークと復元ネットワークを共同で学習させる。また、サンプリング操作を微分可能にするために、予測されたサンプリングスコアを用いて修正された損失関数を設計する。サンプリングされた点群は完全な入力形状を保持せず、部分的な形状を表す。しかし、空間的一様に選択されたシード点により、学習を通じて入力形状のほとんどの領域がカバーされる。提案手法のこのような部分的なサンプリングは、処理可能な入力点数が制限されている場合に、復元対象を大規模なシーンに拡張するために効果的である。

本研究の主な貢献は以下にまとめられる。

- 学習可能なサンプリングネットワークからサンプリングされた点群を用いて3次元形状表現としてのニューラル場を学習させる手法を提案した。
- シーンの部分領域から点をサンプリングするために、サンプリングの起点となるシード点を考慮したサンプリングネットワークを提案した。
- 実環境データセットを用いた評価実験により、提案手法が従来手法に比べて形状復元の高精度化に効果的な点群をサンプリングすることを示した。

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群サンプリングに関連する研究を概説する。第3節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群サンプリング手法を提案する。第4節では屋内シーンを表す実環境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。

2 関連研究

2.1 形状復元

本節では、ニューラル場に基づく点群からの形状復元手法に着目する。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 [10, 11, 19] 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 [12, 13, 18, 20-23]。これらの手法はボクセル格子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

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

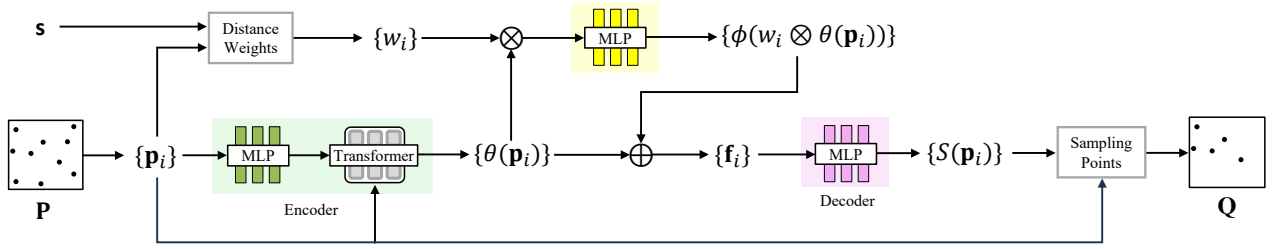


図 1. サンプリングネットワークの概要. サンプリングネットワークは入力点群 P とシード点 s を用いてサンプリングされた点群 Q を構築する.

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 は点畳み込みに基づいて入力点ごとに特徴を抽出した [15]. 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、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. さらに、ALTO は空間表現力を維持しながらデコーダネットワークの計算負荷を低減するために、点特徴と格子特徴を交互に切り替え、最終的に格子特徴に収束させることを提案した [16]. これら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 点群サンプリング

点群のデータ量や伝送量の削減、点群を用いた任意のタスクの計算効率の改善などを目的として、点群のサンプリングが広く利用されている. 最も一般的な手法は、random sampling (RS) と farthest point sampling (FPS) である [24, 25]. これらの手法は空間的に一様な方法で点をサンプリングするように設計されており、詳細な形状が失われる可能性がある. エッジや輪郭、曲率のような特定の幾何学的特性に焦点を当てて点群をサンプリングする手法も提案されている [26–33]. RPCS はグラフニューラルネットワークを用いて入力点群の大域形状を保持しながら、曲率の高い点を優先的にサンプリングする [33]. ただし、サンプリング処理は後段のタスクとは独立しており、タスクに対して最適な性能を実現するとは限らない. 後段のタスクで最適な性能を得るために、学習に基づく点群サンプリング手法が提案された [28, 34–39]. S-Net は後段のタスクに最適化された点を生成し、生成された点群を入力点群と照合することによってサンプリングされた点を取得する [34]. SampleNet は S-Net を拡張するために、微分可能な最近傍マッチングの近似を導入した [35]. 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、その後凍結されたタスクネットワークを用いてサンプリング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された点群を使用することにより、タスクネットワークの計算効率を改善する. したがって、これらの手法は計算資源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さない.

3 提案手法

本研究で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域形状と局所形状の両方を学習するように点をサンプリングすることを目指す. この目的のために、部分的な領域からのサンプリングのための起点となるシード点を考慮したサンプリングネットワークを提案する. また、サンプリングネットワークによってサンプリングされた点群を用いて復元ネットワークを学習させる学習パイプラインも提案する. ここでは、サンプリングネットワークと形状復元ネットワークはタスク損失を用いて共同で学習させる. 以下、サンプリングネットワークについて第 3.1 節で、学習パイプラインについて第 3.2 節で、損失関数について第 3.3 節で説明する.

3.1 サンプリングネットワーク

図 1 にサンプリングネットワークの概要を示す. このネットワークは点群 $P = \{p_i \in \mathbb{R}^3\}_{i=1}^N$ とシード点 $s \in \mathbb{R}^3$ を入力とし、サンプリングされた点群 $Q = \{q_i \in \mathbb{R}^3\}_{i=1}^M$ ($Q \subseteq P$) を出力する. ここで、シード点は任意の座標を持つ 3 次元点である. 以下、サンプリングネットワークの構成要素について説明する.

3.1.1 エンコーダ

入力点群 P に対して、サンプリングネットワークは θ でパラメータ化されるエンコーダを用いて特徴を抽出する. エンコーダは multi-layer perceptrons (MLP) と sparse transformer (ST) [40, 41] で構成される. MLP は 3 つの全結合層とそれに続く ReLU 活性化関数 [42] で構成され、入力点群から特徴を抽出する. ST は各点の座標と特徴を入力とし、入力された特徴と同次元の特徴を出力する. i 番目の点 p_i に対する特徴を $\theta(p_i)$ と表記する.

