J-29

コスト最小化に基づく 三次元形状情報削減アルゴリズムに関する検討

A Study on 3D Shape Compression Algorithms based on Cost Minimization.

中村 真也† Shinya Nakamura　　**甲藤 二郎**† Jiro Katto

1. まえがき

近年、多くの分野でコンピュータグラフィックスが用いられており、とりわけ３次元形状の活用が非常に盛んに行われている。しかし、３次元形状を構成する情報としては、非常に膨大な形状情報を必要とし、計測点群データ等をそのまま実際に利用するには情報量の多さから困難が生じる。そこで、本稿では、形状をできるだけ変化させずに情報量を効率よく削減する手法を提案する。

2. 点群削減アルゴリズム

３次元モデルの幾何形状特徴を保存し、かつ情報量の削減を行う手法としては、各点およびパッチについてエネルギー関数を定義して点群削減を行う手法[1]や、点の統合処理前後の差分をもとに点群の削減判定を行う手法[2]などが知られている。

今回、我々はより簡便な点群削減アルゴリズムの実現を目的として、パッチ内の各点に対していくつかのコスト関数を定義し、それらの最小化に従って点群削減を行う手法を提案する。なお、以下の点群統合処理では、削減の対象となる２点を、その中点の位置に移動、統合することを前提としている。

手法１ パッチ間角度の最小化[3]

この手法では、各点が絡んでいる線分をはさんでいる２つのパッチのなす角度を合計したものをコストとし、閾値を与えた後で各線分に対する重要度を判定する。そして計算により求められた重要度の低い点から削除を実行していく(図１参照)。

具体的なコストは次式で定義する。

$$C_0(x) = \frac{1}{\sum_{k=1}^{n} \theta_k}$$

ただし、x は隣接する２点 (L,R) を表すインデクス、M は２点を統合した点、n は x を含む線分の数、$\theta_{(1\sim n)}$ は L および R をひとつの頂点とするパッチ間のなす角度、である。この値が小さいほど、点 L および R の周囲のパッチが平坦であること、すなわち２点を統合しても形状が大幅に変化しないこと、が期待される。

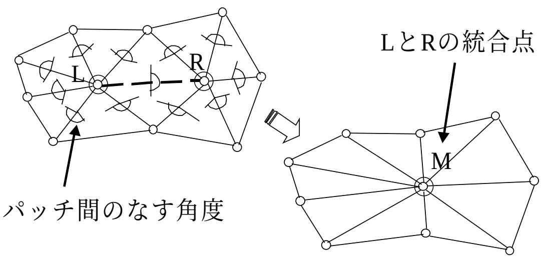

図１　パッチ間角度の最小化

ただし、点群削除を行う際、一定の閾値を用いて計算を続けると、削除のたびに形状への小さな変化が積み重なり、一部分の形状が大幅に変わってしまう可能性がある。このため、点の削除を実行する度に再計算を行い、重要度判定の閾値を変化させていくことで、特定の点群付近で繰り返し削除の処理が行われるのを防止する。

手法２ 点間距離の最小化

この手法では、３次元形状モデル内の隣接する２点において、その２点間距離、および２点と隣接する点との距離の２乗和

$$C_1(x) = d_{L,R}{}^2 + \sum_i d_{L,i}{}^2 + \sum_j d_{R,j}{}^2$$

をコストと定義する。ただし、x は隣接する２点 (L,R) を表すインデクス、i は点 L の隣接点の集合 (点 R を除く)、j は点 R の隣接点の集合 (点 L を除く)、とする。この値が小さいほど、点 x の周囲の点密度が密であること、すなわちオーバーサンプリングされた点群を削減できること、が期待される。

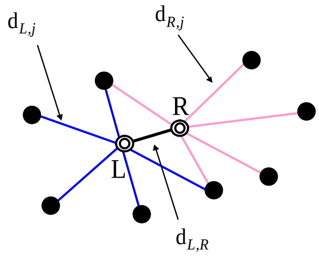

図２　点間距離の最小化

手法３ 点群削減前後の面積変化量の最小化

隣接する２点 L、R の周囲の三角形パッチ（図 3a）の総面積、および２点を統合した点 M の周囲の三角形パッチ（図 3b）の総面積の差

$$C_2(x) = \left| \sum_k S_{L,R}{}^{(k)} - \sum_k S_M{}^{(k)} \right|$$

† 早稲田大学大学院　理工学研究科
Graduate School of Science and Engineering, Waseda University

を、コストと定義する。（ただし、x は上記と同じく隣接する２点 (L,R) を表すインデクスである。この値が小さいほど面積の変化量が小さく、点の削減による形状の変化が小さくなるものと期待される。

