CI-004

八分木圧縮における占有予測に基づく点群の歪補正

Distortion Correction of Point Cloud based on Occupancy Prediction in Octree Compression

松﨑 康平† 小森田 賢史†
Kohei Matsuzaki Satoshi Komorita

概要

本稿では点群圧縮による歪を補正するために，ボクセル化された点群の超解像手法を提案する．点群圧縮手法の多くは点群のボクセル化を伴う八分木圧縮を利用しており，復号点群にはボクセル化に起因する歪が生じる．提案手法は圧縮に用いたボクセルよりも高解像度なボクセルの占有確率を予測することにより，点群を高解像度化する．効率的な予測のために，スパース畳み込みに基づく深層ニューラルネットワークを利用する．また，八分木表現との整合性をとるために，全てのボクセルの中で少なくとも 1 つが占有と予測されることを保証する動的閾値を導入する．そして，大規模な点群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検証する．

1 はじめに

3 次元点群データの計測技術の普及により，点群圧縮の重要性が高まっている．Light Detection and Ranging（LiDAR）に代表される近年の計測技術は，1 秒間に 100 万点を超える大量のデータを取得する．このような大規模なデータの処理において，要求される記憶容量や通信帯域を削減するためには，効率的な点群圧縮が不可欠である．従来の点群圧縮手法の多くは点群のボクセル化を伴う八分木圧縮を利用している [1–5]．八分木圧縮では内部に点を含むボクセルを再帰的に 8 つのボクセルに分割することにより，階層的にボクセルの占有状態を符号化する．この時，点の座標をボクセルの頂点や中心を表す座標に置き換えるとともに，同一座標の点を統合するボクセル化が行われる．これにより，復号点群には原点群に対する座標の誤差や点数の減少による歪が生じる．

八分木圧縮では階層を深めるほどボクセルの解像度が高くなり，原点群に忠実な点群を表現することができる．一方，階層を深めるほど符号量が多くなるというトレードオフがある．このトレードオフに対処するための代表的な方法は，八分木圧縮における階層の深度を制限することである．ユーザが指定した階層で再帰的なボクセルの分割を終了させることにより，符号量と歪のバランスを調整することができる．さらに，符号化および復号の後処理としてボクセル化に起因する点群の歪を補正することも考えられる．復号点群に対して後処理を施すことにより，符号量を増やすことなく，歪を低減することが可能となる．

点群の補正は，形状補完 [6, 7] やアップサンプリング [8, 9]，ノイズ除去 [10, 11] のような様々な目的のために検討されている．これらは 3 次元計測センサによって取得された点群に対して，対象物体の形状への忠実性を改善するために適用される．また，LiDAR センサで取得された低解像度な点群を高解像度化するために，点群超解像手法も検討されている [12, 13]．しかし，これらの手法はセンサなどから得られた未加工の点群を入力とすることを想定しているため，ボクセル化された点群に対して常に有効とは限らない．一方，八分木圧縮の後処理として点群の歪を補正する手法も提案されている [5]．この手法は点の座標を修正するためのオフセットを予測することによって座標の誤差を低減するが，点数の減少による歪には対処できない．

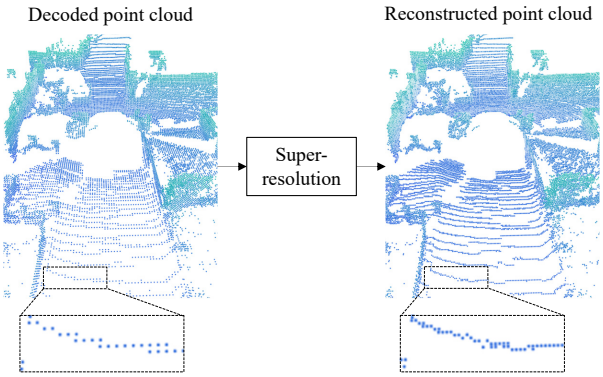


図 1. ボクセル化された点群の超解像の概要．

本稿では八分木圧縮の後処理として歪を補正するために，図 1 に示すようなボクセル化された点群の超解像手法を提案する．提案手法は原点群の圧縮に用いたボクセルよりも高解像度なボクセルの占有確率を予測することにより，ボクセル化された点群を高解像度化する．これにより，符号量を増加させることなく，ボクセル化によって生じた歪を低減する．ただし，LiDAR センサで取得された点群のような大規模な点群の処理は，実行時間やメモリ使用量が大きくなるという課題がある．この課題に対応するために，提案手法はスパース畳み込みに基づく深層ニューラルネットワークを利用する．このネットワークは非零の要素に対してのみ畳み込みを適用するため，大規模な点群に対しても高い効率性を実現する．

本研究の貢献は以下に要約される．

- 高解像度ボクセルに対応する占有確率を予測することにより，ボクセル化された点群の歪を低減する超解像手法を提案する．
- 効率的なスパース畳み込みに基づく超解像のための深層ニューラルネットワークを構築する．
- 評価実験において，提案手法により圧縮の後処理として効率的に歪を低減可能であることを示す．

本稿の以降の構成は次の通りである．第 2 節では点群補正と 3 次元データに対する深層学習についての関連研究を概説する．第 3 節では圧縮の後処理として点群の歪を補正するための超解像手法を提案する．第 4 節では屋外および屋内環境を表す点群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.

2　関連研究

2.1　点群補正

対象物体の形状に対する忠実性を改善するために，様々な点群補正手法が検討されている．形状補完 [6,7] は 3 次元計測センサで計測された不完全な形状を表す点群から，完全な形状を表す点群を生成する．このアプローチは認識や復元の分野における多くのアプリケーションに対して適用可能性がある．しかし，入力点群と大きく異なる形状を復元するため，歪の低減には適さない．点群のアップサンプリング [8,9] は，入力点群からより高密度な点群を生成する．このアプローチは対象物体の表面形状を復元するように均一に点を増加させる．しかし，入力点群における点と点の間を補間する目的で設計されているため，ボクセル化された点群の歪の補正に対して有効とは限らない．ノイズ除去 [10,11] はノイズを含む点群から，ノイズの除去された点群を生成する．このアプローチは滑らかな物体表面上に位置する点群を復元することを目的とするため，LiDAR センサで取得された点群のように，表面情報を利用できない疎な点群に対して適用することが困難である．

