

時間的相関を用いた疎な点群の圧縮のための点拡張

Point Augmentation for Compression of Sparse Point Cloud using Temporal Correlation

松崎 康平[†]
Kohei Matsuzaki

河村 圭[†]
Kei Kawamura

概要

点群圧縮は、没入型テレプレゼンスや自律走行車のような 3 次元センシング応用技術を実用化するための必須技術である。動き補償と占有予測に基づく従来のフレーム間幾何符号化手法は、フレーム内符号化手法に比べて優れた圧縮性能を達成する。しかし、この手法は正確な動き補償の後であっても、現フレームの点群と参照フレームの点群の間の幾何的誤差によって圧縮性能が制限される。本稿では、この幾何的誤差に対処するために、参照フレームにおける点の拡張手法を提案する。提案手法は測域センサの光線方向によって点を分類し、その分類ごとに構築されたオフセット集合を用いて点を生成する。オフセット集合は、収集されたオフセット候補集合から訓練用の点群データの平均符号量を削減するものを選択することによって構築する。動的に取得された点群シーケンスを用いた評価実験では、提案手法は従来のフレーム間幾何符号化手法に比べて 3.29%優れた圧縮性能を達成した。

1 はじめに

3 次元センシング技術の発展に伴い、点群圧縮が大きな注目を集めている [1–7]。点群は最も代表的な 3 次元データ形式の一つであり、没入型テレプレゼンス [8] や自律走行車 [9] 等の様々な 3 次元センシング応用技術に適した表現を与える。一方、点群は一般に多数の 3 次元点で構成されるため、その蓄積や伝送には大きな記憶容量や通信帯域を必要とする。そのため、点群を用いる技術の実用化においては、点群圧縮が必須となる。

国際標準化団体 Moving Picture Expert Group (MPEG) は 2017 年から点群圧縮の標準化活動を開始した [10,11]。この活動において、点群はその特性から 3 つのカテゴリに分類される。カテゴリ 1 は静的な物体およびシーンを表す点群、カテゴリ 2 は動的な物体を表す点群、カテゴリ 3 は動的に取得された点群である。また、点群を構成する各点は幾何情報と属性情報で表現されるため、点群の符号化は幾何符号化 (geometry coding) と属性符号化 (attribute coding) に区分される。本稿では、動的に取得された点群の無損失幾何符号化に焦点を当てる。

この課題に対し、MPEG は Geometry-based Point Cloud Compression (G-PCC) [12,13] と呼ばれる手法を開発した。この手法は八分木符号化法 [14] に基づいて、3 次元空間における点の位置を効率的に符号化する。更に、この手法はより優れた圧縮性能を実現するために、空間的相関を用いたエントロピー符号化や孤立点のダイレクト符号化のようないくつかのモジュールを導入する。しかし、G-PCC はフレーム内符号化手法の一種であるため、時間的相関が利用されていない。



(a) 全体図

(b) 拡大図

図 1 現フレームの点群 (赤い点) と動き補償後の参照フレームの点群 (青い点) の間の幾何的誤差

密な動的物体点群を対象とし、時間的相関を利用することによって圧縮性能を改善する方法については、多くの研究がある [15–20]。一方、疎な動的取得点群を対象とする場合については、そのような研究はほとんどない。S. Lasserre は任意の点群に対して頑健な幾何符号化を実現するために、無損失フレーム間幾何符号化手法を提案した [21]。これは、G-PCC に対して動き補償と占有予測を導入する手法である。しかし、図 1 に示されるように、正確な動き補償の後であっても、現フレームの点群と参照フレームの点群の間には小さな幾何的誤差が残る。この誤差は占有予測の効果を低下させ、結果として圧縮性能を制限する。

本稿では、無損失フレーム間幾何符号化による点群の圧縮性能を改善するための点の拡張手法を提案する。提案手法では、現フレームの点群と参照フレームの点群の幾何的誤差を補正するオフセットの集合を用いて、参照フレームに新たな点を生成する。このオフセットは、参照フレームの点の位置を基準として、生成される点の相対位置を表す。オフセットの集合は、オフセット候補の集合から訓練用の点群データの平均符号量を削減するものを選択することによって構築する。また、参照フレームの各点に対して適切なオフセットの集合を使用するために、測域センサの光線方向を表すクラスへ点を分類する。そして、点の属するクラスごとに異なるオフセット集合を用いて、点を生成する。

本研究の貢献は以下に要約される。

- フレーム間の幾何的誤差を補正するオフセットに基づく点の拡張手法を提案する。
- 適切なオフセット集合を使用するために、測域センサの光線方向による点分類を導入する。
- 多数のフレームにわたって符号量を削減するオフセット集合の構築手法を提案する。

本稿の以降の構成は次の通りである。第 2 章では、提案手法のベースとなるフレーム間幾何符号化手法を説明する。第 3 章では、圧縮性能を改善するための点の拡張手法を提案する。第 4 章で提案手法の有効性を評価し、第 5 章でまとめを述べる。

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

2 フレーム間幾何符号化

本稿で提案する手法は、無損失フレーム間幾何符号化手法 [21] のフレームワークをベースとする。これは G-PCC [12, 13] に対して、動き補償と占有予測のモジュールを導入するフレームワークである。本節では、このフレームワークを簡潔に説明する。

図 2 に、このフレームワークのブロック図を示す。初めに、現フレーム (current frame) の点群と参照フレーム (reference frame) の点群が与えられる。これらの点群に対しては、座標を整数値で表現するために、予めボクセルを用いた量子化が行なわれていると想定する。参照フレームは、現フレームより時系列的に一つ前のフレームである。これらの点群間では、センサの動きに起因してローカル座標系が一致しない可能性がある。そのため、グローバル動き補償 (global motion compensation) [22] によって 2 つの点群の位置合わせを行なう。その後、八分木符号化法 (octree coding) に基づいて現フレームの点群から再帰的に現ノード (current node) の再分割と占有/非占有を表す 8 ビット符号の生成を行なう。ここで、現ノードは 3 次元空間上での一部の領域に対応しており、空間的境界を持つ。ノードの再分割は、この境界内部の空間を再帰的に 8 つの部分空間 (子ノード) に分割することを表す。また、生成される 8 ビット符号は階層を持つデータ構造として表現され、この階層はノードが再分割された回数に対応する。破線のブロックは、八分木符号化法に基づいて再帰的に処理することを表す。ユーザが指定する特定の階層では、ローカル動き補償 (local motion compensation) [23] によって現ノードの点群と参照フレームの探索ウィンドウ内の点群の位置合わせを行う。現ノードが与えられた場合、参照フレームの点群から現ノードの空間的境界内部に含まれる一部の点群 (部分点群) を取得する (subset points obtaining)。そして、この部分点群を用いて、占有予測 (occupancy prediction) [24] に基づいてコンテキストを生成する。このコンテキストを用いて、現ノードにおける占有/非占有を表す 8 ビット符号をコンテキストに基づく算術符号化 (context-based arithmetic coding) [25] によって算術的に符号化する。そして、この符号化によって得られたビットストリーム (bitstream) を出力する。

