グラフィクスとCAD 89－7
（1998. 2. 27）

特徴抽出と稜線操作によるポリゴンメッシュの簡単化

早野 勝之† 松岡 司 植田 健治
株式会社リコー ソフトウェア研究所
†mhayano@src.ricoh.co.jp

ポリゴンメッシュの簡単化技術は、LOD 制御によるレンダリング処理の軽減などにおいて重要な役割を果たしている。これまでにさまざまな簡単化手法が提案されているが、適用目的に応じてメッシュの品質と処理速度とのいずれかに特化しているものが多かった。本論文では、三次元測定機などから得た点群をもとに生成された三角形メッシュの簡単化を目的として、メッシュの品質と処理速度の両方の条件をバランス良く満たす簡単化手法を提案する。本手法では、元形状の形状特徴を局所的に抽出し、かつメッシュ生成過程で失われた特徴を周辺の特徴をもとに復元する操作を適用しながら簡単化するため、エネルギー最小化などコストの高い計算を伴わずにメッシュの品質を保存することができる。また、基本的な位相操作によってポリゴン稜線を削除する手法を用いるため、再メッシュ化の必要がないことも低コスト化に寄与している。

Mesh Simplification Using Edge Operation with Feature Detection

Masayuki Hayano†, Tsukasa Matsuoka, Kenji Ueda
Software Research Center, Ricoh Company, Ltd.
†mhayano@src.ricoh.co.jp

Mesh simplification is an important technique to reduce the cost for rendering. Many previous simplification methods are useful whether in quality of mesh, or in speed. We propose a new method which is able to satisfy both quality and speed to simplify the mesh generated from point-set data. Our method can produce high quality approximations of polygonal meshes without energy minimization because of local feature detection and local restoration of the feature. Moreover, iterative local edge operation for mesh reduction without re-meshing non-triangular regions contributes to reduce the cost.

1 はじめに

3Dコンテンツの表現手段として、ポリゴンメッシュによる形状表現が現在主流になっている。3Dコンテンツの応用域の拡大とともに、より複雑でデータ量の多いメッシュを扱う必要が生じている。特に近年では、直接モデリングしたメッシュのみならず、三次元測定機によって得られる点群データなどから自動的に生成されたメッシュの利用が進められている。このようにして得られたメッシュは、膨大な数のポリゴンによって構成されている場合が多いので、そのままでは表示やデータ転送の効率が悪く、実用的でないという問題がある。そこで、ポリゴンメッシュから冗長な要素を取り除き、実用的なサイズのデータを得るために、メッシュを簡単化する必要がある。

以下に、これまでに提案されている代表的なメッシュ簡単化手法とその特徴を示す。

1. エネルギー最小化による方法

初期メッシュからの誤差距離やメッシュ頂点の数、稜線をばねとしたときのばねエネルギーといった条件をもとにエネルギーを求める。このエネルギーが最小になるようポリゴンを削除していく手法である。品質の良いメッシュが得られるが、計算コストが高い。Hoppe らの方法 [2] に代表される方法である。

2. 局所的な特徴抽出と頂点削除による方法

初期メッシュに対して、近似曲率を算出して抽出した形状特徴の保存を条件とし、ポリゴン頂点の削除と削除部分にできる非三角形領域の再メッシュ化の繰り返しによって簡単化する方法である。エネルギー計算なしに特徴抽出、保存を行っているが、再メッシュ化の必要がある。Schroeder の方法 [4] や Véron の方法 [5] などがある。

3. 空間分割と頂点集約による方法

メッシュの占める空間を規則的に分割し、それぞれ

の部分空間に存在する頂点が1つの頂点に集約するようポリゴン頂点を削除する方法である。メッシュの位相関係を考慮しないので高速に処理できるが、特徴が失われやすく形状が大きく崩れることがある。代表的なのは Rossignac の方法 [3] である。

