空間分割を用いた形状簡略化の一手法

A Method of Model Simplification Using Spatial Partitioning

正会員 金谷孝之[†], 正会員 小堀研一^{††}Takayuki Kanaya[†] and Ken-ichi Kobori^{††}

Abstract We propose an algorithm for automatic simplification of models of 3D objects based on a simplification technique that applies vertex-clustering. The major advantages of the vertex-clustering technique are its low computational cost and high data reduction rate. Thus, the method is suitable for use in real-time rendering.

This paper presents a clustering method for better preservation of important features in the shape. Its major contributions include: improved tight octree, described by Luebke, and high accuracy as a result of estimating vertices for the distance between the octants at each level.

キーワード: 形状簡略化, 空間分割, LOD, 8分木, 頂点クラスタリング

1. ま え が き

仮想空間ウォークスルーシステムやビデオゲームなどといった対話的な3次元グラフィックスにおいては、よりリアルな3次元モデルが要求されるようになった。一方、最近のCADシステムや3次元スキャナーなどのデバイスによって複雑な形状が容易に作成できるようになった。しかし、形状の複雑化に対してハードウェアの進歩が追いついていない。そのため、形状が複雑になると計算コストが急激に増大し、実時間で描画することが困難になる。そこで、例えば、ユーザの視点から遠方にある物体などについては形状を簡略化することなどが行われている。

形状の簡略化についてはこれまで多くの研究がなされている¹⁾。元形状の表面にランダムに新しく頂点を置き、元形状の曲率などを基にその頂点群から再メッシュ化する手法²⁾、ウェーブレットによる手法³⁾、幾何学的な評価値により頂点、稜線、または面を繰り返し除去する手法⁴⁾⁵⁾、評価関数としてエネルギー関数を定義し、それが最適になるような稜線を統合したり、分割したりする手法⁶⁾⁷⁾、および、幾何学的に接近している頂点をその距

離によってグループ化し、頂点を統合していく手法^{8)~12)}などが挙げられる。これらの手法においては、高速であるが位相構造を保持していない、高品質ではあるが計算コストが大きい、解像度が連続的でないなど、手法によって一長一短である。

本手法は、幾何学的に接近している頂点を段階的に統合していく頂点クラスタリングの手法の一種である。一般に頂点クラスタリング手法は、高速な手法として知られているが、位相構造や精度を充分保持していないという問題がある。そこで本手法では、形状の頂点群が存在する空間を8分木¹³⁾を用いて分割し、またその分割した空間の大きさに頂点をクラスタリングする。そして、そのクラスタリング情報を各頂点に付加し、その情報を基に頂点を統合し、形状を簡略化する。以上によって、頂点クラスタリングの高速性を活かしつつ、品質を保持した形状の簡略化を実現する。

2. 頂点クラスタリング法

一般に、描画速度と描画品質はトレードオフの関係にある。一つの形状に数万ポリゴン以上の精度が要求される今日では、その描画品質を保持しながらリアルタイムに描画を行うことは、ハードウェアの面から不可能である。

形状の簡略化手法を分類すると大きく2つに分けることができる。一つは、描画速度に主眼をおいた手法、もう一つは描画品質に主眼をおいた手法である、前者の手法の一つに頂点クラスタリングの手法がある。

頂点クラスタリング手法は、幾何学的に接近している頂点に対し、頂点をクラスタリングし、それぞれのクラ

2000年10月5日, 第61回情報処理学会全国大会で発表

2001年8月6日受付, 2001年11月12日再受付, 2001年12月21日採録

[†]広島国際大学 医療福祉学部

(〒724-0695 広島県賀茂郡黒瀬町学園台 555-36, TEL0823-70-4628)

^{††}大阪工業大学 情報科学部

(〒573-0196 大阪府枚方市北山 1-79-1, TEL072-866-5391)

[†]Faculty of Health and Welfare, Hiroshima International University
(555-36, Gakuendai, Kurose-cho, Kamo-gun, Hiroshima 724-0695, Japan)

^{††}Faculty of Information Science, Osaka Institute of Technology
(1-79-1 Kitayama, Hirakata-shi, Osaka 573-0196, Japan)

図 1 Rossignac-Borrel の手法 (2次元で表示)
A method of Rossignac-Borrel.