以下では、動き補償、占有予測、コンテキストに基づく算術符号化について詳細に説明する。

2.1 動き補償

動き補償は、グローバル動き補償とローカル動き補償の組み合わせで構成される。

グローバル動き補償は、点群全体を用いて剛体変換を推定することによって、参照フレームの点群と現フレームの点群の位置合わせを行なう。剛体変換は 3 次元の回転行列 $\mathbf{R}^{\text{gb}} \in \mathbb{R}^{3 \times 3}$ と並進ベクトル $\mathbf{t}^{\text{gb}} \in \mathbb{R}^3$ で表される。これらの要素は元々は浮動小数値として表現されるが、圧縮のために整数値に量子化し、算術符号化する。

ローカル動き補償は、点群の部分集合を用いて並進ベクトル $\mathbf{t}^{\text{loc}} \in \mathbb{R}^3$ のみを推定することにより、参照フレームの点群の一部と現フレームの点群の一部の位置合わせを行なう。この処理は八分木符号化の過程で、ユーザが指定する階層において各ノードに対して 1 度だけ実行する。この階層では、現ノードの点群とそれに対応する

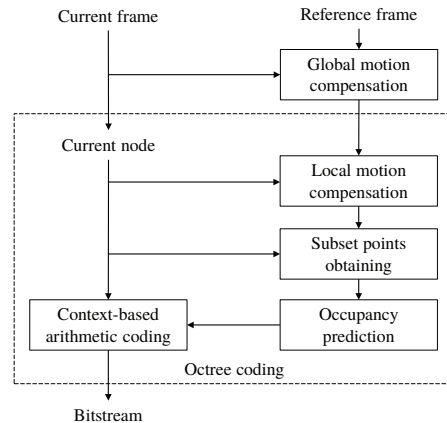


図 2 フレーム間幾何符号化手法のフレームワーク

3 次元探索ウィンドウの内部に位置する参照フレームの点群の間で、歪みを最小化する並進ベクトルを探索する。一方、この階層以外では、この処理はスキップする。並進ベクトル $\mathbf{t}^{\text{loc}} \in \mathbb{R}^3$ は、HEVC [26] における動きベクトル符号化と同様の方法を用いて算術符号化する。

2.2 占有予測

占有予測は、参照フレームの点群から現フレームの点群の占有状態 (occupancy) をノード単位で予測する。現ノードの空間的境界が与えられた場合、参照フレームの点群からその空間的境界の内部に位置する部分点群を取得する。この部分点群を現ノードの再分割と同様に再分割することにより、現ノードの 8 つの子ノードに対応する占有/非占有の予測 $p_i \in \{0, 1\}$ および点の個数の予測 $n_i^{\text{pnt}} \in \mathbb{N}^0$ ($i = 1, \dots, 8$) を得る。このインデックス i は、現ノードの 8 つの子ノードのインデックスに対応する。そして、“no pred”, “pred0”, “pred1”, “predL” のいずれかで表される予測器 (predictor) P_i を生成する。no pred は、不適切な予測を防止するための、予測の非活性化を表す予測器である。この予測器は、現階層の一つ前の階層において、現ノードの親ノードが占有予測を誤った回数 $n^{\text{err}} \in \{0, \dots, 8\}$ が閾値 th^{err} より大きい場合に生成される。また、この予測器は p_i が全て 0 の場合にも生成される。この場合、8 つ予測器全てを no pred とする。pred0 および pred1 はそれぞれ子ノードを非占有および占有と予測することを表す。predL は多くの点を含む占有と予測することを表し、点の個数の予測 n_i^{pnt} が閾値 th^{pnt} より大きい場合に生成する。最後に、この処理はノードごとに予測器の集合 $\mathcal{P} = \{P_i\}_{i=1}^8$ を出力する。

2.3 コンテキストに基づく算術符号化

コンテキストに基づく算術符号化は、現ノードにおける占有/非占有を表す 8 ビット符号をコンテキストを用いて算術符号化する。8 ビット符号が与えられた場合、コンテキストに基づいて適応的に選択した確率分布を用いて、8 つのビットを順番に算術符号化する。このコンテキストは、ビットパターン、近傍構成、占有予測器を用いて生成する。ビットパターンはこれまでに符号化されたビットのパターンであり、全てのビットが 0 の場合を除いて、[1, 255] の範囲の整数で表される。近傍構成は、現ノードと 1 つの面を共有する 6 つの近傍ノードの占有パターンであり、 $2^6 = 64$ 通りのパターンが考えら

れる。ただし、コンテキストの追跡を容易にするために、3次元空間での回転不変性および垂直方向への反転不変性を仮定することによって、パターン数を10通りまで削減する。更に、ビットパターンと近傍構成を組み合わせ、状態と呼ばれる表現に変換する。この際、これらの組み合わせに応じて状態のパターン数を削減する。占有予測器は、2.2節で述べた4通りの予測器である。 i 番目のビットに対応するコンテキスト X_i を生成する際には、初めに状態 $s_i = f(b_i, n^{\text{nbr}})$ を計算する。ここで、 b_i は i 番目のビットまでのビットパターン、 n^{nbr} は近傍構成、 f はビットパターンと近傍構成の組み合わせに応じて状態を出力する関数である。そして、状態 s_i および占有予測器 P_i からコンテキストへのマッピング関数 g を用いて、コンテキスト $X_i = g(s_i, P_i)$ を得る。