メッシュ簡単化は、メッシュの品質と処理速度とのトレードオフであり、従来は、適用目的に応じてそのどちらかに特化した手法が提案されてきた。本論文では、点群から生成されたメッシュの簡単化を目的とし、かつ元形状の特徴の保存と処理速度の両方の条件をバランス良く満たすメッシュ簡単化手法を提案する。本手法では、ポリゴンの面のなす角度をもとに、形状特徴を表すポリゴン稜線を特徴稜線として抽出、保存する。点群からのメッシュ生成や均等なポリゴン削減の過程においては、メッシュの大域的な特徴を表すような連続した稜線列は断続的になってしまっていることが多い。そこで、すでに抽出されている特徴稜線列をメッシュの位相変更によって延長し、特徴稜線列を補間する操作を適用する。これにより、エネルギー最小化計算を行わずにより良質な形状特徴の抽出、保存ができる。また、ポリゴン稜線削除の位相操作でポリゴンを削除するため、簡単化の過程で非三角形領域が生成されることがなく、再メッシュ化の計算も不要である。なお、本手法における形状特徴の抽出は、Véron の方法 [5] をベースにしており、また、特徴稜線の延長とポリゴン削減操作は、Hoppe らの方法 [2] をベースにしている。

2 アルゴリズムの概要

点群から生成された初期三角形メッシュに対し、以下の処理を適用して簡単化を行う。

1. **特徴稜線の抽出と延長**
   隣接するポリゴンのなす角度を評価して特徴を表す稜線を抽出する。また、稜線交換操作を適用して特徴稜線列を延長する。
2. **特徴頂点の抽出**
   特徴稜線の接続数および頂点における曲率を評価して、特徴を表す頂点を抽出する。
3. **稜線縮退化によるポリゴン削減**
   特徴頂点を除くすべてのメッシュ頂点を、特徴稜線の端点である頂点群、それ以外の頂点群に分類する。2種類の頂点群からそれぞれ曲率最小の頂点を取り出し、これに接続する稜線のうち、稜線縮退化操作適用前後で最も形状の変化の少ない稜線に対して、稜線縮退化操作を適用する。
4. **終了条件**
   メッシュが設定した条件を満たすまで1に戻って処理を繰り返す。条件としては、初期メッシュからの誤差距離、もしくは簡単化後のポリゴン数を設定しておく。

3 メッシュ簡単化方法

3.1 特徴稜線の抽出と延長

まず、メッシュのすべての稜線の両側の面についての二面角を評価して特徴稜線を抽出する。図1のように、二面角 A は、稜線を共有している2つの面のなす鋭角の角度とする。二面角 A が指定した閾値 A_t 以下の場合、特徴稜線と判定する。図1中の太線は、特徴稜線と判定されたことを示す。

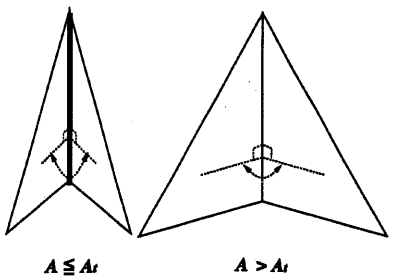


図 1: 二面角 A の評価による特徴稜線の抽出

次に、特徴稜線列を延長する。特徴稜線列の延長には、稜線交換操作を適用する。三角形ポリゴンメッシュにおける稜線交換操作は、図2のように、稜線 $\{V_tV_u\}$ を削除し、稜線をはさむ2つのポリゴンについて向かい合う頂点 $\{V_lV_r\}$ を両端点とする稜線を生成する操作と定義される [2]。特徴稜線の延長は、図3のように適用する。なお、図3中の太線の稜線は、すでに抽出されて いる特徴稜線であることを示す。特徴稜線の途切れる個所にある頂点、つまり1本だけの