スタから新しい頂点を作成する手法である。この手法は高速ではあるが、位相構造や十分な精度を保持していないという問題を持っている。

Rossignac や Borrel は図 1 に示すように、一つのセルに一つの頂点だけが格納されるまで再帰的に 3 次元空間を 8 等分し、それによってできた 8 分木を用いてレベルを制御する方法を提案した⁸⁾。Low や Tan は Floating セルを用いて視界に応じた適応的表示を行った⁹⁾。Reddy は sub-volume で空間を分割し、稜線長や曲率によってポリゴンを削除した¹⁰⁾。また、Luebke や Erikson は Rossignac-Borrel の手法を改良し、再帰的にセルを分割するときにそのセル毎に頂点を包括する最小のセルを再構築し、それによってできた 8 分木 (Tight Octree) を用いて視線の方向によって形状の簡略化のレベルを制御する手法を提案した¹¹⁾。本稿では、空間分割法を用いた頂点クラスタリング法としてよく知られた Luebke-Erikson の手法を従来法として比較を行った。従来法ではレベルづけの際セルの大きさを考慮していないため本来特徴的な頂点が初期の段階で統合されたり、特徴的でない頂点が粗いレベルまで残ったりすることがある。また、頂点合成の際にそのレベルの特徴点しか考慮していないため同等の特徴的な頂点も空間分割によって近隣の特徴点に統合されてしまう。そこで本稿では、簡略化が進んだレベルまで特徴的な頂点の影響を受けるような手法を提案する。

3. 提案手法

本手法では、三角形メッシュの組合せからなる Winged-Edge データ構造¹³⁾ で表現された多面体の 3 次元 CG データを対象にしている。

まず、本手法の概要手順を述べ、その後、各々の詳細について述べる。

- (1) 3 次元空間に存在しているすべての頂点群を包括するような空間を、一つの空間に一つの頂点が格納されるまで再帰的に分割することによって、各々の頂点が葉ノードである 8 分木を生成する。
- (2) (1) で 8 分木を生成する際の分割された空間の大きさによりレベルづけを行い、その情報を 8 分木のノードに付加する。

図 2 8 分木の生成例 (2次元で表示)
A example of generated octree.

- (3) その 8 分木において子のノード頂点座標より親のノードに合成後の頂点座標を作成する。
- (4) 位相情報を基に三角形メッシュを作成し、形状を構築する。その際に、ノードが持つレベル情報によって形状の詳細度を制御し、描画する。

3.1 空間分割による 8 分木の生成

任意の 3 次元空間を分割することによって葉ノードに各頂点が格納された 8 分木を生成する。その 8 分木の生成法は以下の 3 つの手順からなる。

- (1) 3 次元空間に存在しているすべての頂点群を包括するセルを作成する。
- (2) そのセルを 8 等分する。
- (3) 8 等分した各々の直方体に属する頂点数をカウントし、一つの直方体に一つの頂点だけであれば、終了。そうでなければ、各々の直方体について (1) の処理を行う。

上記の 3 つの手順を行うことによって、図 2 に示すような 8 分木を得ることができる。ここでセルとは、 xy 平面、 yz 平面、および zx 平面に平行な面で構成される複数の頂点群を包括するように最小化した直方体のことである。

この手順で作成された 8 分木の葉ノードには各頂点の位置座標とその一頂点が包括されている直方体の大きさの情報が、また葉ノードでないノード (以下、中間ノードという) には複数の頂点を包括したセルの大きさの情報が格納される。葉ノードの直方体も中間ノードのセルも大きさに関し同様に扱うことができるため、以下セルと呼ぶ。

3.2 空間の大きさによるレベルづけ

3.1 節で生成した 8 分木における各ノードに格納されたセルの大きさを考慮し、レベルづけを行う。このレベルを用いて描画時の詳細さのレベルを制御する。レベルの数値が大きい方が精細である。

(1) 基準レベルの定義

まず、基準レベルの作成方法について述べる。ここでは、レベルづけの際に基準となるセルの大きさを定義する。この基準となるセルの大きさと実際のセルの大きさを比較することによってレベルづけを行う。

3 次元空間に存在しているすべての頂点群を包括するような最小の直方体の大きさをレベル 0 とする。それを

図 3 面積が同じでレベルが異なる場合
An example of different level in same area.

xy 平面, yz 平面, および zx 平面に平行な面を用いて 8 等分する. 8 等分されたそれぞれの直方体の大きさがレベル 1 である. 以下同様にして等分割を繰り返し, その分割回数を用いて基準レベル数とする.