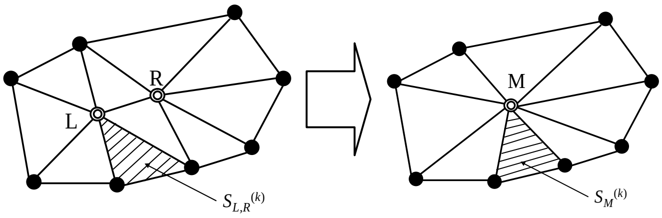

図 3a　点統合前　　図 3b　点統合後

図 3　面積変化量の最小化

点群削減アルゴリズム

各点において以上のコストのいずれか、あるいは組み合わせを計算し、その値が最小となるものから点群削除を行う。現在は、１回のループあたり１点の削除を行っている。すなわち、一度の削除で点が１個、線分が３本、パッチが２個、削除される。

3．実験結果と考察

今回提案したコストを組み合わせ、形状の表面積変化率を評価尺度として得られた結果を表１に示す。ここで評価に用いた３次元形状は、点群数 3874、面数 7744 の馬の形状である。具体的なコストの計算は

$$Cost = \alpha \cdot C_0(x) + \beta \cdot C_1(x) + \gamma \cdot C_2(x)$$

とした。ただし、今回の実験では、(α，β，γ) = 1 or 0 としている。

表 1　形状の表面積変化率

削除点群数	角度①	距離②	面積③	①②	①③	②③	①②③
500	12.6	0.94	0.09	1.48	4.69	0.49	0.98
1000	51.3	1.61	0.79	4.89	12.1	1.67	4.18
1500	80.0	2.39	1.52	5.15	16.9	2.35	5.74

上記の結果より、パッチ間角度だけをコストとして設定した場合 (①) には、削減点群数を増やすにつれて、オブジェクトの表面積変化が著しく大きくなることが分かる。その影響で、コスト①を用いた複合コスト方式（①＋②、①＋③）においても、他方式よりも大きな面積変化が生じている。

また、点間距離をコストとして設定した場合 (②) は、複数の点間距離の差が小さいもの、つまり密度の高い点群を優先的に選択する。表面積変化率は③に劣るものの、主観評価としては、オブジェクトの表面を滑らかに保ちながら点群を削減できていることを確認した。

一方、表面積変化をコストとして設定した場合 (③) は、今回の評価尺度を最小化する効果を持つため、必然的に点群削減後の表面積変化率も、他のコスト定義と比較して小さくなっている。しかしながら、主観評価としては、オブジェクトを滑らかに削減する効果として、②よりは効果は薄かった。

これらの結果より、②および③を合わせたコスト最小化

$$Cost = \beta \cdot C_1(x) + \gamma \cdot C_2(x)$$

による点群削減処理が最適といえることがわかった。なお、予備実験により、(α，β，γ) の値を連続的に変化させてみたが、今回の実験に関する限りは、 (α，β，γ) = (0.0, 1.0, 1.0) の場合に最も良い結果が得られた。ただし、パラメータ値の最適化については今後も検討を続けていく予定である。

以上の点群削減処理を行った結果を図 4 に示す。

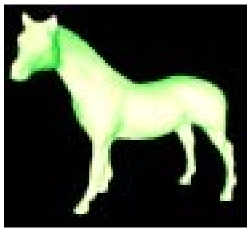

図 4a　元画像

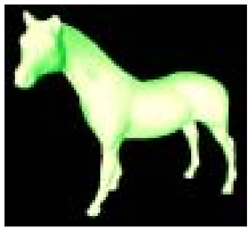

図 4b　500 点削減

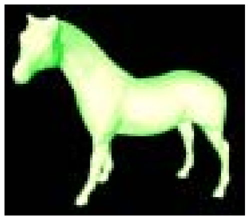

図 4c　1000 点削減

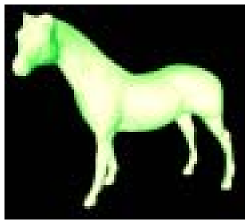

図 4d　1500 点削減

4．おわりに

複数のコスト評価に基づく三次元形状の点群削減・圧縮アルゴリズムについて検討を行った。その結果、点間の距離情報および統合処理前後の面積変化量のふたつのコストを用いて削除する点を決定することで、元形状からの形状の大幅な劣化を抑えることができた。

大量の情報量を削減する場合、削減点を選択する際に形状を保つための尺度が必要となるが、このようなコストを定義することでその変化率を大きく防ぐことができるといえる。今後は、さらなる変化率の抑制を図れるより高精度の手法の提案、および処理速度の向上について検討を進めていく予定である。

5．参考文献

[1] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle: “Mesh Optimization,” ACM SIGGRAPH 1993.
[2] 王、荒川、高坪、和田: “3 次元ポリゴンデータのリダクションアルゴリズム,” 2002 画像電子学会年次大会.
[3] 宇城由登: “三次元形状自動生成及び点群再構成に関する研究,” 早稲田大学大学院修士論文, 2001.