3 提案手法

無損失フレーム間幾何符号化による点群の圧縮性能を改善するために、参照フレームの点を拡張する手法を提案する。測域センサで取得された点群には、センサ自体の計測精度やセンサの動き等に起因による小さな幾何的誤差が含まれる [27–29]。そのため、正確な動き補償の後であっても、現フレームの点群と参照フレームの点群の座標は正確には一致しない。この問題は、特に深い八分木階層において占有予測の効果を低下させ、結果として圧縮性能を制限する。より多くの階層でローカル動き補償を実行することによってこの問題に対処できる可能性はあるが、その場合には並進ベクトル t^{loc} を表すサイド情報が増加する。それに対して、提案手法ではサイド情報を増加させずに、この問題に対処する。

提案手法は、フレーム間の幾何的誤差を補正するオフセットに基づいて参照フレームの点（参照点）を拡張する。点拡張は、事前に構築したオフセット集合を用いて行なう。測域センサは方位角と極角で分類可能な方向への光線を用いて点群を計測するため、その分類に応じて最適なオフセット集合が異なる可能性がある。そのため、光線の方向を表す K 通りのクラスに対して独立にオフセット集合を構築する。オフセット集合を選択するために、参照点を K 通りのクラスのいずれかに分類する。その後、参照点を自身のクラスに対応するオフセット集合を用いて拡張する。この K 通りのオフセット集合は予めエンコーダおよびデコーダに共通であり、サイド情報として伝送しないと想定する。

図3に、提案手法のフレームワークを示す。初めに、参照点（reference points）が与えられる。点分類（point classification）によって各点が属するクラスを求め、 K 通りのオフセット集合（ K sets of offsets）から対応するオフセット集合を取得する。そして、このオフセットを用いて点拡張（point augmentation）を行なうことにより、拡張された参照点（augmented reference points）を得る。

オフセット集合は、現フレームの点群と座標が一致する点を参照フレームの点群から生成することを目的として構築する。そのため、訓練用の点群データ（訓練点群）から現フレームの点群と参照フレームの点群の座標の誤差を補正するオフセットの集合を収集する。ただし、オフセットは常に点群の符号量を削減するとは限らないため、収集したオフセットの集合から、多数の訓練点群の平均符号量を削減するオフセットを選択する。

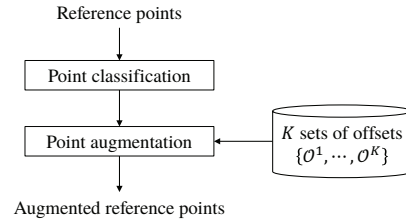


図3 提案手法のフレームワーク

以下では、光線方向による点分類、オフセットに基づく点拡張、オフセット集合の構築について詳細に説明する。

3.1 光線方向による点分類

光線方向による点分類は、方位角 $\phi \in [0^\circ, 360^\circ)$ および極角 $\theta \in [0^\circ, 180^\circ)$ を用いて、参照点 $\mathbf{p} \in \mathbb{R}^3$ を分類する。参照点 \mathbf{p} は通常、3次元直交座標系における座標を表す。そのため、参照点 \mathbf{p} の座標を3次元極座標系における座標に変換することにより、方位角 ϕ および極角 θ を得る。方位角 ϕ および極角 θ はそれぞれ、 x 軸方向および z 軸方向を基準とする。そして、これらを等間隔の量子化範囲を用いて量子化することにより、量子化された方位角 $q_\phi \in \mathbb{N}^0$ および極角 $q_\theta \in \mathbb{N}^0$ を得る。最後に、これらの値を組み合わせることにより、参照点 \mathbf{p} が属するクラスのインデックス $k \in \{1, \dots, K\}$ を得る。

提案手法では、オフセット集合はこのクラスごとに独立に構築し、各参照点の属するクラスに対応するオフセット集合を用いて点拡張を実行する。

3.2 オフセットに基づく点拡張

オフセットに基づく点拡張は、クラスのインデックス k およびそれに対応するオフセット集合を用いて、参照点を増加させる。 k 番目のクラスに属する参照点、 k 番目のクラスに対応するオフセット集合、拡張された点の集合をそれぞれ $\mathbf{p}^k \in \mathbb{R}^3$ 、 $\mathcal{O}^k = \{\mathbf{o}_m^k \in \mathbb{R}^3\}_{m=1}^{M^k}$ 、 $\mathcal{Q}^k = \{\mathbf{q}_m^k \in \mathbb{R}^3\}_{m=1}^{M^k}$ とする。ここで、各点および各オフセットは3次元のベクトルとして表現する。この時、拡張された m 番目の点は $\mathbf{q}_m^k = \mathbf{p}^k + \mathbf{o}_m^k$ と表すことができる。オフセット \mathbf{o}_m^k は点 \mathbf{p}^k の位置を基準として、点 \mathbf{q}_m^k を生成する相対位置を示す。したがって、点 \mathbf{p}^k ごとに M^k 個の点を生成し、それらを全て参照点として扱う。

この処理は、図2におけるローカル動き補償の後に実行する。グローバル動き補償とローカル動き補償はどちらも参照点の点数の増加に伴って処理時間が大幅に増加する。提案手法では、処理時間の大幅な増加を回避するために、これらの動き補償を行なった後に参照点を拡張する。2.1節で述べた通り、ローカル動き補償はユーザが指定した階層で各ノードに対して1度だけ実行する。そのため、点拡張もその階層で各ノードに対して1度だけ実行する。

3.3 オフセット集合の構築

本節では、訓練点群を用いたオフセット集合の構築手法を提案する。訓練点群は、現フレームの点群と、それに対応する動き補償を実行した後の参照フレームの点群の組の集合からなる。提案手法では、 K 通りのクラスに対応する K 通りのオフセット集合を構築する。この構

築は、2つのステップで構成される。第一のステップはオフセット候補の収集、第二のステップはオフセット候補集合からのオフセットの選択である。

第一のステップでは、訓練点群から K 通りのオフセット候補の集合を収集する。初めに、参照点のそれぞれについて、現フレームの点群から最近傍点を探索する。そして、その最近傍点の座標から参照点の座標を減算して得られるベクトルを、オフセット候補とみなす。その後、3.1 節で述べた方法で参照点の属するクラスを計算し、オフセット候補をそのクラスに蓄積する。また、各オフセット候補の出現頻度を数える。同一のオフセット候補が複数回出現した場合には、一つを残して重複を削除する。訓練点群の全てについて同様の処理を行なうことにより、オフセット候補の集合を収集する。

