疎な点群の圧縮のための予測器生成における適応的閾値決定

松﨑　康平†　河村　圭†

† 株式会社 KDDI 総合研究所　〒356–8502 埼玉県ふじみ野市大原 2–1–15
E-mail: †{ko-matsuzaki,kei}@kddi-research.jp

あらまし　点群圧縮は，点群を用いた応用技術を実用化するための必須技術である．時間的相関を用いて疎な点群を圧縮する場合，フレーム間の幾何的誤差によって圧縮性能が制限される．参照フレームに点拡張を適用することによってこの問題に対処することが可能であるが，点密度が変化する．これは，参照フレーム内の点数の閾値処理に基づく予測器生成に偏りを生じさせ，結果として予測器の効果を減少させる．本稿では，点拡張に起因して点密度に変化が生じる場合であっても，予測器の偏りを解消することが可能な閾値決定手法を提案する．提案手法では，八分木符号化法におけるノード内の点数の累積移動平均を用いて，階層ごとに適応的に閾値を決定する．動的に取得された点群を用いた評価実験は，提案手法によって予測器の偏りが解消されるとともに，圧縮性能が改善されることを示した．
キーワード　点群圧縮，予測器生成，閾値決定，G-PCC

Adaptive Threshold Determination in Predictor Generation for Compression of Sparse Point Cloud

Kohei MATSUZAKI† and Kei KAWAMURA†

† KDDI Research, Inc.　2–1–15 Ohara, Fujimino-shi, Saitama, 356–8502 Japan
E-mail: †{ko-matsuzaki,kei}@kddi-research.jp

Abstract　Point cloud compression is an essential technique for realizing point cloud applications for practical use. In the case of compressing a sparse point cloud using temporal correlation, compression performance is limited by geometric error between frames. Although this problem can be addressed by applying a point augmentation to the reference frame, the point density changes. This causes a bias in predictor generation based on thresholding of the number of points in the reference frame, resulting reduction the effectiveness of the predictor. In this paper, we propose a threshold determination method that can solve the bias of the predictors even when the point density changes due to the point augmentation. In the proposed method, the threshold value is adaptively determined for each level using the cumulative moving average of the number of points in the node in the octree coding method. Evaluation experiments on dynamically acquired point clouds showed that the proposed method solves the bias of the predictors and improves the compression performance.
Key words　point cloud compression, predictor generation, threshold determination, G-PCC

1. はじめに

点群圧縮 [1]〜[5] は点群に基づく応用技術の実用化における重要な課題であり，2017 年から国際標準化団体 Moving Picture Expert Group（MPEG）において標準化活動が進められている [6], [7]．この活動において，Geometry-based Point Cloud Compression （G-PCC） [8] と呼ばれる八分木表現に基づくフレーム内符号化手法が開発されている．さらに，時間的相関を用いて幾何情報の圧縮性能を改善するために，フレーム間幾何符号化手法 [9] が提案されている．この手法では，時系列的に連続する 2 つのフレームにおいて，現フレームのノードの占有状態を参照フレームから予測することによって圧縮性能を改善する．しかし，この手法では疎な点群に対してはフレーム間の幾何的誤差に起因して，改善効果が制限される．この問題に対処するために，筆者らは参照フレームの点を拡張する手法を提案した [10]．この手法は予測不可となる頻度を削減する一方で，点密度を変化させる．これは，参照フレーム内の点数の閾値処理に基づく予測器生成に偏りを生じさせ，結果として予測器の効果を減少させる．

本稿では，予測器生成の偏りの解消による圧縮性能の改善を目的とする，適応的な閾値決定手法を提案する．この手法では，訓練用の点群データにおいて圧縮性能が最大となるパラメータ

