

形状近似のよい多重解像度のストリーミングメッシュ法の開発

陳 炳宇 西田友是
東京大学

Multiresolution Streaming Mesh with Shape Preserving

Bing-Yu CHEN Tomoyuki NISHITA
University of Tokyo

アブストラクト

インターネットで3次元モデルを利用する人が多くなったのに伴い、効率的に3次元モデルを構成するメッシュデータを伝送することが重要になっている。しかし、3次元モデルのデータ量が大きい場合でも、ほとんどの場合、高精度なモデルを利用する必要は少ない。そのため、本研究ではインターネット伝送するためのストリーミングメッシュを開発した。このストリーミングメッシュを利用する場合、最初に少ないデータ量で、形状と特徴が認識できる粗いメッシュを伝送する。その後、オリジナルの形状を再構成するために必要なパッチ情報を伝送して、オリジナルの3次元モデルを再構成できる。提案法は従来のメッシュ簡略化のアルゴリズムと異なって、再構成するために必要なパッチのデータ量が少なく、また粗いメッシュでも形状と特徴を認識し易い。さらに、複雑な計算の必要がないので、高速にストリーミングメッシュを生成できる。

1. はじめに

近年、インターネットを利用する人々が増加している。その上、インターネット上での3次元グラフィックスの表示に注目が集まっている。インターネットグラフィックスの分野に関して、3次元モデルを利用することは非常に多い。そのため、効率的に3次元モデルを構成するメッシュデータを伝送することが重要になっている。

しかし、3次元モデルのデータ量は通常大きいので、インターネットから3次元モデルをダウンロードするには時間がかかる。また、多くの場合、高精度なモデルを利用する必要であるとは限らない。そして、形状と特徴が認識できる簡単な形状のメッシュを提供することが必要である。そのため、本研究ではインターネット伝送するためのストリーミングメッシュを開発した。このストリーミングメッシュを利用する場合、最初に少ないデータ量で、形状と特徴が認識できる粗いメッシュを伝送する。簡単なモデルを見て、高精度のモデルが必要となる場合に、若干の必要な情報を追加伝送して、高精度のモデルを提供できる。この論文ではその必要な情報をパッチと呼ぶ。段階的にすべてのパッチを伝送するとオリジナルの3次元モデルを非損失に再構成できる。

このストリーミングメッシュは主に3つのプロセスからなる。最初は3次元メッシュの簡略化、次は簡略化したメッシュとオリジナルの形状を再構成するための必要なパッチ情報の保存、最後はストリーミングメッシュの伝送とオリジナルの形

状の再構成である。

ストリーミングメッシュのメッシュ簡略化の方法は、まずオリジナルの3次元モデルの頂点をレベル n からレベル0まで分類して、レベル間の頂点のトポロジー関係を生成する。頂点結合というオペレーションを使って、オリジナルの3次元モデルを簡略化する。また、簡略化したメッシュの頂点はオリジナルのメッシュの頂点のサブセット（頂点の位置も同様）であり、メッシュのトポロジーも同様である。

本研究のメッシュ簡略化は従来のアルゴリズムと異なって、再構成に必要なパッチでただ1つの頂点の情報を保存する必要があるため、各パッチのデータ量はより少ない。また、メッシュ上の法線ベクトルの変化と材質特性の不連続の頂点と面を自動的に探査するので、粗いメッシュの形状と特徴も認識し易い。さらに、複雑な計算の必要がないので、高速にこのストリーミングメッシュを生成できる。

また、インターネットの発展をみると、「pay-per-use」というソフトウェアが将来の主なビジネスモデルであることと考えられる。この新しいモデルの上に、異なったプラットフォームの上にサーバーからクライアントまでアプリケーションと3次元モデルを配布する必要があるかもしれない。そのため、著者らはJava [1]のみを用いて、ストリーミングメッシュ法を開発した。なお、3次元グラフィックスレンダリングは著者らが開発したjGL¹ [3]を利用した。