第二のステップでは、 K 通りのオフセット候補集合から、訓練点群の符号量を最小化する K 通りのオフセットの集合を選択する。基本的な考えは、オフセット候補を利用する前後で訓練点群の平均符号量が削減されるか否かを検証し、削減される場合にはそのオフセット候補をオフセットとして選択することである。オフセットは、占有予測における true positive の増加と false negative の減少をもたらす一方で、true negative の減少と false positive の増加を引き起こす可能性がある。前者の2つと後者の2つはそれぞれ、符号量を削減および増加する効果が予想される。現フレームの点と参照フレームの点は多くの点が1対1で対応するため、生成する点の個数を増やすほど後者の効果が大きくなる。そのため、選択するオフセットの組み合わせの最適化は重要である。効果的なアプローチの一つとして、貪欲法 [30] を用いて候補ベクトル集合から最も平均符号量を削減するオフセット候補を順番に選択することが考えられる。しかし、多数のオフセット候補に対しては、このアプローチの計算量が大きくなる。計算量を抑制するために、提案手法は出現頻度の高い上位 N 個のオフセット候補に対して、順番に検証を行なう。出現頻度の高いオフセット候補はより多くの点群において幾何的誤差を補正することができるため、平均符号量を削減することが期待される。また、訓練点群に対しては事前に動き補償を実行しているため、検証では動き補償をスキップする。更に、提案手法は、 k 番目のクラスに対するオフセット集合を構築する際には、現フレームの点群と参照フレームの点群における k 番目のクラスに属する部分点群のみを使用する。これは検証に使用する点群の点数を小さくするため、計算量が削減される。オフセットの選択はクラスごとに独立に実行するため、これらの部分点群のみを用いて適切に検証を行なうことができる。

アルゴリズムの疑似コードを Algorithm 1 に示す。入力は、 K 通りのオフセット候補集合 $C = \{C^1, \dots, C^K\}$ および訓練点群 $\mathcal{T} = \{\{\mathcal{U}_1, \mathcal{V}_1\}, \dots, \{\mathcal{U}_S, \mathcal{V}_S\}\}$ である。ここで、 \mathcal{U}_s および \mathcal{V}_s ($s = 1, \dots, S$) はそれぞれ現フレームの点群とそれに対応する参照フレームの点群を表す。出力は、 K 通りのオフセット集合 $O = \{O^1, \dots, O^K\}$ である。初めに、各 O^k ($k = 1, \dots, K$) を空集合へ初期化する。また、 \mathcal{U}_s および \mathcal{V}_s から k 番目のクラスに属する部分点群 \mathcal{U}_s^k および \mathcal{V}_s^k を生成する。そして、 \mathcal{V}_s^k を用いた場合の \mathcal{U}_s^k の平均符号量を計算する。これは、 k 番目のクラスにおける平均符号量の最小値の初期値として

Algorithm 1 K 通りのオフセット集合の構築

Input: K sets of offset candidates $C = \{C^1, \dots, C^K\}$ and training point clouds $\mathcal{T} = \{\{\mathcal{U}_1, \mathcal{V}_1\}, \dots, \{\mathcal{U}_S, \mathcal{V}_S\}\}$

Output: K sets of offsets $O = \{O^1, \dots, O^K\}$

```

1: for  $k = 1$  to  $K$  do
2:    $O^k \leftarrow \emptyset$ 
3:   for  $s = 1$  to  $S$  do
4:      $\mathcal{U}_s^k \leftarrow$  subset of  $\mathcal{U}_s$  that belongs to the  $k$ -th class
5:      $\mathcal{V}_s^k \leftarrow$  subset of  $\mathcal{V}_s$  that belongs to the  $k$ -th class
6:      $r_s^k \leftarrow$  bit rate of  $\mathcal{U}_s^k$  when using  $\mathcal{V}_s^k$ 
7:   end for
8:    $r_{\min}^k \leftarrow$  average value of  $r_s^k$  for all  $s$ 
9: end for
10: for  $k = 1$  to  $K$  do
11:   for  $n = 1$  to  $N$  do
12:      $c_n^k \leftarrow$   $n$ -th offset candidate in  $C^k$ 
13:     for  $s = 1$  to  $S$  do
14:        $r_s^k \leftarrow$  bit rate of  $\mathcal{U}_s^k$  when using  $\mathcal{V}_s^k$ ,  $c_n^k$ , and  $O^k$ 
15:     end for
16:      $r_{\text{ave}}^k \leftarrow$  average value of  $r_s^k$  for all  $s$ 
17:     if ( $r_{\text{ave}}^k < r_{\min}^k$ ) then
18:        $r_{\min}^k \leftarrow r_{\text{ave}}^k$ 
19:       Insert  $c_n^k$  to  $O^k$ 
20:     end if
21:   end for
22: end for

```

r_{\min}^k に代入する。これらの初期化の後に、クラスごとに N 個のオフセット候補を順番に検証する。各検証において、 C^k の n 番目のオフセット c_n^k と、現在の O^k を用いて点拡張を行なう場合の平均符号量を計算する。その値が現在の平均符号量の最小値 r_{\min}^k よりも小さい場合、 r_{\min}^k をその値に更新する。更に、 c_n^k を O^k に挿入する。このようにして、オフセット集合を構築する。

4 評価実験

本章では、動的に取得された点群を用いて、提案手法の有効性を実験的に評価する。

4.1 実験設定

評価のために、MPEG で利用されている LiDAR によって動的に取得された7つの点群シーケンスである “ford_01_q1mm”, “ford_02_q1mm”, “ford_03_q1mm”, “qnxadas-junction-approach”, “qnxadas-junction-exit”, “qnxadas-motorway-join”, “qnxadas-navigating-bends” を使用する。各シーケンスに含まれるフレームの数は、ford グループが 1,500, qnxadas-junction グループが 74, qnxadas-motorway-join が 811, qnxadas-navigating-bends が 300 である。これらのシーケンスにおいて、点群は解像度 1 mm のボクセルを用いて予め量子化されている。

各シーケンスを評価するために、同一グループの他のシーケンスを訓練点群として使用する。例えば、ford_01_q1mm を評価する場合、ford_02_q1mm を訓練点群として使用する。qnxadas-motorway-join と qnxadas-navigating-bends はお互いを訓練点群として使用する。訓練には、等間隔でサンプリングした 100 組の現フレームの点群とそれに対応する参照フレームの点群のみを