と点の密度情報を用いて，八分木階層ごとに閾値を決定する．点の密度情報としては，これまで処理された八分木階層におけるノード内の点数の累積移動平均を使用する．

2. フレーム間幾何符号化

本節では，本稿でベースとするフレーム間幾何符号化手法 [9] を説明する．図 1 に，この手法のフレームワーク示す．はじめに，現フレームの点群と参照フレームの点群が入力として与えられる．参照フレームは，現フレームより時間的に前に取得されたフレームである．これらの点群に対してはあらかじめボクセルを用いた量子化が行なわれており，座標が整数値で表現されると想定する．動的に取得される点群はセンサの動きによってローカル座標系がフレーム間で一致しない場合があるため，グローバル動き補償 [11] によって位置合わせを行なう．その後，八分木符号化法 [12] に基づいて再帰的に現ノードの再分割と，それによって得られる部分空間（子ノード）の占有／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 次元空間の一部を表す立方体で表現され，境界を持つ．現ノードの再分割では，この境界で表される空間を再帰的に 8 つの部分空間に分割する．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るたびに深くなる．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．ユーザ指定の階層においては，ローカル動き補償 [13] によって現フレームにおける現ノード内の点群と参照フレームにおける探索ウィンドウ内の点群の位置合わせを行なう．現ノードが与えられた場合，位置合わせ後の参照フレームの点群から現ノードの境界内部に含まれる部分点群を取得する．そして，この部分点群から，占有予測 [14] に基づいてコンテキストを生成する．現ノードにおける占有／非占有を表す 8 ビット符号は，このコンテキストに基づき算術符号化される [15]．最後に，この符号化によって得られたビットストリームを出力する．

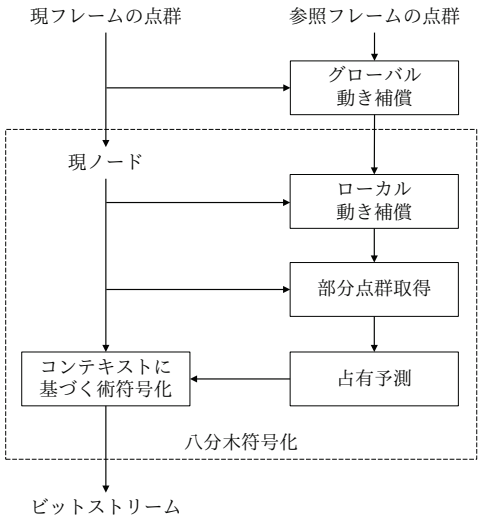


図 1　フレーム間幾何符号化手法のフレームワーク

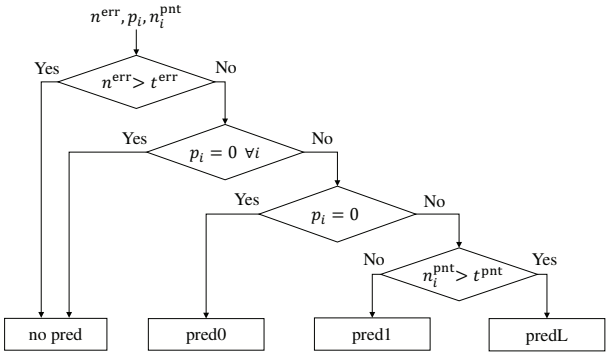


図 2　予測器の生成フロー

本稿で焦点を当てる占有予測についてより詳細に説明する．この処理は，参照フレームの部分点群から現ノードの 8 つの子ノードの占有状態を予測する．部分点群が与えられた場合，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 つの子ノードの占有／非占有の予測 $p_i \in \{0, 1\}$ および点数の予測 $n_i^{\mathrm{pnt}} \in \mathbb{N}^0$ $(i = 1, \ldots, 8)$ 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 4 種類からなる予測器を生成する．図 2 に予測器の生成フローを示す．no pred は，不適切な予測を防止するための，予測不可を表す予測器である．この予測器は，現階層の直前の階層において，現ノードの親ノードが占有予測を誤った回数 $n^{\mathrm{err}} \in \{0, \ldots, 8\}$ が閾値 t^{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．まとめると，この予測器は次の 2 つの条件のどちらかを満たす場合に生成される．

$$n^{\mathrm{err}} > t^{\mathrm{err}}. \tag{1}$$

$$p_i = 0\ \ \forall i \in \{1, \ldots, 8\}. \tag{2}$$

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，次式の条件を満たす場合に生成される．

$$n_i^{\mathrm{pnt}} > t^{\mathrm{pnt}}. \tag{3}$$

ここで t^{pnt} は固定の閾値である．このようにして，点数の閾値処理によって分類される pred1 と predL を用いることにより，pred1 のみを用いる場合と比べて占有予測の効果を向上させることができる．最後に，占有予測処理は現ノードにおける予測器の集合を出力する．