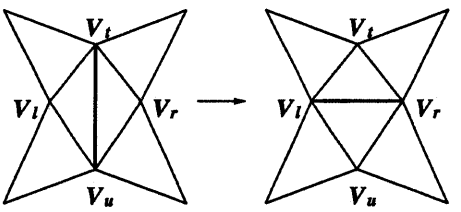


図 2: 稜線交換操作

特徴稜線に接続する頂点を V_l とし、V_l に対向する稜線の両端点を V_t, V_u、稜線 $\{V_tV_u\}$ を共有する二つの面のうち、V_lが属していない面において稜線 $\{V_tV_u\}$ に対向する頂点を V_rとする。

1. 面 $\{V_l, V_r, V_t\}$ および 面 $\{V_l, V_u, V_r\}$ のなす二面角 A を算出し、A_t と比較する。A が A_t 以下の場合、稜線 E に対して稜線交換操作を適用する。A が A_tより大きい場合は適用しない。
2. 稜線交換により生成された新たな稜線を、特徴稜線と判定する。その結果、特徴稜線列が延長される。

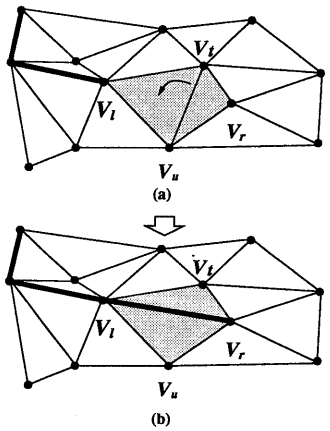
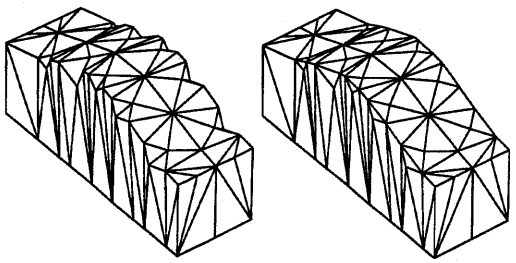


図 3: 稜線交換操作による特徴稜線の延長

この操作により、図4(a) のように、生成過程で特徴稜線付近が欠けてしまったようなメッシュの特徴を、(b) のように復元することができる。



(a) 適用前　　(b) 適用後

図 4: 特徴稜線の延長の効果

3.2 特徴頂点の抽出

特徴頂点とは、形状のコーナー部分など、形状が大きく尖っているかまたは凹んでおり、その頂点において形状の特徴を表すと考えられる頂点である。このような頂点がポリゴン削減の過程で失われないように、頂点に接続する特徴稜線数と曲率を用いて特徴頂点を抽出しておく。

- 特徴稜線数が 1 か、もしくは 3 以上の場合
  頂点は形状のコーナー部分にあると判断し、特徴頂点と判定する。
- 特徴稜線数が 0 の場合
  近似ガウス曲率を求め、閾値よりも大きい場合、特徴頂点と判定する。

近似ガウス曲率は、曲面を近似した多面体の頂点における曲率の近似値である [1]。頂点 V における近似ガウス曲率 K は式 (1) により求める。

$$K = \frac{a}{S} \tag{1}$$

a および S は次のように定義される。a は、図5のように表すことができる。

$$a = 2\pi - (V\text{に集まる面の隅の角度の総和})$$
$$S = \frac{1}{3}(V\text{に集まる面の面積の総和})$$

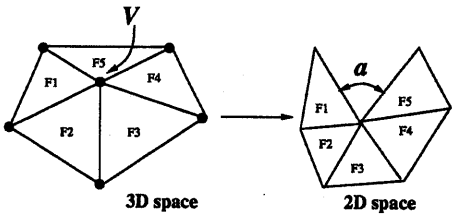


図 5: 近似ガウス曲率算出に用いる角度 a

Kの絶対値が大きいほどその頂点において形状は尖って (凹んで) いるといえる。したがって、特徴頂点検出のための閾値を K_t とすると、$|K|$ が K_t 以上である頂点を特徴頂点と判定する。

