I-079

3 次元ブロックの Run-length Coding による Time-Varying Mesh の圧縮

Compression of Time-Varying Mesh Using Run-length Coding of 3D Block Sequences

韓 昇龍† Seung-Ryong Han　山崎 俊彦‡ Toshihiko Yamasaki　相澤 清晴‡ Kiyoharu Aizawa

1．はじめに

近年，計算機の性能の向上やセンサー技術の発達に伴い，複数の同期カメラから，動く 3 次元被写体を再構成する Time-Varying Mesh (以下，TVM と略す) の取得・生成に関する研究が行われている [1]．TVM は，現実空間での被写体を仮想空間上で容易に再構成できることから，無形文化財の保存やインタラクティブな放送の応用に期待されている．

TVM は 3 次元モデルのシーケンスとして表現される．本稿で用いる TVM は 1 つのモデル当たり 5 万個以上の頂点で構成されており，VRML 形式で表すと 5MB 以上のデータ容量が必要である．そのため，TVM の圧縮は重要な研究課題の 1 つである．

一般的に， 3 次元オブジェクトの形状を現す幾何情報はモデル表面にのみ分布する．そこで，我々はこのような 3 次元モデルの空間的な冗長度を削除し，圧縮を図る手法を提案する．本稿では，空間的な冗長度の削除のため 3 次元モデルの頂点情報を 3 次元ブロック単位で量子化し，ビットストリームに変換する．変換されたストリームをランレングス符号化で圧縮する．TVM シーケンス 49 フレームに対して実験を行った結果，オリジナルでは 96 bit per vertex (bpv) のデータを 2.6 bpv まで圧縮できた．

本稿では，まず，次章では幾何情報のビットストリームに変換し，ランレングス符号化手法について説明する．第 3 章では連続する TVM のシーケンスへの適用について説明し，第 4 章では，提案手法による TVM の圧縮実験結果を示す．

2．ブロック化による TVM の幾何情報圧縮

TVM の幾何情報の圧縮のため，t 番目の 3 次元モデル $\mathbf{M}(t)$を考える．3 次元モデル $\mathbf{M}(t)$は表面モデルであるため，モデルの表面上に頂点群が分布し，モデルの中には形状情報を持っていない．このような空間的な冗長度を削除するため，頂点情報を量子化し格子位置を表すビットストリームに変換し，ランレングス符号化を用いて符号化を行う．

符号化を行うモデル $\mathbf{M}(t)$のバウンディングボックスを求める．この際，求めたバウンディングボックスの縦，横，奥行の長さの中で最も長い辺の長さ $l(t)$を一辺とする正六面体 $B(t)$を作る．その後，$l(t)$を x, y, z 方向で M 個に均等に切ることで，P 個 (P=M×M×M) のキュービックブロック群 $C_p(t)$ (p=1, 2, …, P)が得られる．図 1 の左と中央は $\mathbf{M}(t)$と $\mathbf{M}(t+1)$のブロック化結果を示す．3 次元モデルは表面モデルであるため，表面を含んだブロックの数は少ない．本稿ではこのようなブロックを表面ブロックと呼び，$S_k(t)$ (k=1, 2, …, K)で表すこととする．また，表面を含んでいないブロックを $R_q(t)$ (q=1, 2, …, Q)とする．すなわち，

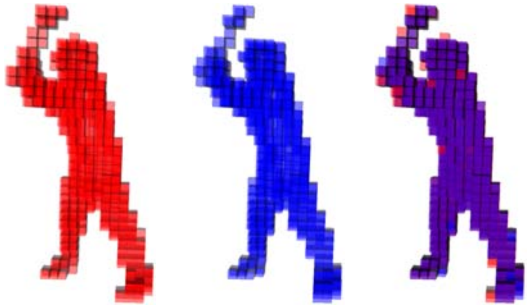
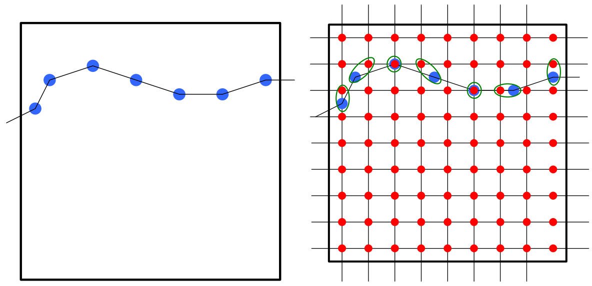