¹ <http://nis-lab.is.s.u-tokyo.ac.jp/~robin/jGL>

2. 関連研究

ストリーミングメッシュに関するメッシュ簡略化の方法には種々のアルゴリズム [4] [6] [8] [11] [13] [16] [17] がある。それぞれの方法は概して2つのカテゴリーに分類できる。1つのカテゴリーは少ないデータ量で圧縮された3次元モデルを提供する。もう1つのカテゴリーは非損失にオリジナルの3次元モデルを再構成できる方法である。この節でオリジナルの3次元モデルを再構成できる後者の方法を簡単に紹介する。

PM (Progressive Meshes) という方法は 1993 年 Hoppe らの Mesh Optimization [11] と 1996 年 Hoppe 氏の Progressive Meshes [12] のアルゴリズムである。簡単にいえば、このアルゴリズムは頂点位置、トポロジー、法線ベクトル、テクスチャ座標などの属性を含んだエネルギー関数の最小値を求めるとメッシュを簡略化する。

PM の主なプロセスはエッジ崩壊というオペレーションである。そのため、このアルゴリズムは2つのループがある。1つは崩壊できるエッジを探す。もう1つは崩壊したエッジの残り頂点の位置を合わせる。そして、簡略化したメッシュのトポロジーはオリジナルのメッシュと同様であるが、頂点の位置を変更させることが可能である。

1998 年 Lee らは MAPS (Multiresolution Adaptive Parameterization of Surfaces) という方法 [13] を提案した。PM と異なって、複雑な計算であるエネルギー関数がない。また、簡略化されたメッシュの形状は変形される可能性がある。

MAPS の主なプロセスはインデペンデント・セットの探索、頂点除去と再三角化というオペレーションである。隣接しない頂点のセットはインデペンデント・セットと定義する。そのセット中のすべての頂点が除去された後、残りのホールを再三角化する。そして、簡略化したメッシュの頂点はオリジナルのメッシュの頂点のサブセット (頂点の位置は同様) であるが、メッシュのトポロジーを変更させることが可能である。

PM のエネルギー関数は非常に複雑なので、メッシュ簡略化の計算時間は掛かる。MAPS の再三角化プロセスで、メッシュのトポロジーが変更されるので、オリジナルのモデルを再構成する必要なパッチのデータ量は多い。また、3次元モデルの特徴が一緒に簡略化されるので、簡略化したメッシュの特徴が認識しない可能性もある。そのため、本論文では、メッシュ簡略化の目標はより少ないデータ量の再構成に必要なパッチと特徴も認識し易い簡略化したメッシュを高速に生成する方法を提案する。

3. メッシュの表示法

3次元モデルは通常三角形メッシュ²で表示され

² 本論文では、3次元モデルを簡略化する前に三角形

る。それぞれの三角形メッシュは3つの頂点から成る。メッシュ M は図1の式で表現される。その式の中で、 V は頂点の位置、 F は3次元モデルの三角形メッシュのトポロジー、 D は面 f と関連付けられる離散的な属性 (例: 材質特性) のセット、 S はウェッジ $w=(v_i, f)$ と関連付けられるスカラー属性 (例: 法線の (N_x, N_y, N_z) 成分) のセットである。同じスカラー属性を持っている頂点と共有する角はウェッジと定義される。そして、メッシュのジオメトリは $f_V(F)$ で表示できる。ここで、 $f_V: \mathfrak{R}^m \rightarrow \mathfrak{R}^3$ は線形対応である。

$$\begin{aligned}
 M &= (V, F, D, S) \\
 V &= \{v_i\}_{i=1}^m, v_i \in \mathfrak{R}^3 \\
 F &= \{f = \{j, k, l\} | v_j, v_k, v_l \in V\}, |F| \subset \mathfrak{R}^m \\
 D &= \{d_f | f \in F\} \\
 S &= \{s_{(v_i, f)} | i \in f\}
 \end{aligned}$$