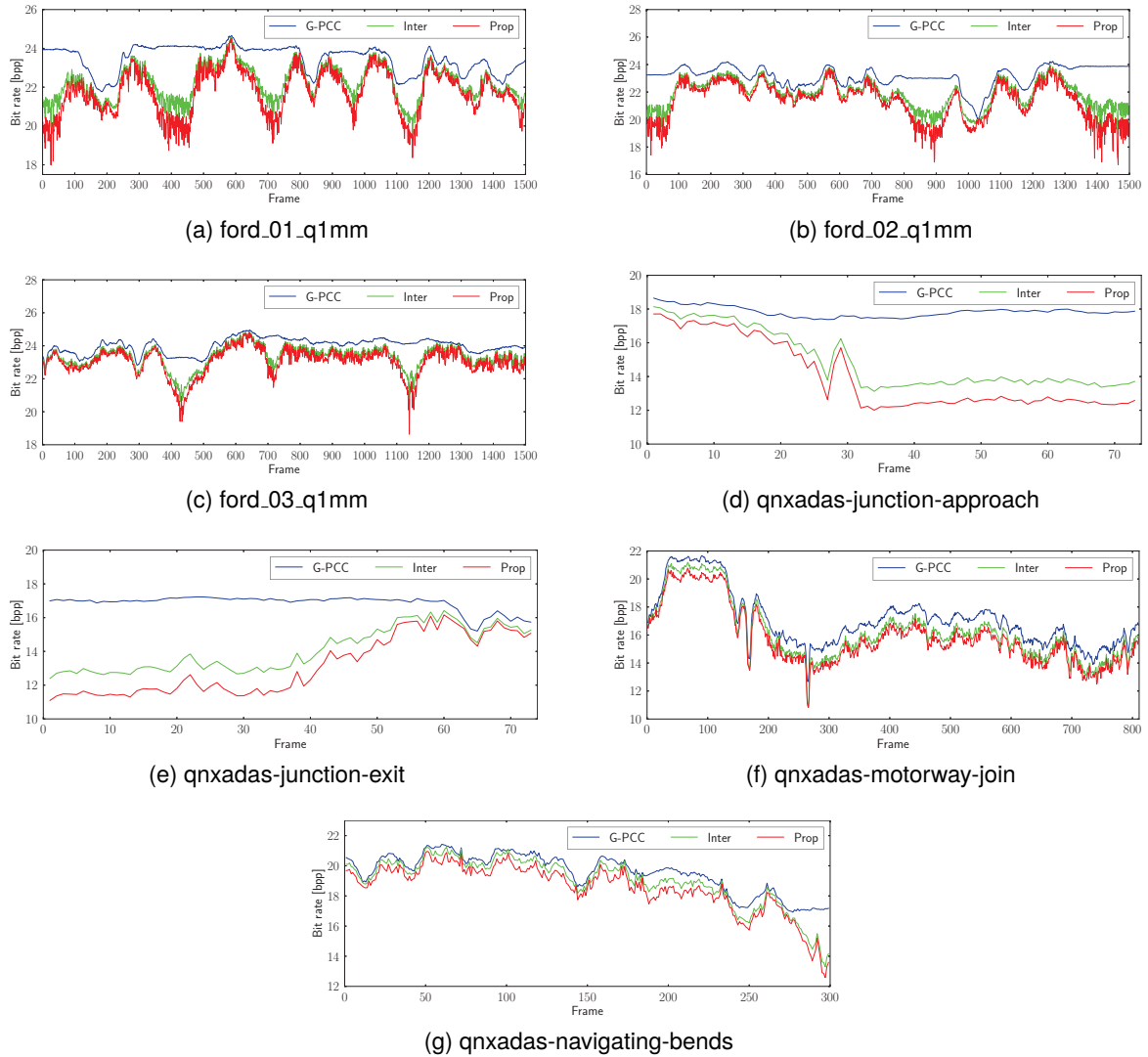


図4 フレームごとの符号量 [bpp] に関する提案手法と従来の無損失幾何符号化手法の比較

使用する。ただし、qnxadas-junction グループには 74 フレームしか含まれないため、73 組の点群を使用する。

点分類のために、方位角 $\phi \in [0^\circ, 360^\circ)$ および極角 $\theta \in [0^\circ, 180^\circ)$ のそれぞれに対して 5° 間隔の量子化範囲を設定する。したがって、個々のクラス数は 72 および 36、それらの組み合わせによるクラス数は $K = 2,592$ となる。オフセット集合の構築のために、クラスごとの候補オフセットの最大数を $N = 1,000$ に設定する。全ての実験は、クロック周波数 3.7 GHz の Intel Core i9-10900X CPU と 32 GB のメモリを搭載した PC 上で実行する。

4.2 圧縮性能

初めに、提案手法による点群の圧縮性能を評価する。本実験では、提案手法 (Prop) と、八分木符号化法 (Octree), G-PCC [12, 13], 無損失フレーム間幾何符号化手法 (Inter) [21] の無損失圧縮性能を比較する。

表 1 に、各シーケンスにおける bits per point (bpp) 単位で表される平均符号量と、提案手法の Inter に対するゲインをまとめる。Octree の符号量は大きく、平均で 50.05 bpp である。G-PCC は Octree に比べて、全ての

表 1 平均符号量 [bpp] に関する提案手法と従来の無損失幾何符号化手法の比較。最終列は提案手法の Inter に対するゲインを表す

	Octree	G-PCC	Inter	Prop	Gain
ford_01_q1mm	55.84	23.48	22.23	21.76	2.10%
ford_02_q1mm	55.01	23.15	21.87	21.35	2.38%
ford_03_q1mm	56.87	24.00	23.44	23.05	1.19%
qnxadas-junction-approach	45.38	17.84	14.97	14.08	5.95%
qnxadas-junction-exit	44.35	16.86	14.08	13.17	6.47%
qnxadas-motorway-join	45.11	17.10	16.22	15.80	2.59%
qnxadas-navigating-bends	47.77	19.56	18.99	18.55	2.32%
Average	50.05	20.29	18.83	18.25	3.29%

シーケンスにおいて大幅な改善を示している。Inter は、参照フレームの点群を用いた動き補償と占有予測を導入することによって、圧縮性能を更に改善する。提案手法は全てのシーケンスにおいて最小の符号量を達成する。全てのシーケンスに対する符号量の平均は 18.25 bpp であり、Inter に対するゲインは 3.29% である。