3.1.2 距離重み

特徴空間においてシード点と入力点群の各点の関連性を表すために、距離重みを導入する. 距離尺度にはユークリッド距離を採用する. i 番目の特徴に対する重み w_i は、ガウス関数を用いてシード点からの距離に応じて次式で計算される.

$$w_i = e^{-\frac{\|s-p_i\|^2}{\sigma^2}} \quad (1)$$

ここで s はシード点、 p_i は i 番目の入力点、 $\|\cdot\|$ は L2 ノルム、 σ は平滑化パラメータである. そして、重み w_i を特徴 $\theta(p_i)$ に乗算することにより、重み付き特徴を得る. さらに、 ϕ でパラメータ化される MLP を用いて重み付き特徴を洗練する.

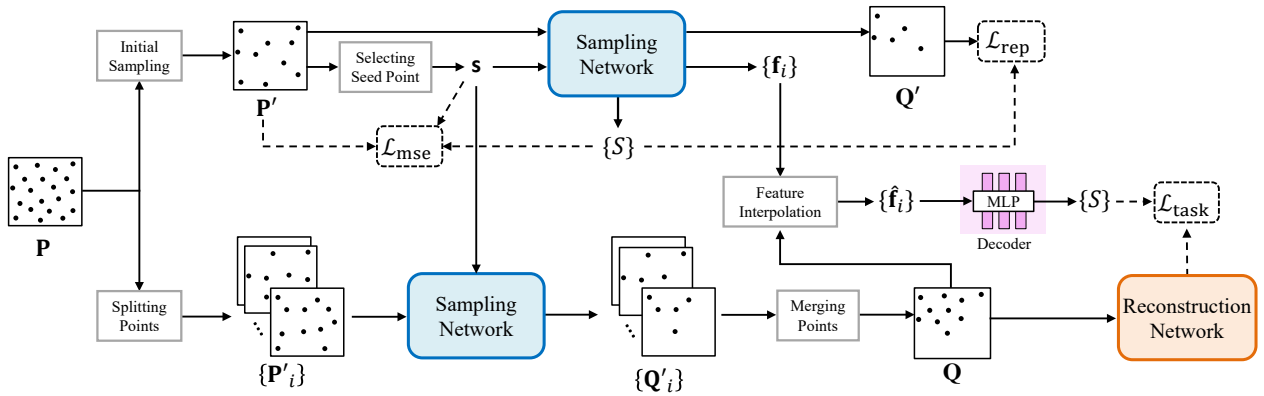


図 2. 提案手法の学習パイプライン. 本パイプラインでは入力点群 \mathbf{P} からサンプリングされた点群 \mathbf{Q} を復元ネットワークへ入力する.

3.1.3 デコーダ

デコーダは特徴から点 \mathbf{p}_i に対するサンプリングスコア $S(\mathbf{p}_i) \in \mathbb{R}$ を予測する. エンコーダで抽出された特徴と, 重み付き特徴を結合することによって生成された次式の特徴 \mathbf{f}_i をデコーダへ入力する.

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

ここで \oplus は結合操作, \otimes は要素ごとの積を表す. デコーダは最後の予測層を除くすべての全結合層の後にバッチ正規化 [43] と ReLU 活性化関数 [42] を適用する MLP で構成される.

3.1.4 点のサンプリング

デコーダで予測されたサンプリングスコアは各点がサンプリングされた点群に存在する確率を表す. 予測されたサンプリングスコア $S(\mathbf{p}_i)$ の上位 M 個に対応している点の集合 $\{\mathbf{q}_i\}_{i=1}^M$ を選択し, サンプリングされた点群 \mathbf{Q} とする. 最終的に, サンプリングネットワークは \mathbf{Q} を出力する.

3.2 学習パイプライン

提案手法は入力点群 $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$ からサンプリングネットワークを用いてサンプリングされた点群 $\mathbf{Q} = \{\mathbf{q}_i \in \mathbb{R}^3\}_{i=1}^{M'}$ を構築し, \mathbf{Q} を復元ネットワークへ入力する. これを実現するための単純な方法は \mathbf{P} をサンプリングネットワークへ入力し, \mathbf{Q} を出力することである. しかし, この方法では \mathbf{P} の点数が多い場合にはメモリ使用量が増加するため, 処理可能な点数が制限される. この問題に対処するために, 点数を抑制した点群をサンプリングネットワークへ入力し, \mathbf{Q} を構築する学習パイプラインを提案する. 図 2 に提案手法の学習パイプラインを示す. このパイプラインは図 2 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ままサンプリングネットワークを学習させ, 下部のブランチは復元ネットワークへ入力するために点数の抑制された点群から \mathbf{Q} を構築する. これらのブランチでは, サンプリングネットワークのモデルパラメータは共有される. また, 2 つのブランチの間では, 上部のブランチで得られた特徴から \mathbf{Q} に対するサンプリングスコアを予測する. 以下, 学習パイプラインの構成要素について説明する.

3.2.1 初期サンプリング

点数の抑制された点群を構築するために, はじめに RS を用いて \mathbf{P} から点群 $\mathbf{P}' = \{\mathbf{p}'_i \in \mathbb{R}^3\}_{i=1}^{N'}$ をサンプリングする. そして, 点群 \mathbf{P}' をサンプリングネットワークへ入力し, さらにサンプリングされた点群 $\mathbf{Q}' = \{\mathbf{q}'_i \in \mathbb{R}^3\}_{i=1}^{M'}$ を得る. 点群 \mathbf{Q}' はサンプリングネットワークに関わる損失関数を計算するために使用される. 初期サンプリングとサンプリングネットワークのサンプリング率をそれぞれ r_{init} および r_{nw} とする. このとき, 点群 \mathbf{P}' と \mathbf{Q}' の点数はそれぞれ $N' = \text{round}(r_{\text{init}}N)$ と $M' = \text{round}(r_{\text{nw}}N')$ と表される.