(2) 実際のレベルづけ

セルの大きさを考慮してレベルづけを行う場合, 3 次元であるため体積で考えるのが普通であるが, セルの最大の一辺を考慮することによりレベルづけを行った. その理由を図 3 にて説明する. ここでは簡単にするために 2 次元にて説明を行う.

図 3 に実線で示す左右の四角形はあるセルを分割した直後の長方形であり, 網掛けの部分はその長方形に存在する頂点群を包括する最小のセルである. この 2 つのセルの大きさを同じ面積であると仮定する. しかし, 両長方形のそれぞれの辺長が大きく異なる. また, 点線の四角形は基準レベル A のセルの大きさである. 面積だけを考慮してレベルづけを行うと, 網掛けのセルに存在する頂点群は同じレベルになってしまうが, 実際には基準レベル A は図 3 に示す点線の四角形の大きさであるため網掛けのセルはその領域からはみ出してしまふ. その結果, 実際のセルの大きさと基準レベルの間に矛盾が生じてしまふ. これは 3 次元の場合についても同様である. そこで, セルの最大の一辺を考慮することによりレベルづけを行う. 具体的には式 (1) を用いてレベルを計算する.

$$level = INT(\log_2 \frac{Length_{max}}{Length_{current}}) \quad (1)$$

ただし, $INT()$ は整数化する関数, $Length_{max}$ はすべての頂点を包括するセルにおける最長の一辺の長さで, $Length_{current}$ は現在の対象となるセルにおける最長の一辺の長さである.

上記で求めたレベルを 3.1 節で求めた 8 分木のそれぞれのノードに格納し, それを用いて 8 分木の深さを制御すると図 2 の 8 分木は図 4 のようになる.

ここで求めたレベル情報を持った葉ノードを頂点として, 位相情報を基に Winged-Edge データ構造を作成する.

3.3 頂点の合成法

現状の 8 分木において頂点の位置座標が格納されているのは葉ノードだけである. そこで, 各レベルの中間ノードに頂点座標を求める頂点の合成手法について述べる.

図 4 図 2 の提案手法による木構造表現
Tree construction generated using equation 1.

図 5 頂点合成の例
A example of vertex synthesis.

本手法ではレベル毎に稜線の長さで重みづけを行った. 精細なレベルから粗いレベルへと頂点の合成を行う. 例えば図 4, 図 5 において, レベル 2 からレベル 1 について頂点合成の処理を行う場合について考える.

8 分木の根ノード E から探索を開始し, レベル 1 のノードを探索する. もしも探索した枝において, 指定したレベル以上のノードがあるにもかかわらず, そのレベルに対応したノードが存在しない場合は, その枝をレベルの大きい方へ探索し, 最初に存在するノードを選択する. 探索の結果, ノードは座標が格納された葉ノード (10), もしくは最大レベルの頂点が格納された子ノードを持った中間ノード (B, C, D) が選ばれる. 葉ノードが選ばれた場合は, そのレベルの頂点としてそのまま採用する. 中間ノードが選ばれた場合は, そのノードに格納されたセルの中にある頂点に接続するすべての稜線長を計算し, それぞれの頂点に接続する稜線長の中で最大の値をそれぞれの頂点の重みとする. しかし, このときに稜線長の計算を元形状のすべての頂点について行うのではなく, 以下で生成法を説明する疑似頂点との稜線長を計算し, 重みとする. この疑似頂点を用いることによって, 長さの短い稜線長計算を省略でき, 計算コストを軽減できる. また, 他の同レベルの頂点を探索して, 同様な処理で重みづけを行う.

先にも述べたように, 中間ノードは座標が確定された頂点を格納している子ノードの親である. したがって, 図 6 に示すようにこのレベルにおけるセルはその子以下の複数の頂点が格納された最小の空間になっている. こ

図 6 セルの中心座標による疑似頂点
Pseudo vertex using center coordinate of cell.