点群の解像度を向上させる超解像手法も検討されている．LiDAR センサで取得された点群の超解像 [12,13] は低解像度の点群から高解像の点群を生成する．このアプローチは入力点群を距離画像に投影することにより，点群の超解像を距離画像の超解像に帰着する．高解像度化された距離画像は 3 次元空間に逆投影され，高解像度点群として出力される．このアプローチは未加工の点群を入力とするため，ボクセル化された点群に対しても有効とは限らない．ボクセル超解像 [14] は低解像度のボリュームから高解像度のボリュームを復元する．ただし，このアプローチは 3 次元形状の陰関数表現に基づいており，モデルの学習には点が物体の内部に位置するか否かを判定するために隙間の無いメッシュデータが必要となる．そのため，適用可能な点群が限定的である．

符号化で生じた歪を補正するために，ボクセル化された点群の座標を修正する手法が提案されている [5]．この手法は coordinate refinement module (CRM) と呼ばれる深層ニューラルネットワークを用いて，点ごとに原点群への誤差を補正するためのオフセットを予測する．しかし，CRM は点ごとに単一のオフセットしか予測しないため，ボクセル化の際にボクセル内部に複数の点が存在する場合に対処できない．すなわち，CRM はボクセル化で失われた情報を復元するようには設計されていない．それに対し，本稿ではボクセルの内部に構築された高解像度ボリュームの占有確率を予測することにより，ボクセル化に起因する歪を低減する手法を提案する．

2.2　3 次元データに対する深層学習

3 次元データに対して深層学習を適用するために，様々な深層ニューラルネットワークが検討されている．従来のアプローチはボクセルベース [15,16]，投影ベース [17–20], 点ベース [21–24] に大別される．

ボクセルベースのアプローチは 3 次元モデルを値を持つボクセルの集合として表されるボリュームに変換し，3 次元の畳み込みニューラルネットワークを用いてボリュームを処理する．ボリュームを構築する際には任意の解像度のボクセルを使用することができるが，解像度の向上に伴って計算量やメモリ使用量が急速に増大するという課題がある．八分木表現に基づくボクセルベースのアプローチの拡張 [25,26] はこの課題に対処するが，依然として利用可能な解像度は制限されている．

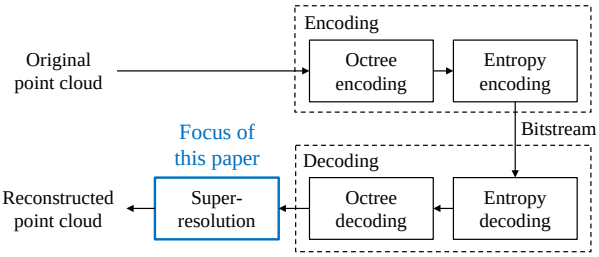


図 2. 点群圧縮フレームワークの概要.

投影ベースのアプローチでは 3 次元モデルを 2 次元平面に投影することにより，多視点画像やシルエット画像，距離画像のような 2 次元データに変換する．そして，2 次元の畳み込みニューラルネットワークを用いてそれらのデータを処理する．これにより，大規模な点群に対しても効率的な処理が可能となる．LiDAR センサで取得された点群を処理する場合には，2 次元格子に対して規則的に 3 次元点を投影できるために，距離画像が利用されることが多い [19,20]．距離画像における物体の境界では，隣り合うピクセル同士が 3 次元空間では遠く離れている可能性がある．そのため，距離画像に対して畳み込みを実行すると，物体の境界においてブラーが発生する恐れがある [27].

点ベースのアプローチはセンサなどから取得された点群を直接処理する．また，このアプローチはメッシュデータやデプスマップのような他の形式で表現された 3 次元データから構築された点群を処理することも可能である [28,29]．先駆的な手法である PointNet [21] は点群全体から大域特徴を抽出するため，局所的な構造を捉えることができない．PointNet を発展させた手法は点群の固定半径近傍探索や k 近傍探索によってグループ化された点の集合から局所特徴を抽出する [22–24]．しかし，これらの手法は近傍探索に起因して，大規模な点群に対しては計算量が大きくなるという課題がある．

3 次元構造を維持しつつ大規模な点群を効率的に処理するために，スパース畳み込みに基づく深層ニューラルネットワークを用いて 3 次元点群を処理する手法が提案されている [30–32]．このアプローチはボクセルベースのアプローチに類似しているが，通常の畳み込みが全てのボクセルを用いて畳み込み操作を行うのに対して，スパース畳み込みはデータを持つボクセルのみを用いるという特徴がある．そのため，特にスパース性の高いデータに対して，計算量やメモリ使用量の増加を抑制する効果が得られる．これらの手法に着想を得て，本稿では点群の超解像のためにスパース畳み込みに基づく深層ニューラルネットワークを構築する．

3　提案手法

本節では，点群圧縮の後処理としての点群超解像手法を提案する．図 2 に本稿で想定する圧縮フレームワークを示す．このフレームワークは多くの点群圧縮手法

で採用される八分木圧縮とエントロピー圧縮を利用する [33]．はじめに，符号化ブロックにおいて原点群に対して八分木符号化を実行する．八分木符号化は点の座標情報の代わりに，ボクセルの内部に点が存在するか否かを表す符号を保存する．すなわち，この処理において原点群はボクセル化される．これらの符号はエントロピー符号化によってビットストリームに符号化される．復号ブロックではエントロピー復号および八分木復号によってビットストリームから点群を復号する．点群の座標系は八分木復号を通じて復元される．そして，復号ブロックから復号点群を出力する．八分木符号化の際にボクセル化されているため，復号点群はボクセル化された点群である．ボクセル化によって情報が失われているため，原点群と復号点群の間には歪が存在する．この歪を低減するために，図 1 に示すような復号点群の超解像を導入する．ここでは，原点群をボクセル化する際に用いたボクセルより高解像度なボクセルの占有確率を予測することにより，高解像度な点群を復元する．本稿では，この超解像に焦点を当てる．最後に，このフレームワークは復元点群を出力する．

3.1 ボクセル化

はじめに，点群のボクセル化について説明する．ここでは八分木表現にしたがって，階層的にボクセルを 8 つに分割することによるボクセル化を想定する．最も深い階層の番号を L とする場合，l 番目 $(l = 1, 2, \ldots, L)$ の階層におけるボクセルサイズは $\delta^{(l)} = 2^{L-l}$ と表すことができる．ボクセル化においては，点群の座標はボクセルの頂点を表す座標に変換される．これは $\delta^{(l)}$ を量子化間隔とする量子化と等しい．原点群を $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^{N}$ とする．l 番目の階層における量子化は次式で表される．

$$\boldsymbol{q}_i^{(l)} = \lfloor \frac{s\boldsymbol{p}_i + \boldsymbol{t}}{\delta^{(l)}} \rfloor, \tag{1}$$