図1: メッシュの表示法。

もし面 $\{j, k, l\}$ のエッジ $\{j, k\}$ が他の面と共有しなければ、エッジ $\{j, k\}$ は境界エッジと定義される。もしエッジ $\{j, k\}$ が境界エッジ、このエッジと隣接した面 f_i と f_r は異なった離散的な属性 (すなわち、 $d_{f_i} \neq d_{f_r}$) があり、このエッジと隣接したウェッジは異なったスカラー属性 (すなわち、 $s_{(v_j, f_i)} \neq s_{(v_j, f_r)}$ あるいは $s_{(v_k, f_i)} \neq s_{(v_k, f_r)}$) があるなら、エッジ $\{j, k\}$ はシャープ³エッジと定義される。

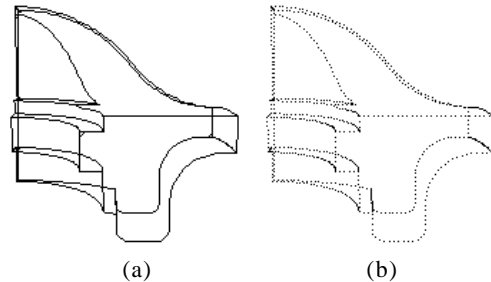


図2: (a) シャープエッジと (b) コーナー頂点の例。

もしエッジ $\{j, k\}$ が境界エッジなら、頂点 $\{j\}$ と $\{k\}$ は境界頂点と定義される。もし頂点 $\{j\}$ は境界頂点、この頂点と隣接したエッジを共有した面 f_i と f_r は異なった離散的な属性 (すなわち、 $d_{f_i} \neq d_{f_r}$) があること、この頂点と隣接したウェッジは異なったスカラー属性 (すなわち: $s_{(v_j, f_i)} \neq s_{(v_j, f_r)}$) があることなら、頂点 $\{j\}$ はコーナ

ではないメッシュを三角形メッシュに変換する。

³ ここでの「シャープ」はジオメトリのシャープだけでなく、表面の材質特性が異なったところもシャープである。

一頂点⁴と定義される。図2はシャープエッジ(a)とコーナー頂点(b)の例である。もし頂点 $\{j\}$ と $\{k\}$ は共有されるエッジ $\{j,k\}$ が存在しなければ、頂点 $\{j\}$ と $\{k\}$ はインデペンデントと定義される。

もしエッジ $\{j,k\}$ と隣接したウェッジは同じスカラー属性(すなわち、 $s_{(v_j, f_i)} = s_{(v_k, f_r)}$)および $s_{(v_k, f_i)} = s_{(v_k, f_r)}$)の場合に、2つの端点のスカラー属性の差はレベル i 、 $i \in [0, n]$ の閾値 e_i より大きいならば、エッジ $\{j,k\}$ はレベル i のベースエッジと定義される。図8の(a)~(e)は各レベルのベースエッジの例である。すべてのレベル i のベースエッジの端点、コーナー頂点およびこれらの頂点を接続する頂点のセット V_i はレベル i のベース頂点と定義される。そして、 $\{V_i\}_{i=0}^n, V_i \subset V_{i+1}, V_n = V$ 、頂点のセット V_i から成り立つメッシュ M^i はレベル i の簡略化したメッシュである。

もし V の頂点数と V_0 の頂点数の差が n' なら、オリジナルメッシュ $M^{n'} = M$ から簡略化したメッシュ M^0 まで、 n' 個のステップがある。1つのステップは1回の頂点結合というプロセスがある。逆に、簡略化したメッシュからオリジナルメッシュを再構成するとき、 n' 個の頂点分割というプロセスが必要となる。

4. メッシュの簡略化

3次元メッシュの簡略化はまずレベル $n-1$ の閾値 e_{n-1} を使って、レベル $n-1$ のベースエッジを探す。ベースエッジの端点とコーナー頂点は除去できない頂点として指定する。逆に、ほかの頂点とこの頂点と連結されたエッジは除去できる頂点とエッジである。それぞれの頂点とエッジを除去するとレベル $n-1$ の最初の簡略化したメッシュを得る。