3. 参照フレームにおける点拡張

前節で述べたフレーム間幾何符号化手法においては，動き補償を行なった後であっても，フレーム間の幾何的誤差によって圧縮性能が制限されるという問題がある．そのため，筆者らは参照フレームの点（参照点）を拡張することにより，この幾何的誤差に対処することを試みている [10]．本節では，この点拡張手法について簡潔に説明する．

図 3 に，点拡張手法のフレームワークを示す．はじめに，参照点が与えられる．点群を計測した測域センサの光線方向による点の分類によって各点が属するクラスを求め，K 通りのオフ

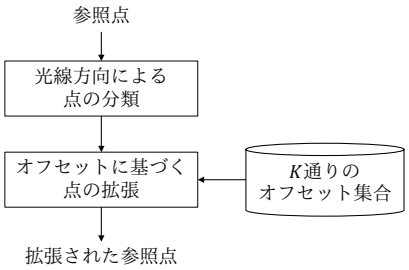


図 3　点拡張手法のフレームワーク

セット集合から対応するオフセット集合を取得する．そして，オフセットに基づく点の拡張を行なうことにより，拡張された参照点を得る．

光線方向による点の分類は，参照フレームの点を，測域センサの光線方向を表す K 種類のクラスに分類する．はじめに，3 次元直交座標系における点の座標を，3 次元極座標系における座標に変換する．これによって得られた点の方位角 θ および極角 ϕ を，等間隔の量子化範囲を用いて量子化する．そして，量子化された方位角および極角の値を組み合わせることにより，点の属するクラスのインデックス $k \in \{1, \ldots, K\}$ を得る．

オフセットに基づく点の拡張は，参照点ごとにオフセット集合を用いて新たに点を生成する．各オフセットは参照点の座標を基準として生成される点の相対位置を表すため，参照点の座標にオフセットの値を加算することにより，新たな点の座標を得る．ここでは，k 番目のクラスに属する参照点に対しては，k 番目のクラスに対応するオフセット集合を用いる．

オフセット集合は，訓練用の点群データ（訓練点群）の符号量を削減するオフセットで構成される．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後の参照フレームの点群の組の集合である．オフセット集合を構築する際には，訓練点群における参照フレームの各点から現フレームの最近傍点を探索し，それらの座標の差分を表すベクトルを収集する．これらはフレーム間の幾何的誤差を補正するオフセットの候補とみなすことができる．そして，収集されたオフセット候補の集合から，訓練点群の符号量を削減するものを選択する．ここでは，オフセット候補を用いる前後で符号量が削減されるかを検証し，削減された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．なお，オフセット集合はクラスごとに独立に構築する．

4. 提 案 手 法

本稿では，式 (3) の閾値処理に対して，現在の八分木階層に応じて適応的に閾値 t^{pnt} を決定する手法を提案する．前節で述べた点拡張手法では，新たに点を生成することにより，参照フレームの点群全体にわたって点密度が増加する．そのため，点拡張を行なわない場合に比べて，ノード内の点数 n_i^{pnt} が大きくなる．固定の閾値を用いる場合，点密度の変化への対応が困難であるため，予測器 pred1 と predL の生成頻度に偏りが生じる可能性がある．生成頻度がどちらか一方に偏る場合，pred1 と predL の 2 つの予測器を用いることによる圧縮性能の改善効果が減少する．この問題に対処するために，現在の階層における閾値を，これまで処理した階層におけるノード内の点数を用いて適応的に決定する．