G-PCC, Inter, 提案手法を詳細に比較するために、図 4 にこれらの手法のフレームごとの符号量を示す。Inter

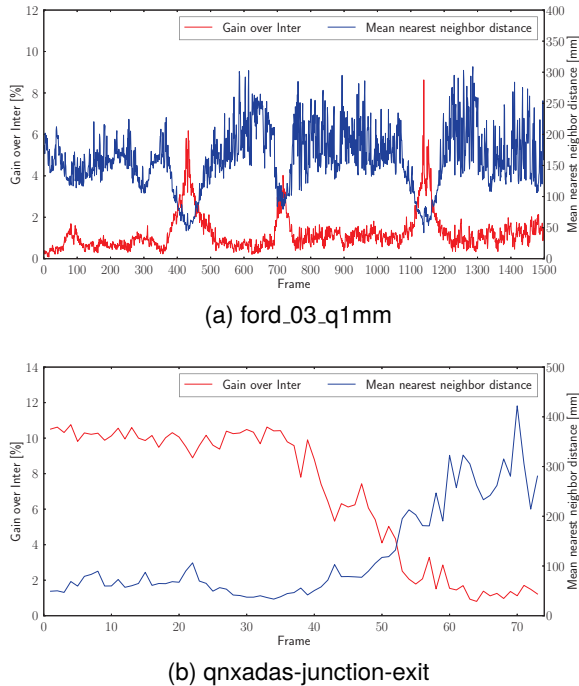


図5 フレームごとのInterに対する提案手法のゲインと平均最近傍距離の関係

は、ほとんどのフレームでG-PCCと比べて符号量を大きく削減する。Interは、占有予測の精度が高くなる場合に圧縮性能が向上する傾向がある。提案手法は、Interと類似した結果を示すが、常に符号量をより多く削減する。これは、提案した点拡張によってフレーム間の幾何的誤差に対処することにより、占有予測の効果が向上するためである。

また、提案手法のInterに対するゲインについて分析する。図5に、最も平均ゲインの小さいシーケンスであるford_03_q1mmと、最も平均ゲインの大きいシーケンスであるqnxadas-junction-exitにおける、フレームごとのゲインを示す。更に、各フレームにおける動き補償後の参照点から現フレームの点群への平均最近傍距離 (Mean Nearest Neighbor Distance; MNND) を重畳する。この図より、ゲインとMNNDは一方が増加すればもう一方が低下する関係にあることがわかる。これは、提案手法によって小さな幾何的誤差が補正されることによって、圧縮性能が改善されることを示している。図5(a)では多くのフレームでMNNDが大きいために、ゲインが小さくなる。一方、図5(b)では、MNNDが小さい場合に大きなゲインが継続的に得られることが示される。本実験においては、10%を超える大きなゲインは、MNNDが100mmを下回るフレームから得られる。このようなフレームは、現フレームの点群と参照フレームの点群の間で幾何的変形や位置合わせの誤差が少ない場合に得られる。提案手法は、そのような場合において特に効果的である。

4.3 処理時間

本節では、提案手法とInterにおける圧縮に要する処理時間を比較する。図6に、各シーケンスに対する処理時間をバイオリン図で示す。fordグループは他のシーケ

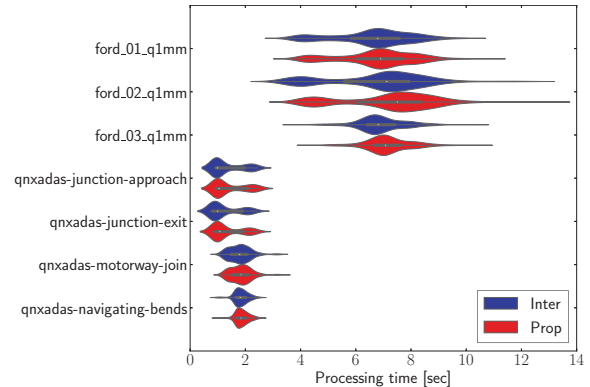


図6 シーケンスごとの処理時間の比較

ンスに比べて点群の点数が多いため、処理時間が長くなる。提案手法は、全てのシーケンスに対して処理時間を増加させることがわかる。これは、提案した点拡張によって参照点の点数が増加するためである。これにより、現ノードの空間的境界の内部に位置する参照点の取得や、それらの再分割に要する処理時間が増加する。元々の参照フレームの点群の点数が多いシーケンス (すなわち、fordグループ) に対しては、処理時間が比較的大きく増加する傾向が見られる。しかし、提案手法は動き補償の後に点拡張を実行するため、圧縮に要する処理時間は大幅には増加しない。

4.4 占有予測結果の分析

本節では、Interおよび提案手法における占有予測について分析する。図7に、7つのシーケンスの全てのフレームにわたる予測器no pred, pred0, pred1, predLの平均生成割合を示す。図7(a)より、Interでは予測の非活性化を表すno predが高い割合を持つことがわかる。疎な点群に対する占有予測では、2.2節で述べた2つの条件のいずれかが満たされることにより、頻繁に予測が非活性化される。対照的に、図7(b)より、提案手法ではno predの割合が大幅に削減されることがわかる。これは提案した点拡張によって、2つの条件に起因する非活性化が抑止されるためである。その結果、占有予測に基づく圧縮性能の改善効果が向上する。また、Interと提案手法を比較すると、pred1の割合が減少し、predLの割合が増加していることがわかる。これは、点拡張によって点の個数の予測 n_i^{pm} が閾値 th^{pm} より大きくなる頻度が増加するためである。

更に、true positive (TP), true negative (TN), false positive (FP), false negative (FN)の観点から占有予測の結果を分析する。これらの測定においては、現ノードと、その空間的境界の内部に位置する参照点のみを使用する。2.2節で述べた通り、これら以外の点は占有予測の結果に影響を与えないため、この測定から除外する。図8に、7つのシーケンスの全てのフレームにわたるこれらの平均割合を示す。Interと比べて、提案手法はTNの減少とFPの増加を引き起こす一方で、TPの増加とFNの減少をもたらすことがわかる。これらは全て提案した点拡張によるものである。提案手法は単一の参照点から複数の点を生成するため、前者の2つが不可避的に発生する。一方、現ノードにおいて占有となる子ノードの空間