ここで $\boldsymbol{q}_i^{(l)} \in \mathbb{R}^3$ は量子化された点，$\boldsymbol{t} \in \mathbb{R}^3$ は全ての座標を非負にするためのオフセット，s はスケール要素である．八分木表現ではボクセルの内部に点が存在するか否かを表す符号のみが保存される．したがって，同一のボクセル内に存在する全ての点は単一の点に統合される．ボクセルごとに点の個数を表すサイド情報を保存する拡張も可能であるが，それは符号量を増加させるため，本稿では考慮しない．元々の座標系における点は次式によって復元される．

$$\boldsymbol{p}_i^{(l)} = \frac{\delta^{(l)}\boldsymbol{q}_i^{(l)} - \boldsymbol{t}}{s}, \tag{2}$$

ここで $\boldsymbol{p}_i^{(l)} \in \mathbb{R}^3$ は l 番目の階層の量子化間隔を用いて復元された点である．

3.2 高解像度点群の復元

次に，$\boldsymbol{p}_i^{(l)}$ から $\boldsymbol{p}_i^{(l+1)}$ を復元することを検討する．八分木表現においては，階層を増加させることはボクセルを再帰的に分割することに相当する．l 番目の階層におけるボクセルサイズは $\delta^{(l)}$ であるため，l が大きいほどボクセルの解像度が高くなる．したがって，l 番目の階層のボクセルは，$m = 2^3$ 個の $(l+1)$ 番目の階層のボクセルで構成されるボリュームとみなせる．$\boldsymbol{y}_i^{(l+1)} \in \{0, 1\}^m$ を $\boldsymbol{p}_i^{(l)}$ に対応する，$(l+1)$ 番目の階層のボクセルの占有確率とする．この時，$\boldsymbol{y}_i^{(l+1)}$ の j 番目 $(j = 1, 2, \ldots, m)$ の要素が占有である場合に，次式にしたがって点を生成することにより，$\boldsymbol{p}_i^{(l+1)}$ が復元される．

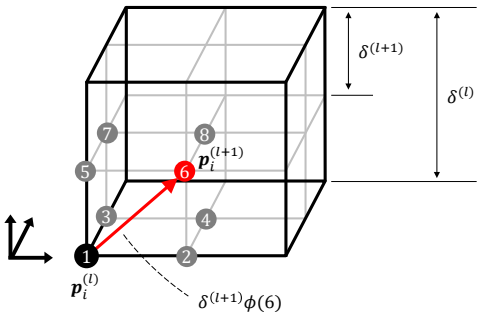


図 3. 点群のボクセル化に用いるボクセルとその高解像度ボクセルの関係．

$$\boldsymbol{p}_i^{(l+1)} = \boldsymbol{p}_i^{(l)} + \delta^{(l+1)}\phi(j), \tag{3}$$

ここで $\phi(\cdot)$ は $\boldsymbol{y}_i^{(l+1)}$ のインデックスから占有されたボクセルの相対位置を表すベクトルへの変換関数である．$\boldsymbol{y}_i^{(l+1)}$ には複数の占有された要素が含まれる場合もあるため，$\boldsymbol{q}_i^{(l)}$ から複数の $\boldsymbol{q}_i^{(l+1)}$ が復元される可能性がある．

図 3 にボクセル化に用いるボクセルとその高解像度ボクセルの模式図を示す．黒い立方体はサイズが $\delta^{(l)}$ のボクセルを表す．l 番目の階層では，このボクセルの内部の点は $\boldsymbol{p}_i^{(l)}$ として量子化される．灰色の立方体は，黒いボクセルを 8 つに分割して得られる，サイズが $\delta^{(l+1)}$ のボクセルである．円で表されるこれらのボクセルの頂点が，$\boldsymbol{p}_i^{(l+1)}$ が復元される位置の候補である．円の中の数字は $\boldsymbol{y}_i^{(l+1)}$ のインデックス，すなわち j に対応する．ここで $j = 6$ に対応するボクセルが占有であるとする．$\boldsymbol{p}_i^{(l+1)}$ が赤い円として復元される時，相対位置を表すベクトル $\delta^{(l+1)}\phi(6)$ は赤い矢印として表される．$\boldsymbol{p}_i^{(l+1)}$ は灰色の円または $\boldsymbol{p}_i^{(l)}$ の位置に復元される可能性もある．

提案手法は八分木圧縮において階層の深度を l に制限した場合の復号点群から，各 $\boldsymbol{p}_i^{(l)}$ に対する占有確率を予測する．ground-truth が $\boldsymbol{y}_i^{(l+1)}$ である，予測された占有確率を $\boldsymbol{x}_i^{(l+1)} \in \mathbb{R}^m$ とする．そして，$\boldsymbol{x}_i^{(l+1)}$ を用いて，$(l+1)$ 番目の階層のボクセルの頂点に対応する，高解像度化された点 $\hat{\boldsymbol{p}}_i^{(l+1)}$ を得る．

3.3 ネットワーク構造

提案手法は深層ニューラルネットワークを用いて復号点群から点ごとの占有確率を予測する．ただし，LiDAR センサで取得された点群のような大規模な点群の処理は，実行時間やメモリ使用量が大きくなるという課題がある．この課題に対応するために，スパース畳み込みのフレームワーク [31] を用いてネットワークを構築する．図 4 に提案手法のネットワーク構造を示す．ネットワークには l 番目の階層でボクセル化された点群 $\mathcal{P}^{(l)} = \{\boldsymbol{p}_i^{(l)} \in \mathbb{R}^3\}_{i=1}^{n}$ から構築されたスパーステンソルが入力される．スパーステンソルは，点ごとの座標と特徴を格納するためのデータ構造である．このネットワークは，はじめにスパース畳み込み層，スパース逆畳み込み層，スキップ接続で構成される U 字型のモジュールから点ごとの特徴を抽出する．畳み込みは，Conv ブ