このレベルでのベースエッジをもう一回探して、もし除去できる頂点とエッジがあれば、それぞれの頂点とエッジを除去する。このようなループを繰り返すとレベル $n-1$ の簡略化したメッシュを得る。

その後、レベル i の簡略化したメッシュを使って、レベル $i-1$ での簡略化したメッシュを生成して、レベル0に到着したならば、メッシュ簡略化を完成する。

4.1. 除去できるエッジの選択

まず、現時点のレベルのベースエッジを探す。すべてのベースエッジの端点とコーナー頂点が除

⁴ 境界頂点のセット V_B はコーナー頂点のセット V_C のサブセット(すなわち: $V_B \subseteq V_C \subseteq V$)である。また、シャープエッジの端点はコーナー頂点ではない可能性がある。

去できないというマークを付ける。図3に示されるように、除去できる頂点と隣接している除去できるエッジが1つ以上の場合、それぞれのエッジ除去の結果のテストも必要となる。そのため、除去できるエッジとベースエッジを探るとき、計算された端点のスカラー属性の差をプライオリティーキューに置く。そして、端点のスカラー属性の差が小さいエッジが優先的に除去される。また、除去できるエッジの1つの端点が除去できない場合もある。

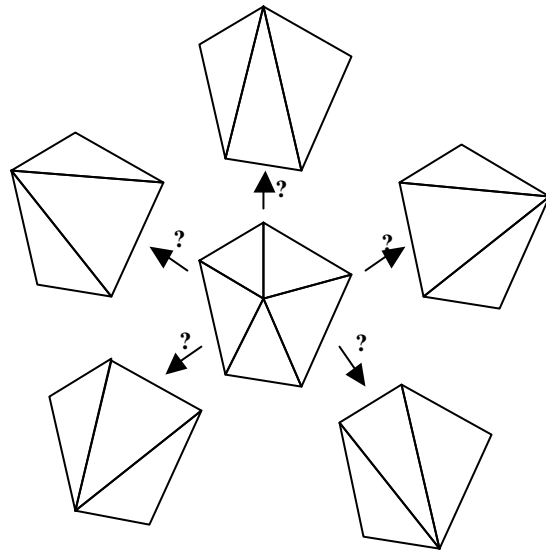


図3：真中の頂点を除去するとき、隣接している頂点と結合する予定であるが、どの頂点と結合した方がいいかのテストも必要となる。

プライオリティーキューから取った除去できるエッジの端点を試しに結合する。変形する予定の面が変形した後、図4に示されるようなフリップという問題を起さずば、これらの頂点は結合しない。

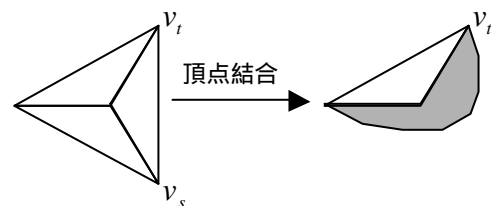


図4：頂点 v_s と頂点 v_t を結合したら、頂点 v_s が除去される。その後、下の三角形の形状が変形される。このとき、フリップという問題が起こる。

また、もしプライオリティーキューから取った除去できるエッジの端点は除去できない頂点ならば、このエッジも除去しない。もしプライオリティーキューから取ったエッジが除去できれば、4.2節で示すような頂点結合というオペレーションが施される。

4.2. 頂点結合

頂点を結合する前に、除去される頂点の情報をオリジナルの形状を再構成するために保存する。除去する予定の頂点を共有する面を変形する。図5に示されるように、頂点 v_s と頂点 v_t を結合した後、エッジ $\overline{v_a v_s}$ とエッジ $\overline{v_s v_d}$ 下の3つの三角形の形状が変形される。