図 7 疑似頂点による重み計算
Calculation of weight by pseudo vertex.

の空間の中心座標を計算し、この値をこのレベルにおける疑似頂点とする。図 5 についてレベル 1 における疑似頂点処理の結果は図 7 のようになる。つまり、要求したレベルに対し孫以下に相当するレベルの頂点は疑似頂点にてすべて置き換えられる。

合成頂点を計算するために 8 分木を探索し、中間ノードが選ばれた場合は、元形状における位相情報を基にその子の頂点に接続する頂点、疑似頂点の稜線長を計算し、その最大値を重みとして式 (2) を用いて合成後の頂点座標 $V_{synthesis}$ を決定する。

$$V_{synthesis} = \frac{\sum_{i=0}^n (V_{previous} \times W_i)}{\sum_{i=0}^n W_i} \quad (2)$$

ただし、合成の対象となる前レベルの頂点を $V_{previous}$ 、その頂点数を n 、それぞれ頂点 i に対する重みを W_i とする。

以下、同様にしてすべてのレベルに関し合成後の頂点を計算し、すべての中間ノードに頂点座標が格納されるまで繰り返す。

3.4 形状の描画

以上で求めた 8 分木および Winged-Edge データ構造を基に、形状を描画させる。ユーザが描画させたいレベルを任意に指定し描画させる。また、観測者の視点から物体までの距離によってレベルを変更するということも可能である。

ここでは三角形メッシュを生成し、描画させる。頂点

データは 8 分木のノードであるので、3.2 節で生成したレベル情報を所有している。そこで、ユーザが任意に指定した描画レベルとその頂点のレベルを比較する。等しい、あるいは頂点レベルが小さい（粗い）レベルであれば、それを三角形の一頂点として採用し、逆に頂点レベルが大きい（詳細な）レベルであれば、等しくなるまで再帰的に 8 分木を探索し、レベルが一致した頂点を三角形の一頂点として採用する。

以下、すべての頂点について同様な処理を行うことによって、三角メッシュで構成された簡略化後の形状を描画することができる。

3.5 レベル間の頂点の補間

上記で述べた処理では、レベル毎の形状は得ることができるが、ポリゴン数を任意に指定することができない。これは頂点クラスタリング手法の特徴でもある。そこで、図 8 に示すように、レベル i (i は整数) とレベル $i+1$ 間の頂点を線形補間することによって、ポリゴン数を任意に指定することができるように拡張した。ただし、レベル i はレベル $i+1$ よりも粗いレベルである。

レベル i とレベル $i+1$ 間のレベルを j (j は実数) とし、 $j-i=m$ 、 $m+n=1.0$ とすると、三角形の各頂点の補間座標 $V(j)$ を

$$V(j) = \frac{nV(i) + mV(i+1)}{m+n} \quad (3)$$

で求める。

図 8 レベル間の線形補間
Linear interpolation between levels.

図 9 任意ポリゴン数の簡略化形状の例
An example of simplification models in specified number of polygon.

図 10 実験形状と簡略化結果
Five objects in the experiments and their simplification results.

しかし、各頂点の座標を補間するだけでは、三角形ポリゴンが小さくなるだけで、全体のポリゴン数は減少せず、任意のポリゴン数を指定することはできない。そこで、閾値を設け、 $V(i)$ と $V(j)$ との距離が以下に示す閾値より小さいときは $V(j) = V(i)$ とし、レベル j のポリゴン数を減少させる。ここで、閾値 $threshold$ は元形状のすべて頂点を包括するセルの最長の一边を L とした場合、

$$threshold = \left(1 - \sqrt[3]{\frac{9}{16}}\right) \frac{L}{2^i} \quad (4)$$

と定義した。

その理由を以下に示す。レベル i のセルの辺長がすべて ℓ の立方体であったと仮定すると、レベル i の体積は ℓ^3 であり、レベル $i+1$ の体積は $\ell^3/8$ となる。よってレベル i のセルの大きさとレベル $i+1$ のセルの大きさとの差が $1/2$ になるときの体積は

$$\frac{(\text{レベル } i \text{ の体積}) - (\text{レベル } i+1 \text{ の体積})}{2} = \frac{7}{16}\ell^3$$