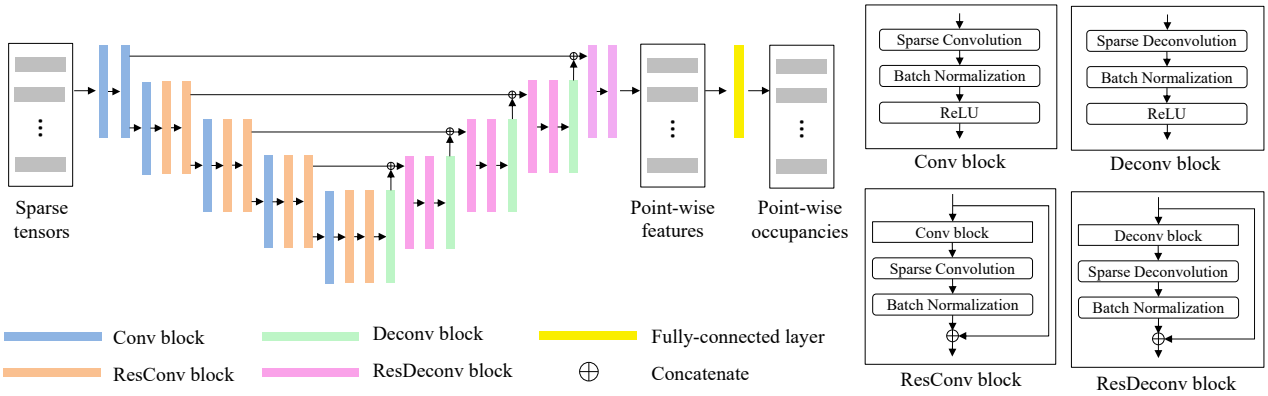


図 4. 提案手法のネットワーク構造.

ロックと ResConv ブロックを用いて実行される．Conv ブロックはスパース畳み込み層，バッチ正規化, ReLU で構成された畳み込みブロックである．ResConv ブロックは残差接続による特徴の結合を導入した畳み込みブロックである．畳み込みでは，ストライド 2 の 4 つの Conv ブロックによりスパーステンソルがダウンサンプリングされる．スパース畳み込みは非零の要素に対してのみ畳み込みを適用するため，通常の畳み込みに比べて高い効率性を実現する．逆畳み込みは，スキップ接続によって層ごとに特徴を結合しながら，畳み込みと逆の様式で実行される．逆畳み込みの際にスパーステンソルはアップサンプリングされ，ボクセル化された点群の座標が復元される．これにより，点群の各点に対応する点ごとの特徴の集合が抽出される．そして，このネットワークは全結合層に基づく予測モジュールによって点ごとに占有確率 $\boldsymbol{x}_i^{(l+1)}$ を予測する．

ネットワークから出力される占有確率は，$[0,1]$ の範囲の実数で表現される．推論時には，この占有確率から 0（非占有），1（占有）の二値で表される占有状態を推定するために，閾値判定を導入する．これにより，m 通りの占有確率から最大で m 個の点が復元される．復元される点は復号点群よりも小さなボクセルによって座標が表現されるため，歪を補正することができる．ここで，m 通りの占有確率の全てが閾値を下回る場合，全てのボクセルが非占有であると推定される．八分木表現では現在の階層のボクセルが占有である場合，それを分割して得られるボクセルの中で少なくとも 1 つは占有となることが保証される．そのため，全てのボクセルが非占有になる場合，八分木符号化の表現と整合性がとれなくなる．この問題に対処するために，次式で表される動的閾値を使用する．

$$\beta = \begin{cases} \max\limits_j(x_{i,j}^{(l+1)}) & \text{if } \max\limits_j(x_{i,j}^{(l+1)}) < \alpha \\ \alpha & \text{otherwise} \end{cases}, \tag{4}$$

ここで α は固定閾値，β は動的閾値である．β 以上の占有確率を占有と判定することにより，各点に対応する m 通りの高解像度ボクセルの内，少なくとも 1 つが占有となることが保証される．

3.4 損失関数

ネットワークの損失関数として，予測された占有確率と ground-truth の間の重み付き二値交差エントロピーを用いる．学習を安定させるために，予測値に対してシグモイド関数 $\sigma(\cdot)$ を適用する．この損失関数は次式で表される．

$$\mathcal{L} = -\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\Big\{\lambda y_{i,j}^{(l+1)}\log\Big(\sigma\big(x_{i,j}^{(l+1)}\big)\Big) + \big(1-y_{i,j}^{(l+1)}\big)\log\Big(1-\sigma\big(x_{i,j}^{(l+1)}\big)\Big)\Big\}, \tag{5}$$

ここで λ は各項のバランスを調整するための重み係数，n は入力点の個数，m は予測する占有確率の個数を表す．

4 評価実験

本節では，提案手法が符号量を増加させることなくボクセル化に起因する点群の歪を低減可能であることを実験的に示す．実験のために，八分木圧縮に基づく代表的な点群圧縮方式である geometry-based point cloud compression [3, 34] を用いて原点群を圧縮する．以下では，この圧縮方式を Anchor と表記する．本節の実験においては，Anchor によって符号化および復号された点群を後処理への入力とする．

4.1 実験設定

屋外および屋内環境を表す点群を用いて提案手法を評価するために，SemanticKITTI [35] および ScanNet [36] データセットを使用する．SemanticKITTI データセットは都市部や高速道路等の屋外環境において車両に搭載された LiDAR センサで取得された大規模な点群データセットであり，22 通りのシーケンスから収集された 43552 個の点群で構成される．モデルの学習にはシーケンス 00 から 10 の中の 08 以外を，検証にはシーケンス 08 を，評価にはシーケンス 11 から 21 を使用する．ScanNet データセットは事務室やアパート等の屋内環境において RGB-D センサで取得された大規模な 3 次元メッシュデータである．点群を構築するためにポアソンディスクサンプリング法 [37] を用いて各メッシュデータから 10 万点をサンプリングする．データセットに定義されたリストに従って，モデルの学習，検証，評価に対してそれぞれ個別のデータを使用する．

ビットレートの評価指標として，復号点群の 1 点あたりの符号量を表す bit per point (bpp) を用いる．復元精度の指標としては，Chamfer distance (CD) [38] および point-to-point PSNR [39] を使用する．2 つの点群 $\mathcal{P}$ およ

表 1. ボクセル化の設定.