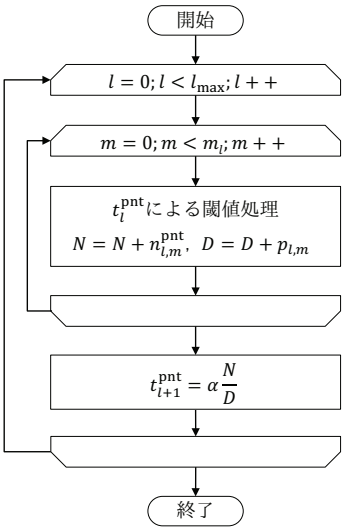


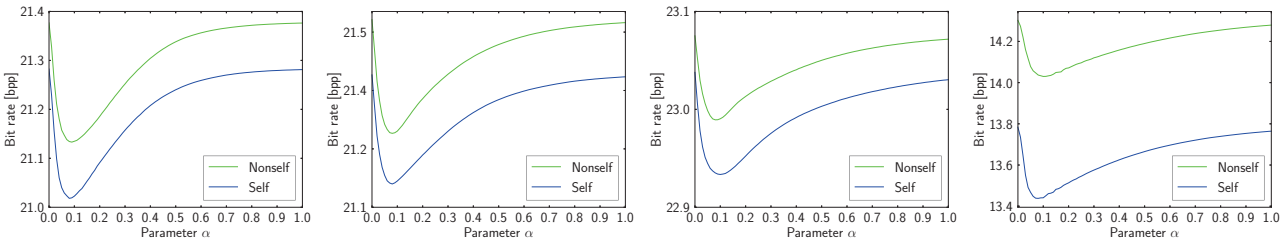
図 4　提案手法のフロー

l 番目の階層のノードから得られる m 番目の占有／非占有の予測を $p_{l,m} \in \{0, 1\}$，点数の予測を $n_{l,m}^{\mathrm{pnt}}$ と表記する．この時，$l+1$ 番目の階層で使用する閾値 t_{l+1}^{pnt} を，次式で決定する．

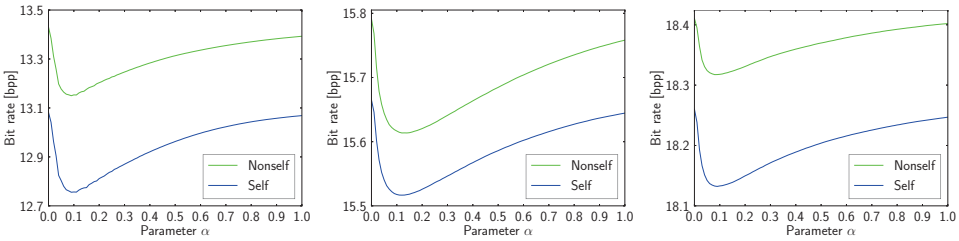
$$t_{l+1}^{\mathrm{pnt}} = \alpha \frac{\sum_{i \in L_l} \sum_{j \in M_i} n_{i,j}^{\mathrm{pnt}}}{\sum_{i \in L_l} \sum_{j \in M_i} p_{i,j}}, \tag{4}$$

ここで $L_l = \{0, \ldots, l\}$ は l 番目の階層までの階層のインデックス集合，$M_i = \{0, \ldots, m_i\}$ は i 番目の階層で得られる予測のインデックス集合，α は調整可能なパラメータである．α を除けば，この式は点数の累積移動平均を表す．この値は直前の階層における点数の平均に比べて平滑化されているため，これまでの階層から得られる情報を用いた閾値の決定により適している．図 4 に，占有予測処理における提案手法のフローを示す．この図では，最大の八分木階層を $l_{\max}$ とする．八分木符号化では，0 番目の階層から開始し，階層を深めつつ反復的に現階層のノードを処理する．各階層では，ノードの符号化に利用する予測器を生成するために，現階層の直前の階層で計算された閾値を用いて，式 (3) の閾値処理を行なう．同時に，現階層のノードから得られる $n_{l,m}^{\mathrm{pnt}}$ および $p_{l,m}$ の累積和 N および D を計算する．そして，現階層の全てのノードを処理した後に，N を D で除算し，α を乗算した値を閾値 t_{l+1}^{pnt} とする．この閾値は，現階層の次の階層の閾値処理で使用する．なお，0 番目の階層で使用する閾値 t_0^{pnt} は，参照フレームの点群の点数に α を乗算した値にあらかじめ設定しておく．