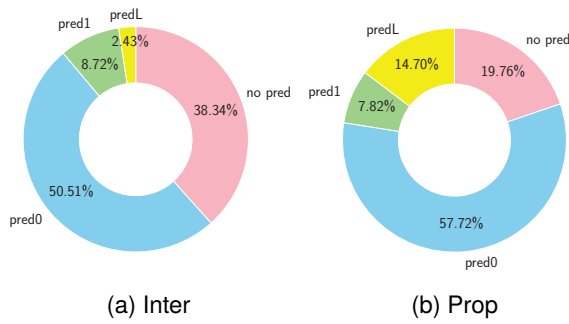


図7 占有予測における予測器 no pred, pred0, pred1, predL の生成割合

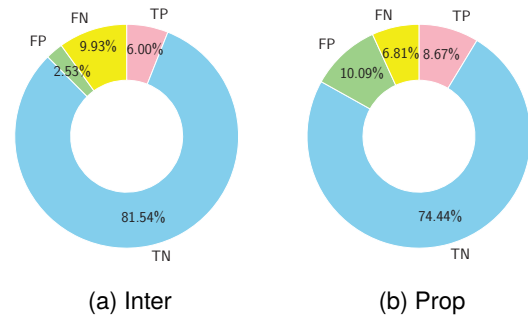


図8 占有予測における true positive (TP), true negative (TN), false positive (FP), false negative (FN) の割合

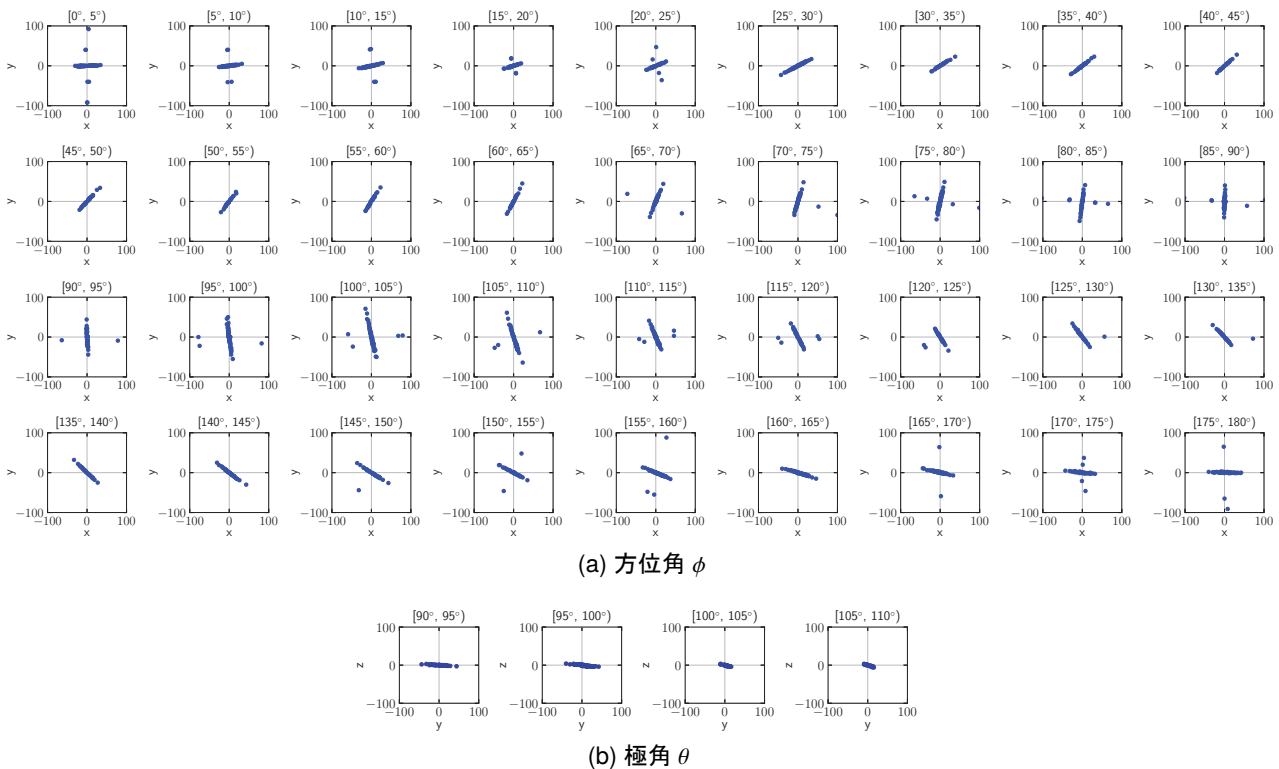


図9 方位角 ϕ と極角 θ に焦点を当てたクラスごとのオフセット集合の可視化

的境界の内部に位置する参照点を生成するため、後者の2つを達成する。提案手法では、Algorithm 1に記載の構築手法によって、多数のフレームにわたって平均符号量を削減するオフセットのみを選択する。そのため、この点拡張によって圧縮性能を改善することができる。

4.5 オフセット集合の可視化

本節では、提案手法によって構築したオフセット集合を可視化する。図9に、ford_01_q1mmシーケンスから構築したオフセットの集合を示す。本実験では、方位角 ϕ と極角 θ に焦点を当てたいいくつかのクラスを選択する。図9(a)は方位角 ϕ に焦点を当てている。ここでは、極角 θ を 90° に固定し、量子化された方位角 ϕ ごとのオフセット集合を x - y 平面上に描画する散布図を示している。各散布図の上部に記載した範囲は、方位角 ϕ の量子化範囲を表している。 0° から 180° までの散布図と 180° から 360° までの散布図は類似していたため、後者の範

囲は割愛する。この図より、多くのオフセットは各方位角 ϕ の方向に向かって分布することがわかる。これは、幾何的誤差が測域センサの光線方向に対して多く発生することを示している。クラスごとのオフセットの分布の差異は、訓練点群の幾何的誤差の差異に起因する。図9(b)は θ に焦点を当てている。ここでは、方位角 ϕ を 90° に固定し、量子化された極角 θ ごとのオフセット集合を y - z 平面上に描画する散布図を示している。各散布図の上部に記載した範囲は、極角 θ の量子化範囲を表している。 90° から 110° の範囲にしかオフセット集合が構築されなかったため、それ以外の範囲は割愛する。これは、測域センサの計測範囲や、計測された環境に起因するものである。この図においても、方位角 ϕ の場合と同様に多くのオフセットは各極角 θ の方向に向かって分布することがわかる。一方、ほとんどの点が 90° から 100° の範囲に属するために、それ以外のクラスでは光線方向への広がり小さくなる。