Level of octree	8	9	10	11	12	13
Voxel size [mm]	1024	512	256	128	64	32

び $\hat{\mathcal{P}}$ の間の CD は，次式で定義される．

$$\mathrm{CD}(\mathcal{P},\hat{\mathcal{P}}) = \frac{1}{|\mathcal{P}|}\sum_{\boldsymbol{p}\in\mathcal{P}}\min_{\hat{\boldsymbol{p}}\in\hat{\mathcal{P}}}\|\boldsymbol{p}-\hat{\boldsymbol{p}}\|_2 + \frac{1}{|\hat{\mathcal{P}}|}\sum_{\hat{\boldsymbol{p}}\in\hat{\mathcal{P}}}\min_{\boldsymbol{p}\in\mathcal{P}}\|\boldsymbol{p}-\hat{\boldsymbol{p}}\|_2. \tag{6}$$

PSNR は 2 つの点群の間で双方向に測定し，各単一方向で測定された PSNR の最小値として表す．Anchor と比べた複数の bpp にわたる復元精度を評価するために，Bjøntegaard delta bit-rate (BD-BR) および Bjøntegaard delta PSNR (BD-PSNR) [40] を使用する．BD-BR は Anchor と同等の PSNR を達成する際の bpp の平均削減量を，BD-PSNR は Anchor と同等の bpp を達成する際の PSNR の平均改善量を表す．また，ボクセル表現における占有確率の予測精度を評価するために，intersection over union (IoU) を使用する．l 番目の階層のボクセルサイズを用いてボクセル化された点群を $\mathcal{P}^{(l)}$ とする．後処理によって $\mathcal{P}^{(l)}$ から $\hat{\mathcal{P}}^{(l+1)}$ を復元する場合，$(l+1)$ 番目の階層のボクセルサイズを用いて $\hat{\mathcal{P}}^{(l+1)}$ からボクセル $\hat{\mathcal{V}}^{(l+1)}$ を構築する．この時，$\hat{\mathcal{V}}^{(l+1)}$ とその ground-truth $\mathcal{V}^{(l+1)}$ の IoU は次式で表される．

$$\mathrm{IoU}(\mathcal{V}^{(l+1)},\hat{\mathcal{V}}^{(l+1)}) = \frac{\Sigma_i \mathbb{1}[\mathcal{V}^{(l+1)}(i)\hat{\mathcal{V}}^{(l+1)}(i) > 0]}{\Sigma_i \mathbb{1}[\mathcal{V}^{(l+1)}(i) + \hat{\mathcal{V}}^{(l+1)}(i) > 0]}, \tag{7}$$

ここで i はボクセルのインデックス，$\mathbb{1}$ は指示関数である．

座標をミリメートル単位で表すために，式 (1) および (2) におけるスケール要素を $s = 1000$ に設定する．ボクセル空間を原点群を包含する大きさにするため，八分木の最も深い階層の番号を $L = 18$ に設定する．そして，八分木圧縮において階層の深度を表 1 の 1 行目に示す 6 通りに制限した場合の復号点群を構築する．この表における 2 行目は各階層に対応するボクセルサイズを示している．各階層においてそれよりも一つ深い階層の点群から占有確率を表すビット列を構築し，それらを現階層の点群の ground-truth ラベルとする．訓練用データにおけるこれらのラベル内の 0 と 1 の個数の比を式 (5) における係数 λ として使用する．提案手法は PyTorch ライブラリを用いて実装する．モデルは学習率 10^{-3} の Adam 最適化器を用いて 100 エポック訓練する．モデルの訓練は，異なる階層で構築された点群に対して独立に実行する．式 (4) における固定閾値は $\alpha = 0.5$ に設定する．モデルの学習や検証，評価には，Nvidia Quadro GV100 GPU を用いる．