3.3 稜線縮退化によるポリゴン削減

特徴頂点以外のすべての頂点について、頂点に接続するいずれかの稜線に対し、稜線縮退化操作を適用してポリゴン数を削減する。稜線縮退化操作は、図6

のように、稜線 $\{V_tV_u\}$ を端点 V_u に向かって縮退させることにより、1 つの頂点 と2つの三角形ポリゴンを削除する操作である [2]。ポリゴン削減は、メッシュの滑らかな部分から順に行う。そこで、すべての頂点について曲率を計算し、曲率最小の頂点に接続する稜線から順に稜線縮退化を適用してポリゴンを削除する。頂点における曲率が小さいほど、その周辺の形状は滑らかで、ポリゴン削除に伴う形状変化が最も小さいと考えられるからである。すでに抽出されている形状特徴を残すため、特徴稜線は特徴稜線がつながっている方向にのみ縮退させる。また、特徴頂点は削除しない。

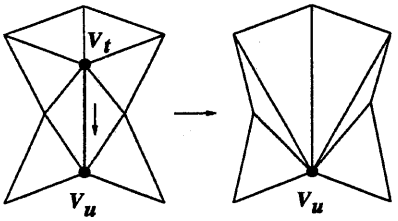


図 6: 稜線縮退化操作

頂点の分類

縮退させる稜線の順番と適用方法を決定するため、次のように頂点を分類する。

1. 特徴頂点の集合 $\mathbf{V}_C$
2. 特徴頂点でなく、かつ特徴稜線に接続していない頂点の集合 $\mathbf{V}_K$
3. 特徴頂点でなく、かつ特徴稜線に接続している頂点、すなわち2本だけの特徴稜線に共有されている頂点の集合 $\mathbf{V}_M$

例えば図 7のように頂点が分類される。図 7中の太線は特徴稜線、黒丸は特徴頂点、文字は頂点が上の3種類のうちどの頂点集合に分類されるかを示している。

一般稜線の縮退化

頂点集合 $\mathbf{V}_K$ に属する頂点に接続する稜線は、必ず特徴稜線以外の一般稜線の端点であり、特徴頂点抽出のために既に近似ガウス曲率 K が設定されている。そこで一般稜線は、$\mathbf{V}_K$ に属する頂点のうち K の絶対値が最も小さい頂点に接続する稜線から順に縮退化の対象とする。また、これらの稜線のそれぞれについて、図 8に示すように、縮退化後に生成される

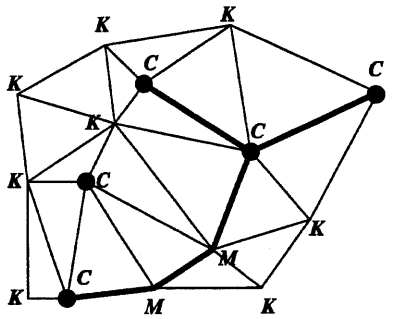


図 7: ポリゴン頂点の性質による分類例

面と削除される頂点 V_t との誤差距離 d が最小の稜線に対して縮退化操作を適用する。

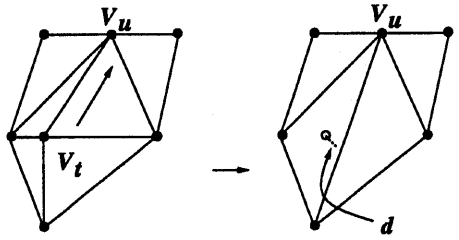


図 8: 稜線縮退化操作前後の誤差距離

特徴稜線の縮退化

頂点集合 $\mathbf{V}_M$ に属する頂点には、特徴稜線上の近似曲率 M を設定する。M は頂点が2本の特徴稜線に共有されている場合にのみ定義される。図 9に示すように、頂点 V を2本の特徴稜線 E_l, E_r が共有しているとき、V における M を式 (2) のように定義する。

$$M = \frac{b}{L} \tag{2}$$

b および L は次のように定義する。

$$b = \pi - (E_l \text{と} E_r \text{のなす角度})$$
$$L = \frac{1}{2}(E_l \text{と} E_r \text{の長さの合計})$$