最後に除去する予定の面と頂点が除去される。また、頂点結合によって変形された面の頂点は除去できないというマークを付ける。例えば、図5の v_a 、 v_b 、 v_c 、 v_d 、 v_t である。そして、同じループで除去された頂点はインデペンデントの頂点である。

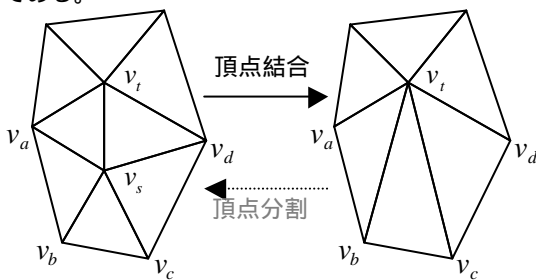


図5：頂点結合と頂点分割の例。

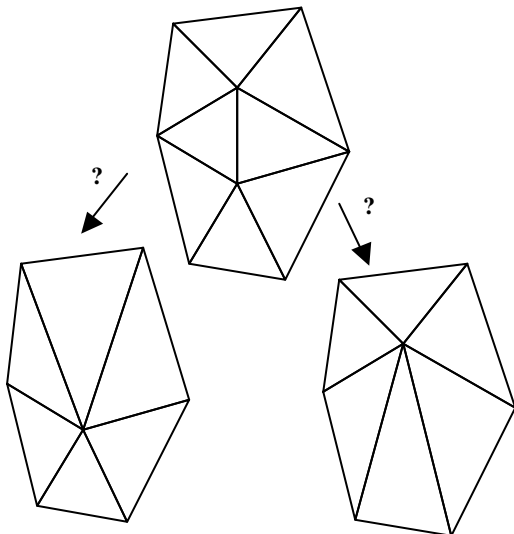


図6：真中のエッジを除去するとき、どの端点を除去した方がいいかのテストも必要となる。

オリジナルの形状を再構成するためのパッチ情報のデータ量を小さくするために、2つの頂点を結合するとき、1つの頂点の位置は変更しないと、もう1つの頂点が除去される。そのため、図6に示されるように、2つの頂点除去の結果のテストも必要となる。ここで、図5のように、もし頂点 v_t が除去できない頂点、頂点 v_t と頂点 v_t との隣接した頂点の曲率は頂点 v_s と頂点 v_s との隣接した頂点の曲率より大きいならば、頂点 v_s を除去する。

5. メッシュの再構成

簡略化したメッシュと簡略化を行ったとき保存したパッチ情報はストリーミングメッシュの構成である。インターネットでストリーミングメッシュを利用するとき、最初に簡略化したメッシュを伝送する。その後、それぞれのパッチ情報を伝送して、5.2節のような頂点分割というオペレーションを実行して、オリジナルの3次元モデルを非損失に再構成する。

5.1. パッチの構造

簡略化したメッシュからオリジナルのメッシュを再構成するとき、 n' 個のステップがあれば、 n' 個のパッチが必要となる。そして、パッチの大きさはインターネットでの伝送時間との関係が深い。メッシュを簡略化するとき、ステップ $i+1$ のメッシュがステップ i のメッシュになるプロセスは頂点結合というオペレーションである。逆に、メッシュを再構成するとき、ステップ i のメッシュがステップ $i+1$ のメッシュになるプロセスは頂点分割というオペレーションである。そのため、伝送したパッチは追加する頂点の位置、属性と頂点分割の必要な情報がある。

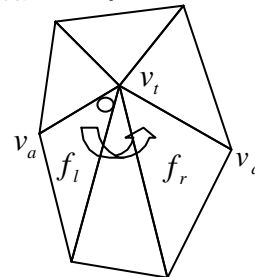


図7：伝送したパッチの情報を使って、分割予定の頂点の位置と変形予定の三角形が探せる。

頂点分割の必要な情報は分割予定の頂点の位置が探せる情報である。図7の v_t は分割予定の頂点であり、エッジ $\overline{v_a v_t}$ とエッジ $\overline{v_t v_d}$ 下の3つの三角形の形状も変形予定である。そのため、分割予定の頂点と変形予定の三角形を探すことが必要となる。そして、1つのパッチでは追加する頂点の位置、属性と三角形 f_l 、頂点 v_t と三角形 f_l から三角形 f_r までの三角形数を保存するだけである。