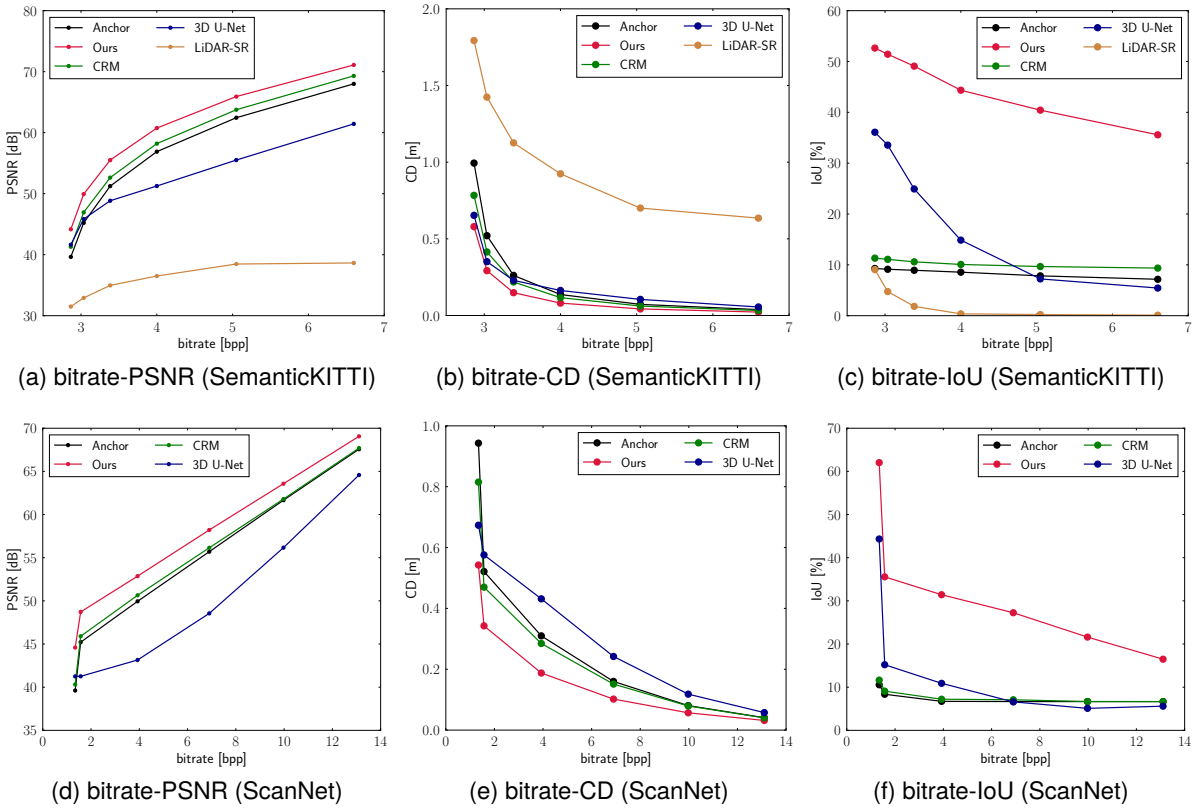
4.2 従来手法との比較

提案手法を最先端の歪補正手法である CRM [5] および点群超解像手法である LiDAR-SR [12] と比較する．また，ボリュームの復元に用いられる代表的なネットワークである 3D U-Net [15] との比較も行う．CRM に対しては，9^3 個のボクセルを用いてローカルボクセル表現を構築する．この手法では，量子化された点のボクセルに含まれる原点群内の点へのオフセットを ground-truth とする．ボクセル内に複数の点が存在する場合には，それらの中心へ向かうオフセットを ground-truth とする．3D U-Net においてはメモリ使用量を抑えるために点群を分割し，個別に超解像を実行する．ここでは，8^3 個のボクセルで構成されるボリュームを 1 単位として点群を分割する．本実験では，提案手法と同様に動的な閾値を用いてボクセルの占有状態を推定する．LiDAR-SR は低解像度な点群を投影した距離画像から，高解像度な点群を投影した距離画像を復元するように設定する．ネットワークへの入力および出力となる距離画像の解像度は 64×1024 ピクセルとする．LiDAR-SR は LiDAR センサで取得された点群に対して設計されているため，SemanticKITTI データセットにおいてのみ評価する．

4.2.1 定量評価

図 5 に各ビットレートにおける PSNR，CD，および IoU を示す．ここでは，ボクセルサイズが小さくなるほどビットレートが大きくなる．(a)–(c) と (d)–(f) はそれぞれ SemanticKITTI と ScanNet における結果を表す．(c) および (f) において，Anchor の IoU は予測ボクセルの代わりに ground-truth ボクセル $\mathcal{V}^{(l)}$ と $\mathcal{V}^{(l+1)}$ の間で測定した．これは，ボクセルサイズの低下に伴って IoU がどの程度低下するかを示すための参考値である．この図より，提案手法は全ての指標において従来手法よりも優れた結果を達成することが分かる．また，提案手法は全てのビットレートにおいて Anchor を基準として歪を低減し，IoU を改善する．これは，後処理によってボクセル化に起因する歪が低減されることを示している．SemanticKITTI に比べて，ScanNet ではボクセルサイズが小さくなるほど IoU が大きく低下する．これは ScanNet では空間的なサンプリングによって点群を構築したことに起因する．それらは LiDAR で取得した点群に比べて不規則性が高いために，高解像度ボクセルの占有確率の予測がより困難となる．ただし，提案手法ではそのような点群に対しても歪を低減することができる．CRM もまた，全てのビットレートにおいて Anchor を基準として歪を低減する．しかし，CRM は点群の座標を修正するが，高解像度な点群を復元するわけではないため，全ての評価指標において改善量が少ない．3D U-Net はボクセルサイズが大きい場合には Anchor を基準として歪を低減するが，ボクセルサイズが小さくなると歪が増加する．これは，3D U-Net が多数のボクセルで構成されるボリューム単位でデータを復元するためである．この手法では，低解像度のボクセルが占有であっても，それに対応する高解像度ボクセルが全て非占有となる可能性がある．同様に，低解像度のボクセルが非占有であっても，それに対応する高解像度ボクセルのいずれかが占有となる可能性もある．また，ボクセルサイズが小さいほどデータがスパースになるため，復元が困難となる．そのため，ビットレートが大きくなるほど，Anchor を基準として相対的に歪が大きくなる．LiDAR-SR は Anchor を基準として歪を大きく増加させる．この手法は点群を投影した距離画像を復元するため，復元される距離画像には復元誤差が含まれる．この復元誤差がボクセル化に起因する歪よりも大きな歪を生じさせるため，Anchor よりも歪が大きくなると考えられる．

表 2 に各手法の BD-BR および BD-PSNR を示す．提案手法は Anchor を基準としてビットレートを平均 11.18–22.70%削減し，PSNR を平均 2.77–3.78 dB 改善す



(a) bitrate-PSNR (SemanticKITTI) (b) bitrate-CD (SemanticKITTI) (c) bitrate-IoU (SemanticKITTI)

(d) bitrate-PSNR (ScanNet) (e) bitrate-CD (ScanNet) (f) bitrate-IoU (ScanNet)

図 5. 各ビットレートにおける PSNR, Chamfer distance (CD)，および intersection over union (IoU) の比較.

表 2. BD-BR [%] および BD-PSNR [dB] の比較.

Method	SemanticKITTI		ScanNet	
	BD-BR	BD-PSNR	BD-BR	BD-PSNR
Ours	**-11.18**%	**3.78** dB	**-22.70**%	**2.77** dB
CRM	-4.00%	1.37 dB	-4.47%	0.56 dB
3D U-Net	16.73%	-4.90 dB	64.74%	-5.77 dB
LiDAR-SR	141.94%	-20.92 dB	-	-

表 3. 実行時間 [sec] の比較.

(a) SemanticKITTI

Level of octree	8	9	10	11	12	13	mean
Ours	0.025	**0.028**	**0.036**	**0.051**	**0.078**	**0.111**	**0.055**
CRM	**0.022**	0.039	0.095	0.205	0.366	0.533	0.210
3D U-Net	0.036	0.070	0.198	0.481	1.138	2.080	0.667
LiDAR-SR	0.361	0.362	0.364	0.380	0.387	0.400	0.376

(b) ScanNet

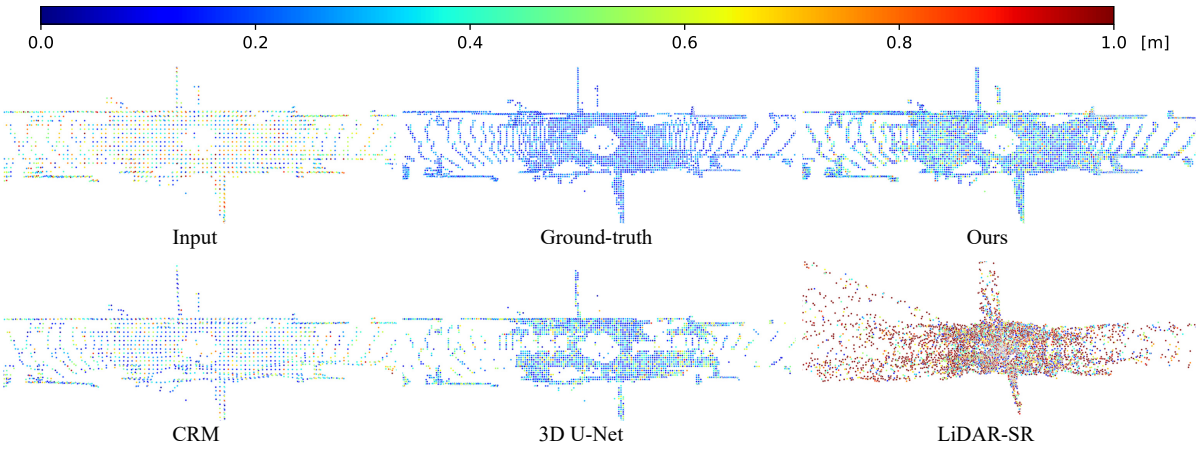
Level of octree	8	9	10	11	12	13	mean
Ours	**0.039**	**0.062**	**0.082**	**0.099**	**0.117**	**0.119**	**0.087**
CRM	0.088	0.282	0.408	0.429	0.447	0.448	0.350
3D U-Net	0.065	0.242	0.883	2.891	4.568	4.632	2.213