3.2.2 シード点の選択

復元ネットワークがシーンの一部の詳細な形状を学習することを促すために, シード点を導入する. サンプリングネットワークがシード点の周囲の局所領域から点を集中的にサンプリングすることにより, 詳細な形状を保持することができる. 提案手法では点群 \mathbf{P}' が与えられた場合, 無作為にその中から 1 点を選択し, シード点 $\mathbf{s} \in \mathbb{R}^3$ とする. シード点の位置は固定されておらず, 学習中に反復的な処理を通じて空間の様にシード点を選択される.

3.2.3 点群の分割

提案手法はサンプリングネットワークへの入力点数を抑制しつつ, 入力点群 \mathbf{P} からサンプリングされた点群 \mathbf{Q} を構築することを目指す. そのために, 点群 \mathbf{P} を大域形状を保持する D 個の点群に分割する. 具体的には, 点群 \mathbf{P} からこれまでにサンプリングされた点を除外しながら反復的に RS を適用することによって D 個の点群を生成する. 得られた点群の集合を $\{\mathbf{P}'_i\}_{i=1}^D$ と表記する. ここでは初期サンプリング率 r_{init} を使用する. したがって, $D = \text{round}(1/r_{\text{init}})$ となる.

3.2.4 点群の統合

入力点数を抑制するために, \mathbf{P} の代わりに各 \mathbf{P}'_i をサンプリングネットワークへ入力する. 各 \mathbf{P}'_i に対するサンプリングは, サンプリングネットワークのモデルパラメータを凍結させながら逐次的に実行する. ここでは, $\{\mathbf{P}'_i\}_{i=1}^D$ のすべてに対して共通のシード点を使用する. そして, D 個のサンプリングされた点群の集合 $\{\mathbf{Q}'_i \in \mathbb{R}^{M' \times 3}\}_{i=1}^D$ が得られる. 最後に, これらの点群を統合することでサンプリングされた点群 $\mathbf{Q} \in \mathbb{R}^{M \times 3}$ を構

築する。この点群は元々の点群 \mathbf{P} からのサンプリング結果の近似とみなせる。

3.2.5 特徴補間

損失関数の計算に利用するために、 \mathbf{Q} に対するサンプリングスコアを予測する。下部のブランチで予測されたサンプリングスコアは、モデルパラメータが凍結されているため、サンプリングネットワークの学習に用いることができない。代わりに、上部のブランチの特徴 $\{\mathbf{f}_i\}_{i=1}^{N'}$ から次式の特徴補間 [44] によって、 \mathbf{Q} に対応する特徴 $\{\hat{\mathbf{f}}_i\}_{i=1}^M$ を取得する。

$$\hat{\mathbf{f}}_i = \frac{\sum_{j=1}^k d(\mathbf{q}_i, \mathbf{p}'_j) \mathbf{f}_j}{\sum_{j=1}^k d(\mathbf{q}_i, \mathbf{p}'_j)} \quad (3)$$

ここで $d(\mathbf{q}_i, \mathbf{p}'_j) = 1/\|\mathbf{q}_i - \mathbf{p}'_j\|$ は \mathbf{q}_i と \mathbf{p}'_j の間の距離の逆数、 \mathbf{p}'_j は \mathbf{q}_i の j 番目の近傍点、 k は近傍点数である。 $\hat{\mathbf{f}}_i$ は上部のブランチのサンプリングネットワークと共通のデコーダへ入力される。結果として、点 \mathbf{q}_i に対するサンプリングスコア $S(\mathbf{q}_i)$ が得られる。

3.2.6 復元ネットワーク

サンプリングされた点群 \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、入力形状のバウンディングボックス内 [10] や表面付近 [14] から一様にサンプリングされたクエリ点群を必要とする。提案手法ではサンプリングされた点群が部分的な領域を表すようになるため、サンプリングされた点群とクエリ点群の分布に差異が生じる恐れがある。これに対処するために、サンプリングされた点とクエリ点を関連付ける。具体的には、入力点にランダム変位 $\mathbf{n}_i \in \mathbb{R}^3$ を加算することにより、クエリ点 $\tilde{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{n}_i$ を作成する。ここで $\mathbf{n}_i \sim \mathcal{N}(0, \Sigma)$ であり、 $\Sigma \in \mathbb{R}^{3 \times 3}$ は対角成分が全て v の対角共分散行列である。そして、あらかじめ入力点 \mathbf{p}_i とそれらから作成されたクエリ点を関連付ける。学習時には、サンプリングされた入力点に関連しているクエリ点を損失関数の計算に利用する。

3.3 損失関数

提案手法ではサンプリングされた点群を復元ネットワークへ入力しつつ、サンプリングネットワークと復元ネットワークを共同で学習させる。点のサンプリングに対する損失関数として、MSE 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 $\mathcal{L}_{\text{task}}$ を使用する。提案手法ではサンプリング操作を微分可能にするために、これらの損失をサンプリングスコアを用いて修正する。したがって、以下の損失 \mathcal{L} を最小化するようにネットワークを学習させる。

$$\mathcal{L} = \mathcal{L}_{\text{mse}} + \alpha \mathcal{L}_{\text{rep}} + \beta \mathcal{L}_{\text{task}} \quad (4)$$