5 まとめ

本稿では、無損失フレーム間幾何符号化による点群の圧縮性能を改善するための点の拡張手法を提案した。この手法は測域センサにおける光線方向によって点を分類し、そのクラスごとに構築されたオフセット集合を用いて点を生成する。オフセット集合は、訓練用の点群データから収集されたオフセット候補集合から、多数のフレームにわたって平均符号量を削減するオフセットを選択することによって構築した。このオフセット集合は、点の分類ごとに独立に構築した。評価実験では、提案手法は従来の無損失フレーム間幾何符号化手法に比べて平均で3.29%優れた圧縮性能を達成した。より詳細な分析では、提案した点拡張には、疎な点群における予測の非活性化率を削減する効果があることが示された。今後は、より大きな幾何的誤差に対処するための手法について検討する。

参考文献

- [1] T. Golla and R. Klein, "Real-time point cloud compression," in Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA), pp.5087–5092, 2015.
- [2] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), vol.27, no.4, pp.828–842, 2016.
- [3] L. Li, Z. Li, V. Zakharchenko, J. Chen, and H. Li, "Advanced 3D motion prediction for video-based dynamic point cloud compression," IEEE Transactions on Image Processing (TIP), vol.29, pp.289–302, 2019.
- [4] C. Cao, M. Preda, and T. Zaharia, "3D point cloud compression: A survey," in Proceedings of the 24th International ACM Conference on 3D Web Technology (Web3D), pp.1–9, 2019.
- [5] X. Zhang, W. Gao, and S. Liu, "Implicit geometry partition for point cloud compression," in Proceedings of 2020 IEEE Data Compression Conference (DCC), pp.73–82, 2020.
- [6] 西尾孝治, 竹林佑介, 手島裕詞, 金谷孝之, 小堀研一, "情報量削減を目的とした点群モデルの情報変換," 第7回情報科学技術フォーラム (FIT2008), 2008.
- [7] 松崎康平, 柳原広昌, "3次元点群を用いた車両の自己位置推定のための圧縮地図表現," 電子情報通信学会論文誌 D, vol.101, no.11, pp.1471–1481, 2018.
- [8] K. Higuchi, Y. Chen, P.A. Chou, Z. Zhang, and Z. Liu, "Immerseboard: Immersive telepresence experience using a digital whiteboard," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI), pp.2383–2392, 2015.
- [9] R. Changalvala and H. Malik, "LiDAR data integrity verification for autonomous vehicle," IEEE Access, vol.7, pp.138018–138031, 2019.
- [10] MPEG 3DG, "Call for proposals for point cloud compression v2", ISO/IEC JTC 1/SC 29/WG 11 N16763, 2017.
- [11] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P.A. Chou, R.A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, *et al.*, "Emerging MPEG standards for point cloud compression," IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), vol.9, no.1, pp.133–148, 2018.
- [12] 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 Transactions on Signal and Information Processing, vol.9, 2020.
- [13] L. Cui, R. Mekuria, M. Preda, and E.S. Jang, "Point-cloud compression: Moving picture experts group's new standard in 2020," IEEE Consumer Electronics Magazine (CEM), vol.8, no.4, pp.17–21, 2019.
- [14] D. Meagher, "Geometric modeling using octree encoding," Computer Graphics and Image Processing (CGIP), vol.19, no.2, pp.129–147, 1982.
- [15] D.C. Garcia and R.L. de Queiroz, "Context-based octree coding for point-cloud video," in Proceedings of 2017 IEEE International Conference on Image Processing (ICIP), pp.1412–1416, 2017.
- [16] R.L. de Queiroz and P.A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," IEEE Transactions on Image Processing (TIP), vol.26, no.8, pp.3886–3895, 2017.
- [17] S. Limuti, E. Polo, and S. Milani, "A transform coding strategy for voxelized dynamic point clouds," in Proceedings of 2018 IEEE International Conference on Image Processing (ICIP), pp.2954–2958, 2018.
- [18] C.F. Santos, F. Lopes, A. Pinheiro, and L.A. da Silva Cruz, "A sub-partitioning method for point cloud inter-prediction coding," in Proceedings of 2018 IEEE Visual Communications and Image Processing (VCIP), pp.1–4, 2018.
- [19] 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 Transactions on Image Processing (TIP), vol.29, pp.313–322, 2019.
- [20] E. Peixoto, E. Medeiros, and E. Ramalho, "Silhouette 4D: An inter-frame lossless geometry coder of dynamic voxelized point clouds," in Proceedings of 2020 IEEE International Conference on Image Processing (ICIP), pp.2691–2695, 2020.
- [21] S. Lasserre. "Exploratory model for inter-prediction in G-PCC", ISO/IEC JTC1/SC29/WG11 N18096, 2018.
- [22] S. Lasserre and D. Flynn. "Global motion compensation for point cloud compression in TM3", ISO/IEC JTC1/SC29/WG11 MPEG2018/m44751, 2018.
- [23] S. Lasserre and D. Flynn. "On motion compensation for geometry coding in TM3", ISO/IEC JTC1/SC29/WG11 MPEG2018/m42521, 2018.
- [24] S. Lasserre and D. Flynn. "How to use a predictive set of points for geometry coding in TM3", ISO/IEC JTC1/SC29/WG11 MPEG2018/m42520, 2018.
- [25] S. Lasserre and D. Flynn. "A new binary entropy coder with update for geometry coding in TM3", ISO/IEC JTC1/SC29/WG11 MPEG2018/m44750, 2018.
- [26] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), vol.22, no.12, pp.1649–1668, 2012.
- [27] M. Kim, S. Park, J. Danielson, J. Irwin, G. Stensaas, J. Stoker, and J. Nimetz, "General external uncertainty models of three-plane intersection point for 3D absolute accuracy assessment of lidar point cloud," Remote Sensing, vol.11, no.23, p.2737, 2019.
- [28] J. Laconte, S.P. Deschênes, M. Labussière, and F. Pomerleau, "Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping," in Proceedings of 2019 IEEE International Conference on Robotics and Automation (ICRA), pp.8100–8106, 2019.
- [29] M. Segata, R.L. Cigno, R.K. Bhadani, M. Bunting, and J. Sprinkle, "A lidar error model for cooperative driving simulations," in Proceedings of 2018 IEEE Vehicular Networking Conference (VNC), pp.1–8, 2018.
- [30] D.P. Williamson and D.B. Shmoys, The design of approximation algorithms, Cambridge university press, 2011.