る．CRM ではそれらの値はそれぞれ 4.00–4.47%および 0.56–1.37 dB である．3D U-Net および LiDAR-SR はどちらの指標においても歪を増加させることが確認できる．

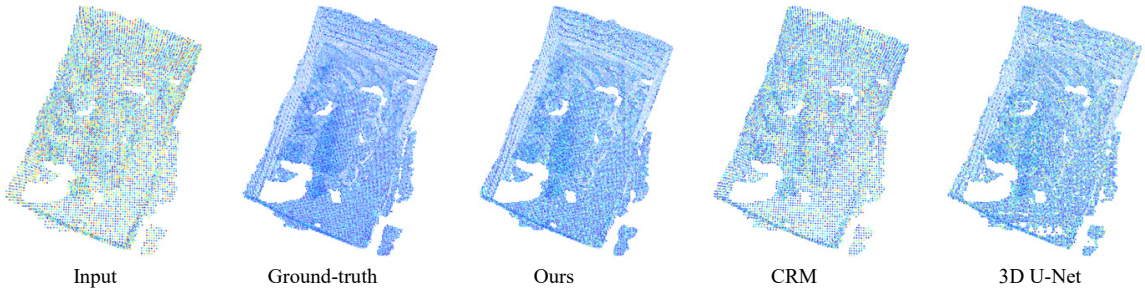
表 3 に八分木圧縮において制限する階層の深度に対応する復号点群を用いた後処理の実行時間 [sec] を示す．本稿では後処理にのみ焦点を当てるため，符号化および復号の実行時間は含めない．提案手法は他の手法と比べて最も平均実行時間が短い．これは，提案手法がスパース畳み込みに基づいて効率的に処理可能であるためである．階層が深くなるにつれてボクセルサイズが小さくなり，処理すべき点の個数が多くなるため，実行時間が増加する傾向が見られる．ただし，LiDAR-SR は全ての点を距離画像に投影するため，実行時間が階層の深度にほとんど依存しない．CRM および 3D U-Net では通常の畳み込みに基づくため，提案手法に比べて実行時間が大きく増加する傾向が見られる．3D U-Net の方が層数が多いため，その傾向が特に顕著である．また，ScanNet では鉛直方向へ点が広く分布するため，SemanticKITTI に比べて実行時間が大きくなる．

4.2.2 定性評価

図 6 に各手法による後処理を実行した後の点群を示す．後処理への入力および ground-truth は八分木圧縮において階層の深度をそれぞれ $l = 8$ および $l = 9$ に制限した場合の復号点群である．各点は，原点群内の最近傍点への距離で表される誤差に応じて色付けされる．入力ではボクセルサイズの大きさに起因して，全体にわたって大きな誤差が生じている．一方，ground-truth ではボクセルサイズが小さくなることで誤差が低減されている．提案手法は最も ground-truth に近い点群を復元することがわかる．CRM は点ごとに誤差を補正するが，点群の



(a) SemanticKITTI



(b) ScanNet

図 6. 原点群に対する復元誤差の可視化．カラーバーは点の色と誤差の対応関係を表す．Ground-truth は入力点群よりも高解像度なボクセルを用いてボクセル化された点群である．

解像度が低いままである．3D U-Net は点が疎な領域で形状の復元精度が低くなり，ほとんど点のない領域を生み出すことがある．LiDAR-SR は距離画像の復元誤差に起因して，入力点群よりも大きな歪を発生させる．

4.3 Ablation Study

提案手法における動的閾値 (Dynamic Threshold; DT) および式 (5) の損失関数における重み (Weight; WT) の効果を検証する．表 4 に，提案手法においてこれらを使用しない場合の BD-BR および BD-PSNR を示す．この表では ✓ および ✗ がそれぞれ使用と不使用を表す．DT を使用しない場合には，常に固定閾値が使用される．WT を使用しない場合には，重み係数が $\lambda = 1$ に設定される．DT も WT も使用しない場合，性能が最も低下することがわかる．点群は空間的にスパースなデータであるため，点群から構築されるボクセルの占有状態は占有よりも非占有の方が多くなる．このようなデータの不均衡に起因して，非占有と予測されるボクセルが多数発生し，歪が大きくなると考えられる．WT を使用する場合，損失関数において不均衡に対処することにより，性能を改善できる．一方，WT を使用しない場合であっても，DT を使用する場合には性能を改善できることがわかる．高解像度なボクセルに対する全ての占有確率の予測値が閾値を下回る場合，点の欠落した領域が発生する．DT にはそのような領域の発生を防ぐことにより，歪を抑制する効果がある．そして，DT と WT の両方を使用する場合に，最も優れた性能を達成することがわかる．

表 4. 動的閾値 (Dynamic Threshold; DT) および損失関数における重み (Weight; WT) の効果．

Setting		SemanticKITTI		ScanNet	
DT	WT	BD-BR	BD-PSNR	BD-BR	BD-PSNR
✓	✓	**-11.18**%	**3.78** dB	**-22.70**%	**2.77** dB
✓	✗	-10.80%	3.62 dB	-21.80%	2.65 dB
✗	✓	9.87%	-8.49 dB	72.90%	-1.29 dB
✗	✗	33.36%	-25.51 dB	143.74%	-18.45 dB

5 まとめ

本稿では，点群圧縮フレームワークにおいてボクセル化に起因する歪を補正するための超解像手法を提案した．提案手法は符号化および復号の後処理としてボクセル化された点群を高解像度化することにより，符号量を増加させることなく歪を低減する．LiDAR センサで取得されたような大規模な点群に対して効率的な処理を実現するために，スパース畳み込みに基づく深層ニューラルネットワークを用いて点ごとに高解像度ボクセルの占有確率を予測した．提案手法はこのネットワークの出力から，八分木表現と整合性を持つように動的な閾値を用いて二値の占有状態を推定する．SemanticKITTI および ScanNet データセット上での実験では，提案手法の有効性と効率性が示された．今後は，より高解像度なボクセルの利用による歪補正性能の改善や，ボクセルに依存しない超解像手法について検討する．

参考文献

[1] D. Meagher, “Geometric modeling using octree encoding,” CGIP, vol.19, no.2, pp.129–147, 1982.