ここで，パラメータ α を調整する方法を説明する．この α は，前節で述べた訓練点群によるオフセット集合の構築時に調整する．この調整では，α を変化させながら訓練点群を符号化し，符号量が最小化となる α を求める．本稿では，変化させる α として [0, 1] の範囲から 0.01 間隔でサンプリングした値を用いる．オフセット集合を用いた点拡張によって符号量が最小となる α が変化する可能性があるため，この調整はオフセット集



(a) Ford01　(b) Ford02　(c) Ford03　(d) Approach



(e) Exit　(f) Join　(g) Bends

図 5　パラメータ α と符号量の関係

合の構築の前後で実行される．すなわち，オフセット集合を構築する前に，点拡張を行わない設定で α の初期値（α_{init}）を決定する．その後，この α_{init} を用いてオフセット集合を構築する．そして，構築されたオフセット集合を用いて点拡張を実行する設定で，最終的な値（α_{final}）を決定する．以降，点拡張を伴う符号化では，この α_{final} を使用する．

5. 評価実験

提案手法の有効性を評価するために，従来の無損失幾何符号化手法との実験的な比較を行なう．

5.1 実験設定

評価のために，MPEG で利用されている LiDAR を用いて動的に取得した 7 つの点群シーケンスである “ford_01_q1mm"（Ford01），“ford_02_q1mm"（Ford02），“ford_03_q1mm"（Ford03），“qnxadas-junction-approach"（Approach），“qnxadas-junction-exit"（Exit），“qnxadas-motorway-join"（Join），“qnxadas-navigating-bends"（Bends）を使用する．各シーケンスに含まれるフレームの数は，Ford01 から Ford03 が 1,500，Approach および Exit が 74，Join が 811，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あらかじめ量子化されている．点の拡張手法におけるオフセット集合の構築のために，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組を使用する．ただし，Approach および Exit には 74 フレームしか含まれないため，73 組の点群のみを使用する．点の分類については，文献 [10] と同じ設定を使用する．

5.2 圧縮性能

はじめに，提案手法（Ours）と従来手法の無損失圧縮性能を比較する．従来手法として，八分木符号化法（Octree），G-PCC [8]，フレーム間幾何符号化手法（Inter） [9]，筆者らの従来手法 [10] を用いる．筆者らの従来手法は本稿の提案手法を基準として閾値を固定したものとみなせるため，“Ours（Fixed）"と表記する．

表 1　提案手法と従来手法の平均符号量 [bpp] の比較

	Octree	G-PCC	Inter	Ours（Fixed）	Ours
Ford01	55.84	23.48	22.23	21.76	**21.67**
Ford02	55.01	23.15	21.87	21.35	**21.26**
Ford03	56.87	24.00	23.44	23.05	**22.99**
Approach	45.38	17.84	14.97	14.08	**14.03**
Exit	44.35	16.86	14.08	13.17	**13.15**
Join	45.11	17.10	16.22	15.80	**15.74**
Bends	47.77	19.56	18.99	18.55	**18.47**
Average	50.05	20.29	18.83	18.25	**18.19**

表 1 に，bits per point（bpp）単位で表される平均符号量をシーケンスごとに示す．Octree は全てのシーケンスにおいて符号量が最大となり，平均で 50.05 bpp である．G-PCC は，Octree に対して空間的相関に基づくエントロピー符号化等の多くのモジュールを導入することによって，圧縮性能を大きく改善する．Inter は，G-PCC に対して動き補償と占有予測を導入することによって，圧縮性能をさらに改善する．Ours（Fixed）は，フレーム間の幾何的誤差を解消するための点拡張を利用することにより，Inter より優れた圧縮性能を達成する．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し，その平均は 18.19 bpp である．これは，提案した適応的閾値決定により，予測器生成の偏りが解消されたためである．