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

3.3.1 MSE 損失

サンプリングネットワークにシード点の周囲の部分領域から集中的に点をサンプリングすることを促すために、シード点と入力点群 \mathbf{P} の間の MSE を損失として使用する。点 \mathbf{p}_i に対する予測されたサンプリングスコア

$S(\mathbf{p}_i)$ を用いて、MSE 損失を次式で定義する。

$$\mathcal{L}_{\text{mse}} = \frac{1}{N} \sum_{i=1}^N g(S(\mathbf{p}_i)) \|\mathbf{s} - \mathbf{p}_i\|^2 \quad (5)$$

ここで $S(\mathbf{p}_i)$ は点 \mathbf{p}_i に対する予測されたサンプリングスコア、 $g(\cdot)$ はシグモイド関数である。

3.3.2 反発損失

サンプリングネットワークがシード点から遠く離れた点もサンプリングすることを促すために反発損失 [45, 46] を導入する。この損失はサンプリングされた点群 \mathbf{Q} において、各点の近傍点との距離を遠ざける効果を持つ。MSE 損失と同様に、サンプリングスコアを用いて反発損失を次式で定義する。

$$\mathcal{L}_{\text{rep}} = \frac{1}{MK} \sum_{i=1}^M \sum_{j=1}^K g(S(\mathbf{q}_j)) \eta(\|\mathbf{q}_i - \mathbf{q}_j\|) \omega(\|\mathbf{q}_i - \mathbf{q}_j\|) \quad (6)$$

ここで \mathbf{q}_j は \mathbf{Q} 内の自身を除く j 番目の近傍点、 $\eta(a) = -a$ 、 $\omega(a) = e^{-a^2/b^2}$ 、 b は調整可能なパラメータである。

3.3.3 タスク損失

後段のタスクである形状復元に寄与する点のサンプリングを促すためにタスク損失を導入する。多くの復元ネットワークに対して使用される二値クロスエントロピー損失をタスク損失とし、次式で定義する。

$$\mathcal{L}_{\text{task}} = -\frac{1}{M} \sum_{i=1}^M g(S(\mathbf{q}_i)) [\mathbf{o}_i \log(\hat{\mathbf{o}}_i) + (1 - \mathbf{o}_i) \log(\hat{\mathbf{o}}_i)] \quad (7)$$

ここで \mathbf{o}_i および $\hat{\mathbf{o}}_i$ はそれぞれ i 番目のクエリ点の占有確率の真値と予測値である。各クエリ点はサンプリング点に関連付けられているため、タスク損失によってサンプリングネットワークは正確な占有確率の予測に寄与する点をサンプリングすることを促される。なお、二値クロスエントロピー損失以外であっても、点ごとに計算可能な任意の損失はタスク損失として使用可能である。

4 評価実験

4.1 実験設定

シーンの形状復元に関する実験によって提案手法を評価するために、ScanNet-v2 [47] データセットを使用する。ScanNet-v2 データセットは実世界の屋内環境で RGB-D センサを用いて取得された大規模な 3 次元形状モデルで構成される。公式の分割リストに従い、データセット内の 1201 個、312 個、100 個の 3 次元形状モデルをそれぞれ学習、検証、評価に使用する。前処理として多様体形状生成手法 [48] を用いて 3 次元形状モデルを水密モデルに変換する。各モデルの表面から一様にサンプリングし、単位立方体を用いて正規化した点群を入力点群とする。復元ネットワークのために物体表面付近からクエリ点をサンプリングし、3 次元形状に対する内部/外部情報を表す占有確率を計算する。

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、intersection over union (IoU)、Chamfer distance (CD)、normal consistency (NC) [10] 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 [49] を用いて実装し、学習率 10^{-3}