[2] D.C. Garcia, T.A. Fonseca, R.U. Ferreira, and R.L. de Queiroz, “Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts,” IEEE TIP, vol.29, pp.313–322, 2019.

[3] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC),” APSIPA TSIP, vol.9, 2020.

[4] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, “Octsqueeze: Octree-structured entropy model for lidar compression,” Proc. of CVPR, pp.1313–1323, 2020.

[5] Z. Que, G. Lu, and D. Xu, “VoxelContext-Net: An octree based framework for point cloud compression,” Proc. of CVPR, pp.6042–6051, 2021.

[6] D. Stutz and A. Geiger, “Learning 3D shape completion from laser scan data with weak supervision,” Proc. of CVPR, pp.1955–1964, 2018.

[7] X. Wang, M.H. Ang, and G.H. Lee, “Point cloud completion by learning shape priors,” Proc. of IROS, pp.10719–10726, IEEE, 2020.

[8] 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.

[9] R. Li, X. Li, C.W. Fu, D. Cohen-Or, and P.A. Heng, “PU-GAN: A point cloud upsampling adversarial network,” Proc. of ICCV, pp.7203–7212, 2019.

[10] S. Luo and W. Hu, “Score-based point cloud denoising,” Proc. of ICCV, pp.4583–4592, 2021.

[11] P. Hermosilla, T. Ritschel, and T. Ropinski, “Total denoising: Unsupervised learning of 3D point cloud cleaning,” Proc. of ICCV, pp.52–60, 2019.

[12] T. Shan, J. Wang, F. Chen, P. Szenher, and B. Englot, “Simulation-based LiDAR super-resolution for ground vehicles,” RAS, vol.134, p.103647, 2020.

[13] J. Yue, W. Wen, J. Han, and L.T. Hsu, “3D point clouds data super resolution-aided LiDAR odometry for vehicular positioning in urban canyons,” IEEE TVT, vol.70, no.5, pp.4098–4112, 2021.

[14] 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.

[15] Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense volumetric segmentation from sparse annotation,” Proc. of MICCAI, pp.424–432, Springer, 2016.

[16] C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L.J. Guibas, “Volumetric and multi-view CNNs for object classification on 3D data,” Proc. of CVPR, pp.5648–5656, 2016.

[17] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, “Pix2Vox: Context-aware 3D reconstruction from single and multi-view images,” Proc. of ICCV, pp.2690–2698, 2019.

[18] S. Tulsiani, A.A. Efros, and J. Malik, “Multi-view consistency as supervisory signal for learning shape and pose prediction,” Proc. of CVPR, pp.2897–2905, 2018.

[19] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, “SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation,” ECCV, pp.1–19, Springer, 2020.

[20] Z. Liang, Z. Zhang, M. Zhang, X. Zhao, and S. Pu, “RangeIoUDet: Range image based real-time 3D object detector optimized by intersection over union,” Proc. of CVPR, pp.7140–7149, 2021.

[21] C.R. Qi, H. Su, K. Mo, and L.J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” Proc. of CVPR, pp.652–660, 2017.

[22] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” NeurIPS, vol.30, 2017.

[23] H. Thomas, C.R. Qi, J.E. Deschaud, B. Marcotegui, F. Goulette, and L.J. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” Proc. of ICCV, pp.6411–6420, 2019.

[24] W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep convolutional networks on 3D point clouds,” Proc. of CVPR, pp.9621–9630, 2019.

[25] G. Riegler, A. Osman Ulusoy, and A. Geiger, “OctNet: Learning deep 3D representations at high resolutions,” Proc. of CVPR, pp.3577–3586, 2017.

[26] P.S. Wang, Y. Liu, Y.X. Guo, C.Y. Sun, and X. Tong, “O-CNN: Octree-based convolutional neural networks for 3D shape analysis,” ACM TOG, vol.36, no.4, pp.1–11, 2017.

[27] Y. Wang, W.L. Chao, D. Garg, B. Hariharan, M. Campbell, and K.Q. Weinberger, “Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving,” Proc. of CVPR, pp.8445–8453, 2019.

[28] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning representations and generative models for 3D point clouds,” Proc. of ICML, pp.40–49, PMLR, 2018.

[29] C.R. Qi, W. Liu, C. Wu, H. Su, and L.J. Guibas, “Frustum PointNets for 3D object detection from RGB-D data,” Proc. of CVPR, pp.918–927, 2018.

[30] B. Graham, M. Engelcke, and L. Van Der Maaten, “3D semantic segmentation with submanifold sparse convolutional networks,” Proc. of CVPR, pp.9224–9232, 2018.

[31] C. Choy, J. Gwak, and S. Savarese, “4D spatio-temporal ConvNets: Minkowski convolutional neural networks,” Proc. of CVPR, pp.3075–3084, 2019.

[32] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, “Searching efficient 3D architectures with sparse point-voxel convolution,” ECCV, pp.685–702, Springer, 2020.

[33] C. Cao, M. Preda, and T. Zaharia, “3D point cloud compression: A survey,” Proc. of Web3D, pp.1–9, 2019.

[34] MPEG Group, “Geometry based point cloud compression (G-PCC) test model.” `https://github.com/MPEGGroup/mpeg-pcc-tmc13` [2022 Accessed].

[35] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” Proc. of ICCV, pp.9297–9307, 2019.

[36] 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.

[37] G. Ranzuglia, M. Callieri, M. Dellepiane, P. Cignoni, and R. Scopigno, “Efficient and flexible sampling with blue noise properties of triangular meshes,” TVCG, vol.18, no.6, pp.914–924, 2012.

[38] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liua, and D. Lin, “Density-aware chamfer distance as a comprehensive metric for point cloud completion,” NeurIPS, 2021.

[39] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, “Geometric distortion metrics for point cloud compression,” Proc. of ICIP, pp.3460–3464, IEEE, 2017.

[40] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, 2001.