5.3 パラメータ α の影響

次に，提案手法におけるパラメータ α の符号量への影響を検証する．本実験においては，式 (4) のパラメータ α を [0, 1] の範囲で 0.01 間隔で変化させた場合の，訓練点群の符号量を調査する．ここでは，オフセットの構築に使用したシーケンスと同一のシーケンスを符号化する場合（Self）と，オフセットの構築に使用したシーケンスと異なるシーケンスを符号化する場合（Nonself）の両方の関係を調査する．

図 5 に，全てのシーケンスに対する結果を示す．この図より，

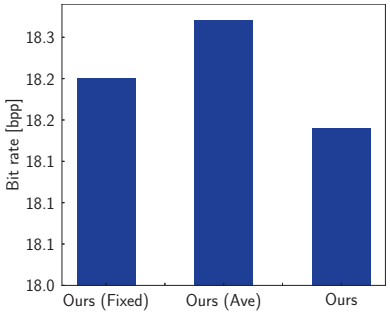


図 6　閾値決定方法の比較

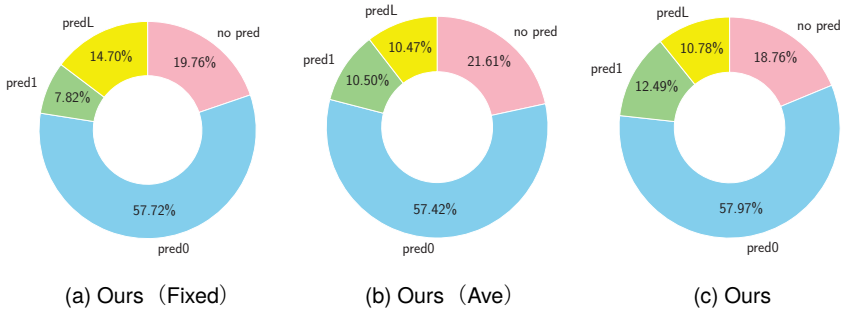


図 7　占有予測における予測器の生成割合

Self と Nonself の両方において，α の変化が符号量に影響を与えることがわかる．全てのシーケンスにおいて，Self の場合の方が Nonself の場合より符号量が小さくなる．これは，シーケンス内の幾何的誤差と符号化に使用されたオフセットの差異が少ないために，圧縮性能がより大きく改善するためである．また，全てのシーケンスにおいて，α が 0.1 程度の場合に符号量が最小になることがわかる．これは，予測器 pred1 と predL を分ける閾値が，現階層の直前の階層までの点数の累積移動平均の 0.1 倍程度の場合に，最良の割合で予測器を生成できることを示している．

5.4　累積移動平均の効果

最後に，提案手法における累積移動平均の効果を検証する．本実験では，式 (4) において現階層の直前の階層における点数の平均を用いる，提案手法の変形との比較を行なう．以下では，この変形を “Ours（Ave）"と表記する．さらに，ベースラインとして Ours（Fixed）との比較も行なう．

図 6 に，7 つのシーケンスにおける Ours (Fixed)，Ours (Ave)，Ours の平均符号量を示す．Ours（Fixed）をベースラインとする場合，Ours（Ave）は符号量が増加する一方で，Ours は符号量が減少する．これにより，累積移動平均を用いることによって符号量の削減効果が得られることがわかる．

この結果をより詳細に分析するために，図 7 に符号化時の予測器の生成割合を示す．Ours（Fixed）では，pred1 と predL の生成割合に関して，predL への偏りが生じていることがわかる．Ours（Ave）と Ours では，どちらもこの偏りが解消されている．提案した適応的閾値決定においては，符号量が最小になるようにパラメータ α が調整されるため，結果的に最適な pred1 と predL の生成割合が得られる．したがって，符号量を削減するためには，pred1 と predL の偏りを解消することが効果的であることが確認できる．この結果は，予測器の偏りが大きいほど，どちらか 1 つの予測器のみを用いる場合と同等の圧縮性能へ近づくのに対して，予測器の偏りが解消するにつれて，2 つの予測器を用いることによる圧縮性能の改善効果が大きくなることを示唆している．また，Ours（Fixed）と比べて，Ours（Ave）は no pred の割合が増加する一方で，Ours は no pred の割合が減少することがわかる．no pred の割合が大きいほど占有予測の効果が減少するため，符号量の削減効果が得られなくなる．したがって，Ours（Fixed）に比べて Ours（Ave）の符号量が大きい原因は，no pred の割合が大きいためであると考えられる．