Mの絶対値の小さい頂点ほど、特徴稜線のつながっている方向について滑らかであると判断する。そ

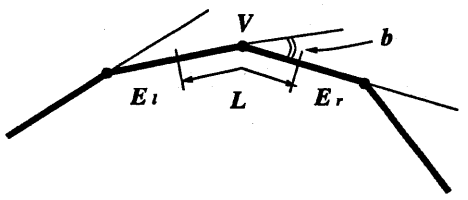


図 9: 特徴稜線上の近似曲率の定義

こで特徴稜線は、$|M|$ 最小の頂点に接続する特徴稜線から順に縮退化の対象とする。さらに頂点に接続する2本の特徴稜線それぞれについて、一般稜線と同様に誤差距離 d を求める。d の小さい、すなわち形状変化の少ない方の特徴稜線に対して稜線縮退化操作を適用する。

曲率の閾値による簡単化の制御

曲率を閾値として与えることにより、メッシュ上の指定した曲率以下の部分だけを簡単化することができる。閾値以上の $|K|$ や $|M|$ を持つ頂点に接続する稜線を、縮退化の候補に加えないようにすることでこれを実現している。

3.4 終了判定

簡単化を終了させるための条件として、次のいずれかを指定する。

- 初期メッシュの頂点からの誤差距離

  簡単化後のメッシュが、初期メッシュから指定した誤差内に入るようなメッシュを得たい時に指定する。稜線縮退化操作は、初期メッシュの頂点からの誤差距離が指定した値を越えない場合にのみ適用することとし、すべての稜線について縮退させることができなくなった時、処理を終了する。

- 簡単化後のポリゴン数

  簡単化後のメッシュが、指定したデータ量を持つようにしたい時に指定する。指定したポリゴン数まで削減されたら処理を終了する。

4 適用例

本手法を PC/AT 互換機 (Pentium II, 128MB-Memory) 上に実装してメッシュ簡単化を実行した結果を、図10および図11に示す。各図の (a) は初期メッシュ、(b)(c) は適用結果である。また、各図にはメッシュのポリゴン数と計算時間も示した。なお初期メッシュは、CAD ソフトで作成したモデルを元に生成した。図10は、初期メッシュからの誤差距離 t の指定を変化させて適用した例である。(b) は t が小さいため、比較的滑らかな外周部分のみ簡単化され、全体として初期メッシュからの形状変化が抑えられている。一方 (c) は t が大きいため、全体的に簡単化が行なわれていることがわかる。図11は、t を固定し、稜線縮退化の閾値として曲率 K を与えて簡単化した例である。(b) と (c) では、平面の部分ではほぼ同程度簡単化されているのに対し、円柱状の部分において簡単化の度合に違いが表れている。またこの例からは、特徴稜線の保存と、特徴稜線方向への稜線縮退化が効果的に行なわれていることが分かる。

5 まとめ

本稿では、元の形状の特徴をよく保存し、かつ低コストな新しいメッシュ簡単化法を用いて、必要に応じたデータ量のメッシュを生成する手法を提案した。しかし現状では、稜線縮退化の順番を曲率のみで決定しているため、平面上など曲率の等しい頂点が多く存在する部分では、ある一部分に集中して簡単化が行なわれることがあり、誤差距離や曲率の条件を満たしているポリゴンでも、位相的な制限で削除できずに残ってしまう。今後の課題としては、条件を満たす範囲内でランダムに簡単化されるような方法を考慮すること、生成される三角形形状の品質をより向上させること、さらなる高速化、メッシュの詳細度を調節できるような拡張、などが考えられる。

