

大局的形狀を考慮する三角形メッシュの簡略化*

4 N-9

佐藤 秀則

北澤 仁志

小野澤 晃

NTT 入出力システム研究所

1 はじめに

CGの世界では、コンピュータの高速化、モデリング技術の進歩に伴い、複雑かつ巨大な、三角形メッシュの組合せによるデータが生成されるようになってきており、近年そのデータを扱うためのソフトウェア技術として、形状を可能な限り保ちつつ三角形メッシュの数を減らし、Level-of-Detail(LOD)形式に対応するためのアルゴリズムの研究開発が行われてきた[1,2]。

今回我々は、対象物体の大局的な形状を考慮する簡略化アルゴリズムを考案し、WebReality[3]に実装したので報告する。手法は、頂点削除の繰り返しを基本としており、その際の三角形メッシュの再構成時の形状変化が最小となるものから優先的に簡略化を進めていく。

2 エネルギー変化最小頂点削除による簡略化

2.1 アルゴリズム概要

本報告では、3次元形状測定器や3次元モデリングCADで作成された、三角形メッシュの組合せからなる多様体の3次元CGデータを簡略化対象としている。

我々は、Shroederらの方法[2]同様、頂点の削除を基にする手法を行なうこととした。[2]では、処理の高速性を重視するために、削除頂点の判定法や頂点削除後の付加枝の作成法が単純過ぎるという欠点がある。そこで、提案手法では、以下の手法を行なうこととした。

1. 頂点削除に伴う形状や面積変化をそれぞれエネルギー変化値とみなし、それらの線形和を総エネルギー変化値とする。
2. 指定エネルギー項が上限値を越えないものの中で総エネルギー変化値が最小となる頂点に対する形状変化を実行する。

2.2 フロー

以下にアルゴリズム全体のフローを示す。

* Mesh Simplification Algorithm Considering Global Feature
Hidenori Sato, Hitoshi Kitazawa, Akira Onozawa
NTT Integrated Information & Energy Systems Laboratories.

Step1. 初期エネルギー変化値計算

入力全頂点について計算

Step2. 頂点の総エネルギー変化値順のソート

指定エネルギー項が上限値以下のものを除く

Step3. エネルギー変化値最小頂点の削除

Step4. 枝付加による新規三角形メッシュ生成

Step5. エネルギー変化値再計算

削除頂点に影響を受けた頂点についてのみ計算

Step6. ソート結果変更

Step5. で変更を受けた頂点について変更

Step7. 終了判定

頂点数が目標数以下、またはソート結果がない場合、終了。

そうでない場合 Step3. へ

Step1. と Step5. では、処理の高速化のため、頂点削除後の再メッシュ化のパターンをあらかじめデータとして記憶しておく。削除頂点に対して、削除時の多角形が6角以下の場合には全ての中から、それ以外の場合には各頂点を共有するメッシュ数が平均化するような組合せの中から、最小エネルギー変化値をとるパターンをひとつだけ選択するようにする。また、本手法では、Hoppeらの手法[1]のように、形状変化をエネルギー変化の最適化問題としていないため計算コストが低く、大規模なデータについても実時間で解くことが可能となっている。

2.3 形状変化に伴うエネルギー変化値

図1に頂点を削除した時の頂点及び三角形メッシュ情報の変化の例を示す。本節では、この表記に従い形状変化に伴うエネルギー変化値の計算式を説明する。

- 合計面積変化(凹凸変化の面積的抑制):

$$E_{total} = |(\sum_j S(Q_j) - \sum_i S(P_i)) / \sum_i S(P_i)| \quad (1)$$

- 頂点の移動距離(凹凸変化の距離的抑制):

$$E_{move} = d/\sqrt{S} + \text{はみだしペナルティ値} \quad (2)$$

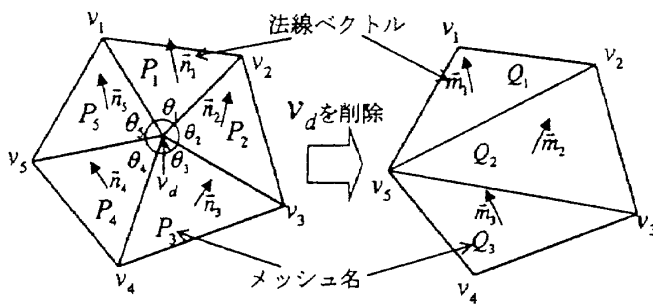


図1: 頂点削除による三角形メッシュの生成

- 法線ベクトルのずれ(面の方向変化の抑制):

$$E_{norm} = (1/M) \sum_j^M |\vec{m}_j - \sum(\theta_i \cdot \vec{n}_i) / \sum \theta_i|^2 \quad (3)$$