図 1: ブロック化の結果



(a)　(b)

図 2: (a)表面ブロック $S_k(t)$の頂点群 $\mathbf{V}_k(t)$，(b) 量子化代表点と頂点群の対応付け結果

P=K+Q である．表面ブロック $S_k(t)$には，$\mathbf{M}(t)$の頂点群が含まれている．ここで，$S_k(t)$に含まれる頂点群を $\mathbf{V}_k(t)$で表す．図 2 にその例を示す．図 2 (a)は $S_k(t)$を 2 次元空間に投影したものである．青い点が $S_k(t)$に含まれた頂点群 $\mathbf{V}_k(t)$を示す．図で分かるように，頂点群はブロックの中で偏って分布している．すなわち，頂点群が存在するところは隣接したところで他の頂点群が存在する確率が高く，頂点群が分布していないところは，周りに頂点が存在する確率が少ない．このような性質から，ランレングス符号化を用いて $S_k(t)$内の頂点座標の符号化を行う．

頂点座標の符号化はランレングス符号化を用いる．そのためには $S_k(t)$の頂点群 $\mathbf{V}_k(t)$をビットストリームに変換する必要がある．$\mathbf{V}_k(t)$のビットストリーム化を行うため，まず $S_k(t)$の中に等間隔量子化代表点を設ける．図 2 右の赤い点が量子化代表点を示す．次に，$\mathbf{V}_k(t)$とユークリッド距離が最も近い量子化代表点を $S_k(t)$の中から探す．図 2 右には緑の楕円で囲んだ頂点ペアが，最も近い頂点同士である．量子化代表点の間隔を小さくし，対応付けられた量子化代表点を $\mathbf{V}_k(t)$の値の代わりに用いることができる．しかし，量子化代表点の数が大きくなるため，符号化効率は低下する．次に，対応付けられた量子化代表点には”1”を，対応付けられていない量子化代表点には”0”を割り当て，ラスタ走査順に量子化代表点を走査しながら，ビットストリームを生成する．

†東京大学大学院新領域創成科学研究科
‡東京大学大学院情報理工学系研究科

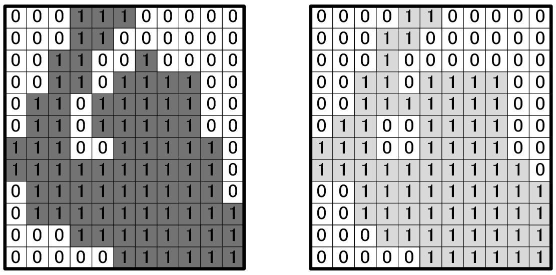


図 3: $\mathbf{M}(t)$と $\mathbf{M}(t+1)$のブロック群のビットストリーム化の例

３次元モデルは表面モデルであるため，各々の表面ブロック $S_k(t)$のビットストリームは，モデル頂点と対応付けされた量子化代表点の数は少なく，対応付けされてない量子化代表点が殆どである．すなわち長い”0”ランが発生する確率が高い．このような性質から $S_k(t)$のビットストリームはランレングス符号化を用いて効率よく圧縮をすることができる．

３．時間的冗長度を用いたブロック情報の圧縮

第 2 章では，$S_k(t)$内の頂点群 $\mathbf{V}_k(t)$の符号化手法について説明した．次は，ブロック群 $S_k(t)$の符号化を行う．$S_k(t)$の符号化もランレングス符号化手法を用いる．前章のブロック化によって $\mathbf{M}(t)$からブロック群 $C_p(t)$が得られる．ここで，$C_p(t)$は $S_k(t)$と $R_q(t)$で構成されている．ここで，$S_k(t)$には”1”を $R_q(t)$には”0”を割り当て，ラスタースキャン順に $M\times M\times M$ のブロック群 $C_p(t)$を走査しながら，ビットストリーム化を行う．この様子を図 3 の左に示す.