参考文献

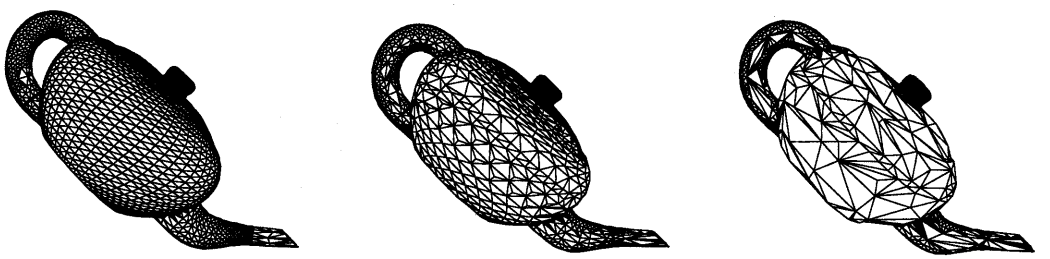
[1] C.R.Calladine, Gaussian curvature and shell structures, *The Mathematics of Surfaces*, Oxford University Press, pp.179-196, 1986.

[2] H.Hoppe, T.DeRose, T.Duchamp, J.McDonald and W.Stuetzle, Mesh Optimization, *Proc. ACM SIGGRAPH '93*, pp.19–26, 1993.

[3] J.Rossignac and P.Borrel, Multi-resolution 3D approximation for rendering complex scenes, *Modeling in Computer Graphics*, pp.455–466, 1993.

[4] W.J.Schroeder, J.A.Zarge and W.E.Lorensen, Decimation of triangle meshes, *Proc. ACM SIGGRAPH '92*, pp.65-70, 1992.

[5] P.Veron and J.C.Lenon, Static polyhedron simplification using error measurements, *Computer-Aided Design*, 29, 4, pp.287–298, 1997.

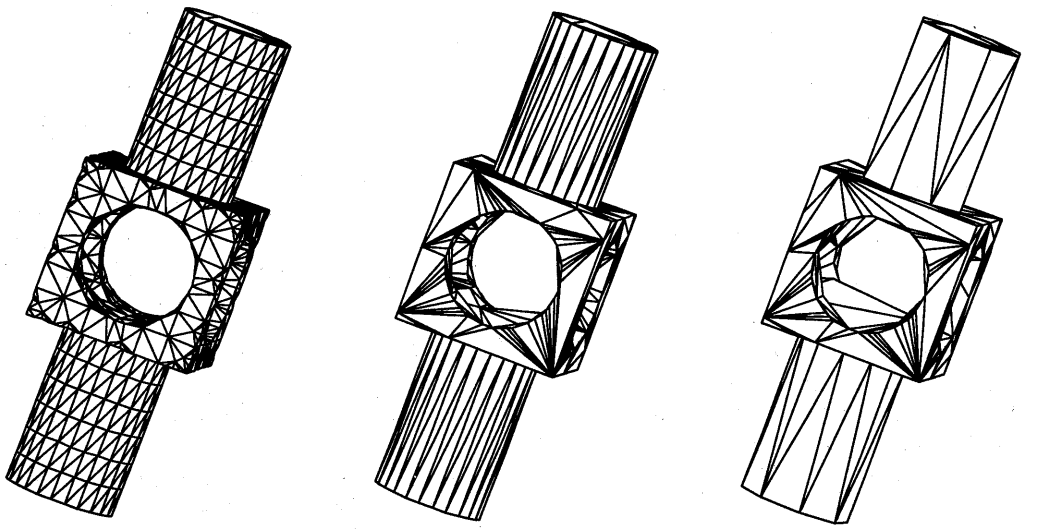


(a) 初期メッシュ
(5932 polygons)

(b) 適用後 ($t = 6$)
(4218 polygons : 366 sec.)

(c) 適用後 ($t = 20$)
(2904 polygons : 276 sec.)

図 10: 誤差距離指定による簡単化の例



(a) 初期メッシュ
(2028 polygons)

(b) 適用後 ($K = 0.028$)
(706 polygons : 583 sec.)

(c) 適用後 ($K = 0.666$)
(450 polygons : 827 sec.)

図 11: 曲率による簡単化制御の例