- 最大三角形メッシュ生成(巨大メッシュ生成の抑制):

$$E_{maxs} = \max(S(Q_j)) / \bar{S} \quad (4)$$

ここで、 $S(P)$ はメッシュ P の面積、 d は v_d から、法線方向 $\sum(\theta_i \cdot \vec{n}_i) / \sum \theta_i$ にある Q_j までの距離、 \bar{S} は入力メッシュの平均面積、をそれぞれ意味している。

この時の総エネルギー変化値 E_{total} は、 a, b, c, d を任意の実数の重み係数として、以下の式で求める。

- 総エネルギー変化値:

$$E_{total} = a \cdot E_{stotal} + b \cdot E_{vmove} + c \cdot E_{norm} + d \cdot E_{maxs} \quad (5)$$

上式を用い、重み係数や指定エネルギー項の上限値を制御することにより、簡略化の進行具合に応じて平面や曲面を主体に簡略化することが可能となり、対象物体の大局的な形状に応じた簡略化が可能となる。

3 実験結果

提案アルゴリズムを基にした LOD データ生成プログラムを開発し、WebReality 上に実装した [3]。開発プログラムを PC/AT 互換機 (Windows NT, Pentium II 266MHz 128MB) 上で適用した結果を図2に示す。実験では、パラメータとして、 E_{stotal} と E_{vmove} の上限値を、簡略化が進むに従い6通りに増加させた。このことにより、簡略化の前半では平面を主体に、後半で曲面まで含めて削減している。また、頂点削除時にできる多角形の角数が20角以上にならないように制御した。図から、95%まで頂点数を削減しても原型を留めていることがわかる。

さらに実設計データ等に対しても適用した結果、データの大局的な形状と式(5)の係数に関して、以下の傾向があることがわかった。

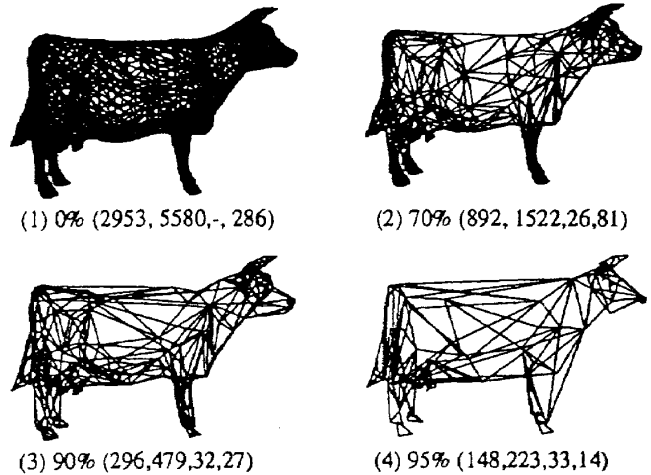


図2: 実験結果の簡略化率(頂点数、メッシュ数、実行時間(秒)、データ量(Kb))

- 平面が主の場合、係数 d のみを小さくし、巨大メッシュの生成を許す。
- 曲面が主の場合、係数 b のみを大きくし、頂点移動による形状の崩れを防ぐ。
- 境界がある場合、 E_{stotal} を内部頂点用と境界頂点用に分け、境界用の係数を大きくし、境界面の崩れを防ぐ。

4 まとめ

LOD データ生成プログラムを開発し、WebReality 上に実装した。プログラム内で用いる簡略化手法は、頂点の削除を基にし、メッシュの変形時の形状変化を総エネルギー変化値として表し、任意の指定エネルギー項が上限値を越えない範囲で、最小総エネルギー変化値をとる頂点から順次削除していく。実験の結果、パラメータ制御により、データの大局的な形状に合わせた効率的な簡略化が行なえることが判明した。今後は、色、テクスチャへの対応、非多様体への拡張等を行なう予定である。

謝辞

本研究を行なうにあたり、貴重な実設計データを提供して下さい、東陶機器(株)様に感謝致します。

参考文献

[1] Hugues Hoppe: "Progressive Meshes", SIGGRAPH '96, 99-108(1996).
 [2] William J. Schroeder: "A Topology Modifying Progressive Decimation Algorithm", Proc. of Visualization '97, 205-212(1997).
 [3] 北澤, 小野澤, 佐藤: "WebReality: ネットワークベース仮想空間レイアウトシステム", 情報第57回全国大会, 4N-08(1998)