である。これにレベル $i+1$ のセルの体積を加えると、レベル i とレベル $i+1$ の境界は $9\ell^3/16$ となる。よって、3.2 節で述べたようにセルの一边でレベルづけを行うため、このときの立方体の一边は $\sqrt[3]{9/16}\ell$ となる。したがって、レベル i のセルの一边から求めた境界の一边を引いた値を閾値とした。

図 9 にポリゴン数を 3,000, 2,000, および 1,000 に指定したときの bottle 形状の簡略化結果を示す。

4. 実験

本手法の有効性を検証するために、いくつかの形状に対し従来法と本手法の簡略化結果の比較、各レベルに対するポリゴン数の比較、およびポリゴン数に対する形状のメッシュ生成時間の測定を行った。なお、実験には AT 互換機 (CPU : Pentium III 600MHz) を使用した。

実験で扱った多面体形状の一例を図 10 (a) に示す。また、名前の横に元形状のポリゴン数を示す。

そして、従来法によるレベル 5, 4, および 3 における形状の簡略化結果を図 10 (b), 本手法による形状の簡略化結果を図 10 (c) に示す。なお、レベルの横にそれぞれのポリゴン数を示す。

次に、bottle 形状におけるレベル毎に対する形状のポリゴン数について従来法と本手法の計測結果を図 11 に示す。

また、bottle 形状におけるポリゴン数に対する形状のメッシュ生成時間について従来法と本手法の比較を図 12 に示す。メッシュ生成時間の実験では最大レベルから表示させたいレベルにメッシュを作り直すのに必要な時間について計測し、その平均値とした。

図 11 レベルに対するポリゴン数
Number of Polygons in each level.

図 12 ポリゴン数に対するメッシュ生成時間
Mesh generating time in number of polygon.

5. 考察

まず、図 10 (a) に示す元形状と (b) に示す従来法の簡略化結果と、(c) に示す本手法の簡略化結果とを視覚的に比較してレベル 5 においては (b), (c) とともに元形状とほとんど遜色がない。しかし、(b) と (c) のポリゴン数を比較すると、例えば、apple 形状の場合において従来法ではポリゴン数が元形状の約 43% であるのに対し、本手法では約 37% と少ない。他の形状においても同様に本手法の方がポリゴン数は少ない。このことから、本手法の方が冗長的な頂点を早期の段階で削除していることがわかる。さらに、レベルが小さくなるにつれ、特にレベル 3 においては従来法では形状の凹凸が著しく現れ、car 形状ではひだが目立つようになっているが、本手法ではそのようなひだが抑えられ滑らかに表現されている。しかし、camera 形状においては、全体としては形状特徴を残しているものの、レンズ部の特徴である鋭さが従来法と比べて鈍ってしまっている。その理由は、従来法では最長の稜線が接続する頂点をそのレベルの特徴点として採用しているのに対し、本手法では式 (2) に示すように頂点に接続する稜線の重みで、平均化しているからである。

一方、図 11 に示すレベルに対するポリゴン数において、本手法は従来法に比較して、ポリゴン数の減少曲線が緩やかになっていることがわかる。これは、冗長的な頂点を早い段階で削除し、特徴的な頂点を最後まで残す傾向を示している。また、曲線が緩やかになり直線的になることにより表現したい形状のポリゴン数を制御することが容易となり、レベル間での急激なポリゴン削除が抑制される。

また、図 12 に示すポリゴン数に対するメッシュ生成時間においては従来法では木構造を探索するだけなのに対して、本手法では木構造の探索に加え、3.5 節で述べた線形補間の処理も行っているため、従来法より僅かに時間を要している。しかし、縦軸と横軸とも線形であるため、2つの手法に処理時間の差はほとんどないといえる。なお、他の形状についても図 11、図 12 と同様な傾向がみられた。

図 10、図 11、および図 12 からわかるように、本手法を用いることによって従来法と比較して時間的な差がほとんどなく、レベル数が大きいときには冗長的な頂点を早期に統合し、レベル数が小さいときでも形状の特徴を表わすことができたといえる。