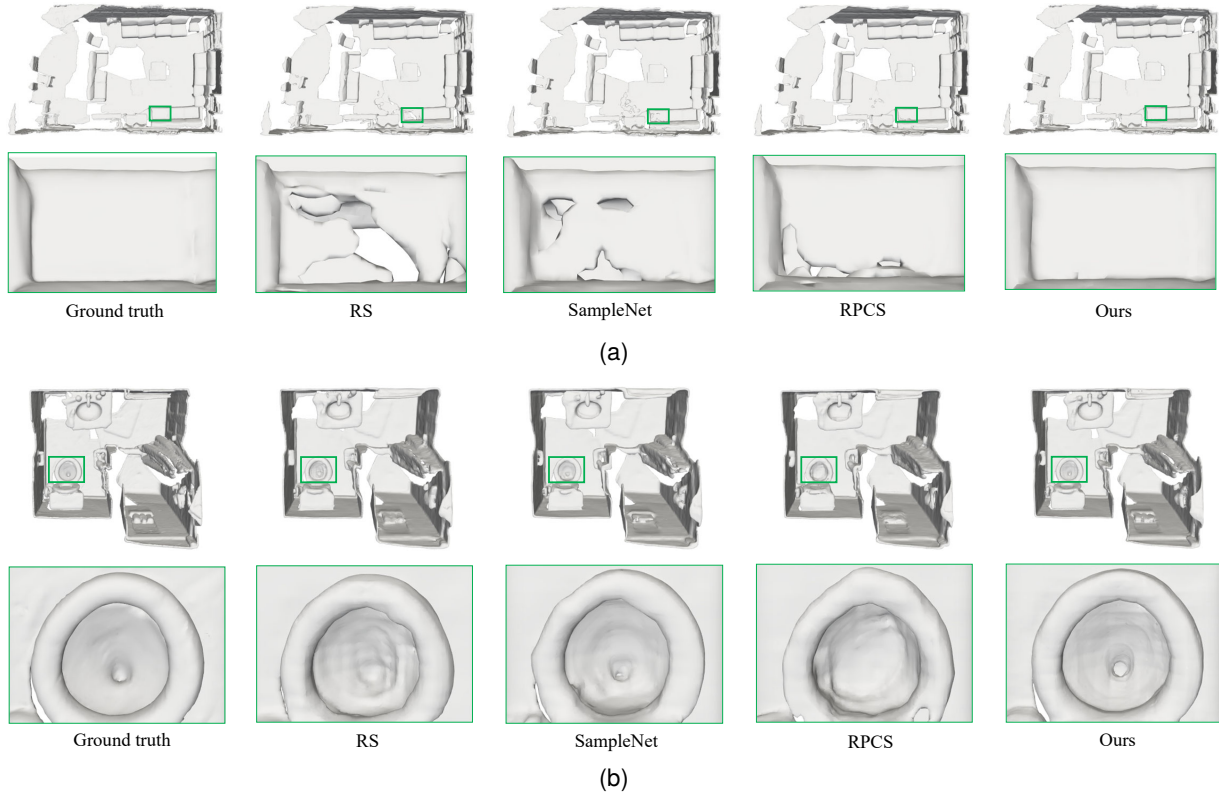


図 3. ScanNet-v2 データセットを用いた形状復元結果の比較. 上段は全体図, 下段は一部の拡大図を表す. (a) は POCO, (b) は ALTO を用いた場合の結果を表す.

の Adam optimizer [50] を用いてネットワークを学習させる. バッチサイズは 8 に設定し, 100k iterations の学習を行う. 式 (4) において $\alpha = 1.0$, $\beta = 1.0$ とする. 初期サンプリングおよびサンプリングネットワークのサンプリング率を $r_{\text{init}} = 0.1$, $r_{\text{nw}} = 0.1$ に設定する. したがって点群の分割における分割数は $D = 10$ である. 点群 \mathbf{P} , \mathbf{P}' , \mathbf{Q} , \mathbf{Q}' の点数はそれぞれ $N = 100\text{k}$, $N' = 10\text{k}$, $M' = 1\text{k}$, $M = 10\text{k}$ とする. 復元ネットワークの学習に使用するクエリ点の点数は 10k とする. ランダム変位の作成には $v = 0.01$ を用いる. 式 (1) においては $\sigma = 1.0$, 式 (3) においては $k = 10$, 式 (6) においては $K = 10$, $b = 1.0$ に設定する. 特徴 $\theta(\mathbf{p}_i)$, $\phi(w_i \otimes \theta(\mathbf{p}_i))$, \mathbf{f}_i , $\hat{\mathbf{f}}_i$ の次元はそれぞれ 64, 64, 128, 128 とする. ST で使用されるウィンドウサイズは 0.01 とする. 評価時には, 物体表面から一様にサンプリングされた 1k 点/ m^2 の点群を使用する. すべての実験は NVIDIA RTX A6000 GPU, Intel Core i9-13900K CPU (3.0 GHz), および 128 GB の RAM を搭載したコンピュータで実施する.

4.2 従来手法との比較

定量的および定性的な評価を通じて, 提案手法と従来の点群サンプリング手法を比較する. 比較手法には広く利用される RS を選択する. また, 学習に基づく最先端の手法である SampleNet [35] と RPCS [33] とも比較する. SampleNet の学習には事前に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

表 1. 形状復元精度の比較. CD は 10^2 が乗算される.

Method	POCO			ALTO		
	IoU \uparrow	CD \downarrow	NC \uparrow	IoU \uparrow	CD \downarrow	NC \uparrow
RS	0.782	0.321	0.931	0.768	0.346	0.906
SampleNet [35]	0.824	0.304	0.939	0.801	0.325	0.918
RPCS [33]	0.821	0.309	0.936	0.785	0.334	0.911
Ours	0.906	0.293	0.949	0.828	0.314	0.927

M 点をサンプリングするように 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 点をサンプリングするように独立に学習させる. そして, 各手法でサンプリングされた M 個の点を入力として復元ネットワークを学習させる.

ニューラル場に基づく形状復元のための最先端手法である POCO [15] と ALTO [16] を復元ネットワークとして選択し, 表 1 に形状復元精度を示す. 提案手法は, 両方の手法において最も優れた復元精度を達成する. RS は局所形状の詳細を失うため, すべての評価尺度にわたって最も低い性能を示す. SampleNet は空間的に一様な点をサンプリングするが, タスク損失を導入しているため RS に比べて優れた復元精度を達成する. RPCS は曲率の高い領域から点を集中的にサンプリングし, 局所形状の学習に有効な点群を構築する. そのため, RS と比べて性能の改善が確認される. しかし, 依然として空間的一様性の高い点をサンプリングする傾向があるため, 改善量は限定的である.

形状復元精度を定性的に評価するために, 図 3 に復元

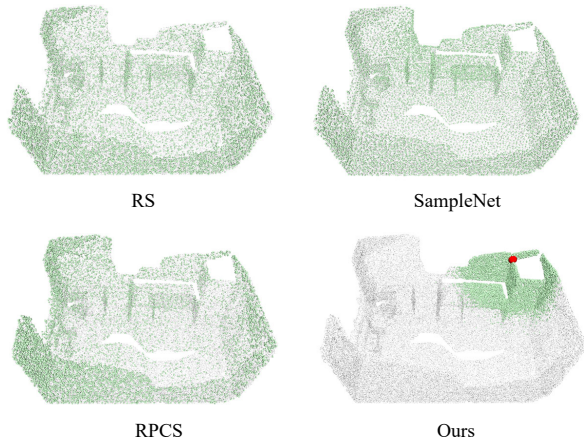


図 4. サンプリングされた点群の比較. 灰色, 緑色, 赤色の点はそれぞれ入力点群, サンプリングされた点群, シード点を表す.