表 2　平均および累積移動平均を用いた場合のパラメータ α の比較．最終行は全てのシーケンスにわたる α の分散を表す

	Ours（Ave）			Ours		
	α_{init}	α_{final}	$\lvert\Delta\rvert$	α_{init}	α_{final}	$\lvert\Delta\rvert$
Ford01	0.25	0.48	0.23	0.05	0.09	0.04
Ford02	0.40	0.25	0.15	0.06	0.08	0.02
Ford03	0.38	0.29	0.09	0.07	0.10	0.03
Approach	0.57	0.29	0.28	0.15	0.08	0.07
Exit	0.65	0.45	0.20	0.16	0.09	0.07
Join	0.47	0.40	0.07	0.10	0.12	0.02
Bends	0.52	0.35	0.17	0.10	0.09	0.01
Variance	0.017	0.0076	-	0.0018	0.00019	-

Ours（Ave）において no pred の割合が大きくなる原因として，最適な α の不安定性が考えられる．表 2 に，Ours（Ave）および Ours におけるパラメータ α_{init}，α_{final}，およびそれらの差の絶対値 $|\Delta|$ をシーケンスごとにまとめる．また，最終行に全てのシーケンスにわたる α_{init} および α_{final} の分散を示す．この表より，Ours（Ave）は Ours に比べて $|\Delta|$ が大きく，さらに α_{init} および α_{final} の分散が大きいことがわかる．すなわち，Ours（Ave）は点拡張に伴う点密度の変化に対して頑健でないことがわかる．したがって，この手法でオフセット集合を構築する場合，符号量を削減するオフセットを適切に選択できない恐れがある．結果として，no pred となる割合が高くなったと考えられる．それに対して，Ours では点密度の変化に対して頑健であるため，より適切にオフセットを選択することができる．結果として no pred の割合の低減と，pred1 と predL の割合の最適化を両立できるために，最も符号量を削減したと考えられる．

6.　ま　と　め

本稿では，点拡張を導入したフレーム間幾何符号化において，点密度の変化に対して頑健に予測器生成の偏りを解消するための閾値決定手法を提案した．提案手法では，八分木符号化法における現階層の直前の階層までのノード内の点数の累積移動平均を用いて，現階層の閾値を決定した．ここでは，訓練用の点群データの符号量を最小化するパラメータ α を探索し，閾値の決定に利用した．評価実験では，提案手法によって圧縮性能が改善されることを確認するとともに，パラメータ α の影響と累積移動平均の効果を示した．今後は，点群の粗密によらず圧縮性能を改善可能な閾値決定について検討する．

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] MPEG 3DG, “Call for proposals for point cloud compression v2”, ISO/IEC JTC 1/SC 29/WG 11 N16763, 2017.

[7] 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.

[8] 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.

[9] S. Lasserre. “Exploratory model for inter-prediction in G-PCC”, ISO/IEC JTC1/SC29/WG11 N18096, 2018.

[10] 松﨑康平，河村圭, “時間的相関を用いた疎な点群の圧縮のための点拡張,” 第 20 回情報科学技術フォーラム（FIT2021）, 2021.

[11] S. Lasserre and D. Flynn. “Global motion compensation for point cloud compression in TM3”, ISO/IEC JTC1/SC29/WG11 MPEG2018/m44751, 2018.

[12] D. Meagher, “Geometric modeling using octree encoding,” Computer Graphics and Image Processing (CGIP), vol.19, no.2, pp.129–147, 1982.

[13] S. Lasserre and D. Flynn. “On motion compensation for geometry coding in TM3”, ISO/IEC JTC1/SC29/WG11 MPEG2018/m42521, 2018.

[14] 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.

[15] 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.