ランレングス符号化は長いランを発生させることで符号化の効率が上がる．そこで，本稿では TVM のモデル間に存在する時間的冗長度を用いる．一般的に連続するフレーム間はお互いに類似している．図 2 の右にこの様子を示す.すなわち，ブロック化後の $\mathbf{M}(t)$と $\mathbf{M}(t+1)$の表面ブロック $S_k(t)$と $S_k(t+1)$を比較してみると，動きがあるところに，$S_k(t)$と $S_k(t+1)$の違いが現れる．図 3 で分かるように，モデル間の動きが存在するところで，”0”が”1”に，”1”が”0”に変わることが確認できる．ここで，$C_p(t)$のビットストリームと $C_p(t+1)$のビットストリームの排他的論理輪を取ることで，変化したブロックだけが”1”に，変化のないブロックはすべて”0”になる．すなわち，前後するモデルの情報を用い，長いランを発生させることで，ブロック情報の圧縮率を上げることができる．

4．実験

圧縮実験のため，富山ら[1]の TVM シーケンスを用いた.シーケンスの各 3 次元モデルは 5 万個以上の頂点構成されている．まず，M=30 と設定し，3 次元モデル $\mathbf{M}(t)$ (t=1, 2, ..., 49)のブロック化を行った．また，$\mathbf{M}(t)$の表面ブロック $S_k(t)$に $10\times10\times10$ の量子化代表点を設けた．その後，ビットストリーム化を行い，修正ハフマン符号化手法を用いて圧縮符号化を行った．図 4 (a)に，圧縮結果を示す．圧縮前の頂点情報は 96bpv の容量を必要であるのに対し，提案手法では，2.6~3.7bpv まで圧縮することができた．また，圧縮による誤差 D を確かめるため，符号化前後の i 番目の頂点を $\mathbf{v}_i=(x_i, y_i, z_i)$と $\mathbf{v}_i^*=(x_i^*, y_i^*, z_i^*)$で頂点間の平均誤差を求める．平均誤差は次式で求められる．

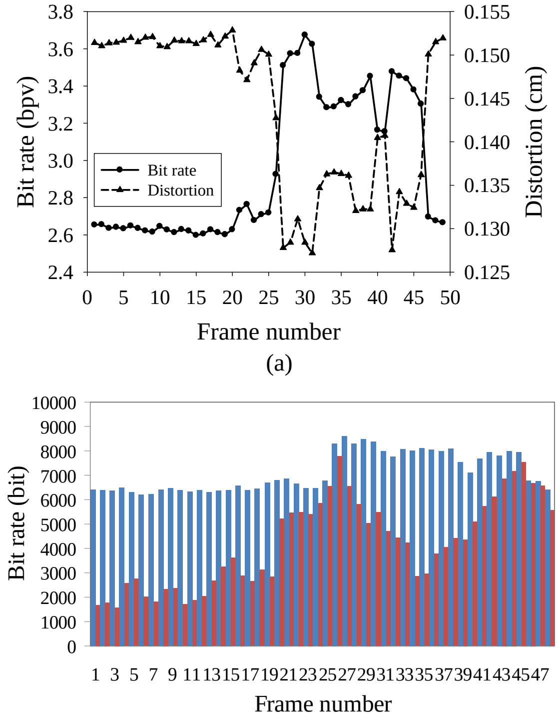


図 4: (a) 幾何情報のビットレートと誤差，(b) ブロック群の圧縮結果

$$D=\frac{1}{3N(t)}\sum_{i=1}^{N(t)}\left(\left|x_i-x_i^*\right|+\left|y_i-y_i^*\right|+\left|z_i-z_i^*\right|\right)$$

ここで $N(t)$は $\mathbf{M}(t)$の頂点数を示す．図 4 (a)には，上の式で求めた誤差を示す．誤差の範囲は 0.12~0.15 cm であった．TVM のシーケンスの 3 次元モデルのダイナミックレンジは約 200 cm 前後であるため，誤差はほぼ無視するできる.

次に，$\mathbf{M}(t)$のブロック群 $C_p(t)$の符号化実験を行った．図 4 (b)にその結果を示す．青いバーが TVM のモデル間の時間的な冗長度を用いる前のビットレートで，赤いバーが時間的な冗長度を用いた後のビットレートである．明らかであるように，時間的な冗長度を用いることで，$\mathbf{M}(t)$のブロック群に対して更なる圧縮が実現できた．

５．結論

我々は TVM の幾何情報の圧縮のため，TVM に存在する空間的な冗長度と時間的な冗長度を用いて，幾何情報をビットストリームに変換した後，ランレングス符号化で圧縮する手法を提案した．提案手法により，頂点の元データサイズに比べ，2.7%~3.8%まで圧縮できた．

参考文献

1. K. Tomiyama *et al.* “A Dynamic 3D Object Generating Algorithm from Multiple Viewpoints Using the Volume Intersection and Stereo Matching Methods”, ITE vol. 58, no 6, pp. 796-806, June 2004.