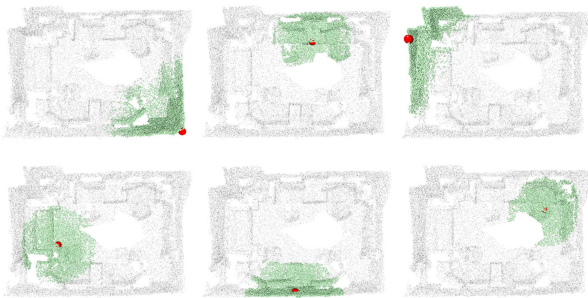


図 5. 提案手法において異なるシード点を用いてサンプリングされた点群.

結果を可視化する. 上段は全体図, 下段は一部の拡大図を表す. 全体図では全ての手法が大域的な形状を正確に復元できていることがわかる. 一方, 拡大図では比較手法は局所的な形状の復元精度が低く, 大きな歪みが発生することがわかる. 対照的に, 提案手法は他の手法に比べて局所的な形状をより正確に復元する. これは提案手法のシード点を用いたサンプリングネットワークによって, 復元ネットワークがシーンの一部の詳細な形状を学習することが可能になるためである.

図 4 は入力点群からサンプリングされた点群を示す. 灰色と緑色の点はそれぞれ入力点群とサンプリングされた点群を表す. また, 提案手法における赤色の点はシード点を表す. この図では, 入力点群, サンプリングされた点群, シード点はそれぞれ小, 中, 大の大きさの球体として描画する. RS は無作為に点をサンプリングする. SampleNet によりサンプリングされた点群は RS と比較して空間的な一様性が高いことがわかる. RPCS は曲率の高い領域からより多くの点をサンプリングし, 平坦な領域からはあまりサンプリングしない傾向が見られる. 提案手法はシード点の近くから集中的に点をサンプリングする一方で, シード点から遠い点もサンプリングしていることが分かる. 提案手法のこの特性は, 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し, 結果として復元精度を改善する.

表 2. 構成要素の影響. CD は 10^2 が乗算される.

Method	IoU \uparrow	CD \downarrow	NC \uparrow
k -NN sampling	0.899	0.581	0.935
Random query	0.876	0.338	0.937
Remove \mathcal{L}_{mse}	0.784	0.311	0.938
Remove \mathcal{L}_{rep}	0.901	0.306	0.947
Remove \mathcal{L}_{task}	0.905	0.295	0.948
Complete	0.906	0.293	0.949

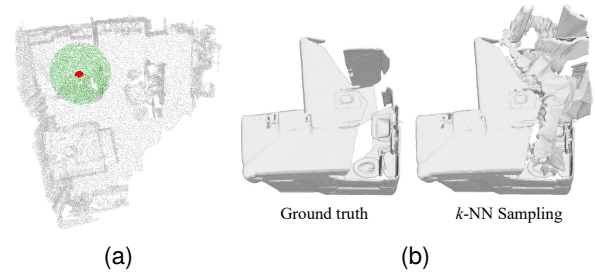


図 6. k 近傍サンプリングの結果. (a) はサンプリングされた点群, (b) は形状復元結果を表す.

図 4 におけるシード点は一例であり, その位置は固定されていない. 提案手法は, 図 5 に示すように学習時に様々なシード点を入力点群から無作為に選択する.

4.3 構成要素の影響

表 2 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. 本実験では復元ネットワークとして POCO を使用する. 一行目はサンプリングネットワークの代わりにシード点からの k 近傍探索によってサンプリングされた点群 Q を構築した場合の復元精度を示している. これは図 6 (a) に示されるような極端に局所的に集中した点群を構築するため, 復元ネットワークは大域形状をほとんど学習することができない. これにより, 復元された形状には図 6 (b) に示すような大きな歪みが発生する場合がある. 二行目は入力点とクエリ点の関連付けを行わず, 無作為にクエリ点を選択する場合の復元精度を示している. この場合にはサンプリングされた点とクエリ点の分布の違いに起因して, 復元精度が低下する. 表 2 の三行目から五行目は式 (4) から各損失項を除外した場合の復元精度を示している. タスク損失の除外は復元ネットワークの学習にはタスク損失を使用するが, サンプリングネットワークの学習には使用しないことを意味する. 図 7 は MSE 損失 \mathcal{L}_{mse} および反発損失 \mathcal{L}_{rep} を除外した場合のサンプリングされた点群を示している. MSE 損失を除外する場合, 図 7 (a) に示されるように空間的に一様なサンプリングを行うために, 局所的な詳細が失われる. 反発損失を除外する場合, 図 7 (b) に示されるようにサンプリングされた点がシード点周辺に集中するために, 大域形状の学習が困難になる. したがって, これらの損失を除外することにより, 復元精度が低下する. 表 2 の五行目よりタスク損失の除外も復元精度の低下をもたらすことがわかる. これはサンプリングネットワークが復元タスクに対して最適化されなくなるためである. したがって, 完全な提案手法が最も優れた復元精度を達成する.