6. む す び

本手法では、形状の特徴的な頂点をより特徴づけるために Luebke-Erikson の手法を改良し、セルの大きさを考慮したレベルづけを行った。また、頂点の合成に疑似頂点を導入し、各レベルにおいてそれぞれの頂点に重みづけを行った。さらに、頂点クラスタリング手法では困難であったポリゴン数の任意指定を可能とした。以上の手法を実装して、実験を行い、頂点クラスタリング手法の高速性を活かしつつ、従来法に比べポリゴン数が少ない場合でも、形状の特徴をより特徴づけることができることを確認した。

また本論文では、実装を行っていないが、Luebke-Erikson のように視線によって簡略化レベルを制御することも容易である。本手法は、Luebke-Erikson の手法における頂点クラスタリング手法の改良であるので、その頂点クラスタリングの手法を置き換えることによって、画面上での誤差評価、シルエット評価、三角形総数評価などを行い、視線による簡略化レベルの制御も可能となる。

今回は、簡略化形状に対して視覚的な評価を行ったが、Garland らが提案しているような定量的な評価法¹²⁾を確立し、視覚以外の客観評価も行い、また、プログレッシブな形状の簡略化を検討する。

【文 献】

- 1) P. Cignoni, C. Montani and R.Scopigno: "A COMPAISON OF MESH SIMPLIFICATION ALGORITHMS", Computer & Graphics 22, 1, pp.37-54, (1998)
- 2) G. Turk: "Re-Tilling Polygonal Surfaces", Proceeding of SIGGRAPH '92, pp.55-64, ACM Press, (1992)

- 3) M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle: "Multiresolution Analysis of Arbitrary Meshes", Proceeding of SIGGRAPH '95, pp.173-181, ACM Press, (1995)
- 4) J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright: "Simplification Envelops", Proceeding of SIGGRAPH '96, pp.119-128 ACM Press, (1996)
- 5) W. J. Schroeder, J. Zarge, and W. E. Lorensen: "Decimation of Triangle Meshes", Proceeding of SIGGRAPH '92, pp.65-70, ACM Press, (1992)
- 6) H. Hoppe: "Progressive Meshes", Proceeding of SIGGRAPH '96, pp.99-108, ACM Press, (1996)
- 7) H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle: "Mesh optimization", Proceeding of SIGGRAPH '93, pp.19-26, ACM Press, (1993)
- 8) J. Rossignac, P. Borrel: "Multi-resolution 3D approximations for rendering complex scenes", Geometric Modeling in Computer Graphics, pp.455-465, (1993)
- 9) K. L. Low, T. S. Tan: "Model Simplification Using Vertex-Clustering", ACM Symposium on Interactive 3D Graphics, pp.75-82, (1997)
- 10) M. Reddy: "SCROOGE: Perceptually-Driven Polygon Reduction", Computer Graphics Forum, 15(4), pp.191-203, Blackwell Publishers, (1996)
- 11) D. Luebke, C. Erikson: "View-Dependent Simplification Of Arbitrary Polygonal Environments", Proceeding of SIGGRAPH '97, pp.199-208, ACM Press, (1997)
- 12) M. Garland, P. Heckbert: "Surface Simplification Using Quadric Error Metrics", Proceeding of SIGGRAPH '97, pp.209-216, ACM Press, (1997)
- 13) F. Foley, A. VanDam, S. Feiner, J. Hughes: "Computer Graphics Principles and Practice Second Edition", pp.533-562, Addison Wesley Publishing Company, (1990)

かなや たかゆき
金谷 孝之 1994 年、大阪工業大学大学院修士課程修了。1994 年、シャープ(株)入社。1998 年、広島国際大学に奉職。現在、同大学医療福祉学部医療経営学科助手。3 次元形状モデリングに関する研究に従事。正会員。

こぼり けんいち
小堀 研一 1975 年、山梨大学大学院修士課程修了。1975 年、シャープ(株)入社。以後、一貫して CAD、CAM、CG に関する研究開発に従事。1991 年、大阪工業大学に奉職。現在、同大学情報科学部情報処理学科教授。CAD、CG、バーチャルリアリティの研究に従事。工学博士。正会員。