図7. 特定の損失項を除外した場合のサンプリングされた点群。(a)は \mathcal{L}_{mse} を除外した場合、(b)は \mathcal{L}_{rep} を除外した場合を表す。

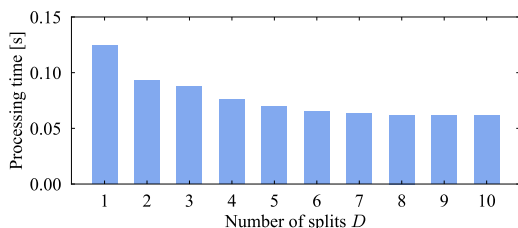


図8. 異なる分割数 D を用いた場合の処理時間。

4.4 計算効率の評価

時間効率およびメモリ効率の観点から提案手法を評価する。図8にサンプリングネットワークの処理時間を示す。ここで、処理時間は図2の上部と下部のブランチにおけるサンプリング時間の合計を表す。また、図9に前処理によってメモリ消費なしでサンプリングする場合に比べた、提案手法の学習時のメモリ使用量の増加率を示す。前処理の場合の復元ネットワークの学習に対するメモリ使用量は20.5 GBである。提案手法における分割数 D の影響を検証するために、これらの値を D の関数として描画する。図8より、提案手法の処理時間は D を大きくするほど短くなることがわかる。これは D が大きい場合には初期サンプリング率 r_{init} が小さくなり、上部のブランチにおけるサンプリングネットワークへの入力点数が減少するためである。図9より D を大きくするほどメモリ使用量の抑制に効果的であることがわかる。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 が大きいほどサンプリングネットワークへの入力点数は抑制される。したがって、 D の増加に伴いメモリ使用量が抑制される。 $D = 10$ を用いた場合の増加率は3.1%であり、メモリ使用量をほとんど増加させないことがわかる。これらの結果より、提案手法の学習パイプラインは時間とメモリの両方について効率的な処理を可能にすることがわかる。

さらに、学習パイプラインの変更がメモリ使用量に与える影響を図10に示す。単純な学習パイプライン(Naive)は、入力点群 \mathbf{P} を直接サンプリングネットワークに入力し、サンプリングされた点群 \mathbf{Q} を得るものである。このパイプラインでは、サンプリングネットワークに入力される点群の点数が多いため、メモリ使用量が大幅に増加する。モデルパラメータを凍結させずに図2の下部のブランチのみを用いる学習パイプライン(Lower branch)も考えられる。このパイプラインにおいても、サンプリングネットワークには合計で N 点が入力されるために、メモリ使用量が同様に増加する。対照

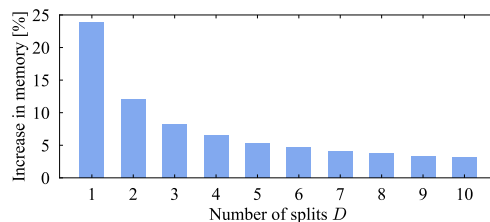


図9. 異なる分割数 D を用いた場合のメモリ使用量の増加率。

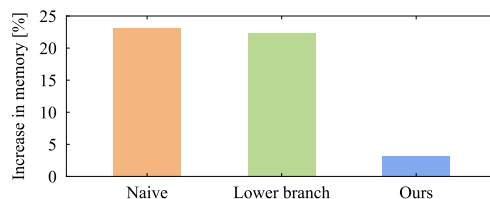


図10. 異なる学習パイプラインにおけるメモリ使用量の増加率の比較。

的に、提案手法は2つのブランチと特徴補間を用いてメモリ使用量を抑制しており、その有効性が確認できる。

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群サンプリング手法を提案した。シーンの一部の領域を表す点群をサンプリングするために、起点となるシード点を考慮したサンプリングネットワークを導入した。さらに、メモリ使用量を抑制しつつサンプリングされた点群を構築することを可能にする学習パイプラインも提案した。提案手法ではタスク損失を用いて、サンプリングネットワークと復元ネットワークを共同で学習させた。屋内シーンを表すデータセットを用いた実験では、提案手法が学習時のメモリ使用量の増加を防ぎつつ、形状復元精度を大幅に改善することが示された。今後は点の座標に加えて色やセマンティックラベルなどの他の種類の情報の利用を検討する。

謝辞

本研究成果は、国立研究開発法人情報通信研究機構(NICT)の委託研究(JPJ012368C06801)により得られたものです。

参考文献

- [1] D. Cohen-Steiner and F. Da, "A greedy Delaunay-based surface reconstruction algorithm," *Vis. Comput.*, vol.20, pp.4–16, 2004.
- [2] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *TVCG*, vol.5, no.4, pp.349–359, 1999.
- [3] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," *Proc. SGP*, pp.61–70, 2006.
- [4] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *TOG*, vol.32, no.3, pp.1–13, 2013.
- [5] L. Nan and P. Wonka, "Polyfit: Polygonal surface reconstruction from point clouds," *Proc. ICCV*, pp.2353–2361, 2017.
- [6] Y. Li, A. Dai, L. Guibas, and M. Nießner, "Database-assisted object retrieval for real-time 3D reconstruction," *Proc. CGF*, pp.435–446, Wiley Online Library, 2015.

- [7] T. Groueix, M. Fisher, V.G. Kim, B.C. Russell, and M. Aubry, “A papier-mâché approach to learning 3D surface generation,” *Proc. CVPR*, pp.216–224, 2018.
- [8] P.S. Wang, C.Y. Sun, Y. Liu, and X. Tong, “Adaptive O-CNN: A patch-based deep representation of 3D shapes,” *TOG*, vol.37, no.6, pp.1–11, 2018.
- [9] Y. Luo, Z. Mi, and W. Tao, “DeepDT: Learning geometry from Delaunay triangulation for surface reconstruction,” *Proc. AAAI*, pp.2277–2285, 2021.
- [10] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” *Proc. CVPR*, pp.4460–4470, 2019.
- [11] J.J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” *Proc. CVPR*, pp.165–174, 2019.
- [12] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” *Proc. ECCV*, pp.523–540, Springer, 2020.
- [13] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, “Local implicit grid representations for 3D scenes,” *Proc. CVPR*, pp.6001–6010, 2020.
- [14] 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. ICML*, pp.7246–7257, PMLR, 2021.
- [15] A. Boulch and R. Marlet, “POCO: Point convolution for surface reconstruction,” *Proc. CVPR*, pp.6302–6314, 2022.
- [16] Z. Wang, S. Zhou, J.J. Park, D. Paschalidou, S. You, G. Wetzstein, L. Guibas, and A. Kadambi, “ALTO: Alternating latent topologies for implicit 3D reconstruction,” *Proc. CVPR*, pp.259–270, 2023.
- [17] W.E. Lorensen and H.E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” *Siggraph Comput. Graph.*, vol.21, no.4, pp.163–169, 1987.
- [18] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, “SA-ConvOnet: Sign-agnostic optimization of convolutional occupancy networks,” *Proc. ICCV*, pp.6504–6513, 2021.
- [19] M. Michalkiewicz, J.K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, “Implicit surface representations as layers in neural networks,” *Proc. ICCV*, pp.4743–4752, 2019.
- [20] 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. ECCV*, pp.608–625, Springer, 2020.
- [21] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Stoll, and C. Theobalt, “PatchNets: Patch-based generalizable deep implicit 3D shape representations,” *Proc. ECCV*, pp.293–309, Springer, 2020.
- [22] 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. ICCV*, pp.13087–13096, 2021.
- [23] C. Chen, Y.S. Liu, and Z. Han, “Latent partition implicit with surface codes for 3D representation,” *Proc. ECCV*, pp.322–343, Springer, 2022.
- [24] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi, “The farthest point strategy for progressive image sampling,” *TIP*, vol.6, no.9, pp.1305–1315, 1997.
- [25] C. Moenning and N.A. Dodgson, “Fast marching farthest point sampling,” *tech. rep.*, University of Cambridge, Computer Laboratory, 2003.
- [26] M. Pauly, M. Gross, and L.P. Kobbelt, “Efficient simplification of point-sampled surfaces,” *Proc. VIS*, pp.163–170, IEEE, 2002.
- [27] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, “Edge-aware point set resampling,” *TOG*, vol.32, no.1, pp.1–12, 2013.
- [28] C. Wu, J. Zheng, J. Pfaff, and J. Beyerer, “Attention-based point cloud edge sampling,” *Proc. CVPR*, pp.5333–5343, 2023.
- [29] N. Leal, E. Leal, and S.T. German, “A linear programming approach for 3D point cloud simplification,” *IJCS*, vol.44, no.1, pp.60–67, 2017.
- [30] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, “Fast resampling of three-dimensional point clouds via graphs,” *TSP*, vol.66, no.3, pp.666–681, 2017.
- [31] J. Qi, W. Hu, and Z. Guo, “Feature preserving and uniformity-controllable point cloud simplification on graph,” *Proc. ICME*, pp.284–289, IEEE, 2019.
- [32] K. Zhang, S. Qiao, X. Wang, Y. Yang, and Y. Zhang, “Feature-preserved point cloud simplification based on natural quadric shape models,” *Appl. Sci.*, vol.9, no.10, p.2130, 2019.
- [33] R.A. Potamias, G. Bouritsas, and S. Zafeiriou, “Revisiting point cloud simplification: A learnable feature preserving approach,” *Proc. ECCV*, pp.586–603, Springer, 2022.
- [34] O. Dovrat, I. Lang, and S. Avidan, “Learning to sample,” *Proc. CVPR*, pp.2760–2769, 2019.
- [35] I. Lang, A. Manor, and S. Avidan, “SampleNet: Differentiable point cloud sampling,” *Proc. CVPR*, pp.7578–7588, 2020.
- [36] Y. Lin, Y. Huang, S. Zhou, M. Jiang, T. Wang, and Y. Lei, “DA-Net: Density-adaptive downsampling network for point cloud classification via end-to-end learning,” *Proc. ICPRAI*, pp.13–18, IEEE, 2021.
- [37] Y. Ye, X. Yang, and S. Ji, “APSNet: Attention based point cloud sampling,” *Proc. BMVC*, BMVA Press, 2022.
- [38] Y. Qian, J. Hou, Q. Zhang, Y. Zeng, S. Kwong, and Y. He, “Task-oriented compact representation of 3D point clouds via a matrix optimization-driven network,” *TCSVT*, 2023.
- [39] X. Wang, Y. Jin, Y. Cen, T. Wang, B. Tang, and Y. Li, “LightTN: Light-weight transformer network for performance-overhead tradeoff in point cloud downsampling,” *TMM*, 2023.
- [40] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, “Stratified transformer for 3D point cloud segmentation,” *Proc. CVPR*, pp.8500–8509, 2022.
- [41] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, “Spherical transformer for lidar-based 3D recognition,” *Proc. CVPR*, 2023.
- [42] V. Nair and G.E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” *Proc. ICML*, pp.807–814, 2010.
- [43] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *Proc. ICML*, pp.448–456, PMLR, 2015.
- [44] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Proc. NeurIPS*, vol.30, 2017.
- [45] L. Yu, X. Li, C.W. Fu, D. Cohen-Or, and P.A. Heng, “PU-Net: Point cloud upsampling network,” *Proc. CVPR*, pp.2790–2799, 2018.
- [46] L. Yu, X. Li, C.W. Fu, D. Cohen-Or, and P.A. Heng, “EC-Net: An edge-aware point set consolidation network,” *Proc. ECCV*, pp.386–402, 2018.
- [47] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” *Proc. CVPR*, pp.5828–5839, 2017.
- [48] J. Huang, H. Su, and L. Guibas, “Robust watertight manifold surface generation method for ShapeNet models,” *arXiv:1802.01698*, 2018.
- [49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” *Proc. NeurIPS*, vol.32, 2019.
- [50] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980*, 2014.