5.2. 頂点分割

ステップ i のメッシュがステップ $i+1$ のメッシュになるとき、伝送した5.1節のようなパッチを使って、分割予定の頂点を決定する。この頂点を分割して増加する頂点の位置と属性は同じく伝送したパッチに含まれる。図5のように、頂点 v_a 、 v_d 、 v_t を決定すると頂点 v_s とエッジ $\overline{v_s v_t}$ を生成した後、エッジ $\overline{v_a v_s}$ とエッジ $\overline{v_s v_d}$ の下の三角形の形状が

変形される。

6. 結果

図 8 は 3 次元モデル bunny のオリジナルのレベルから簡略化したレベルまでの各レベルでのベースエッジと簡略化したメッシュである。図 9 は他の 3 次元モデルの簡略化したメッシュ (b) とオリジナルの 3 次元モデル (a) の比較である。粗いメッシュでも形状と特徴が認識できる。表 1 は 3 次元モデルの面数、頂点数と簡略化した実行時間である。形状と特徴が認識できる粗いメッシュを生成するので、形状と特徴が多いメッシュの圧縮率 (例: hand) は他のモデルより小さいくなる。

Model	Original mesh		Simplified mesh		Time (ms)
	#faces	#vertices	#faces	#vertices	
bunny	2915	1494	531	302	681
dragon	2730	1257	1704	744	540
hand	2130	1055	1852	916	370
cessna	13546	6795	5720	2882	5040
fandisk	12946	6475	1518	761	6477

表 1: 簡略化したメッシュの面数、頂点数と実行時間。実験するプラットフォームの CPU は Intel Mobile Pentium III 850MHz、メモリは 256MB、Java の環境は Sun Java™2 SDK, Standard Edition v 1.3.1。

表 2 はストリーミングメッシュの実際のファイルサイズである。比較するために、オリジナルのモデルも同じファイル形式 (圧縮したアスキーファイル) を変更した。なお、個々のパッチファイルのサイズは小さいので、圧縮しないアスキーファイルで提供される。また、オリジナルのモデルを再構成するに必要なパッチ数は表 1 での頂点数の差である。

Model	file size of M^n (bytes)	file size of M^0 (bytes)	file size of patches
bunny	73238	13984	33.451
dragon	63811	37328	34.058
hand	50715	42960	33.856
cessna	528002	183435	25.653
fandisk	280979	63345	25.089

表 2: 簡略化したメッシュ M^0 、オリジナルのメッシュ M^n とパッチのファイルサイズ。また、ここでのパッチファイルサイズはパッチファイルの平均サイズ (バイト) である。

オリジナルのモデルを再構成するに必要なパッチでの保存された情報は除去した頂点とこの頂点のトポロジーである。本研究のパッチのトポロジー位置の情報は PM と同様であるが、頂点の情報は PM の半分である。MAPS のパッチの頂点の情報は本研究と同様であるが、トポロジー位置の情報は PM より多い。

また、すべてのアルゴリズムは Java のみで開発されたので、ウェブ上⁵でオンライン実行すること

も可能である。

7. おわりに

本研究はインターネット伝送するためのストリーミングメッシュを開発した。このストリーミングメッシュを利用する場合、最初に少ないデータ量で、形状と特徴が認識できる粗いメッシュを伝送する。その後、オリジナルの形状を再構成するために必要なパッチ情報を伝送して、オリジナルの 3 次元モデルを非損失に再構成できる。

ストリーミングメッシュの開発は主な 3 つの部分から成る: 3 次元メッシュの簡略化、簡略化したメッシュとオリジナルの形状を再構成するための必要なパッチ情報の保存、ストリーミングメッシュの伝送とオリジナルの形状の再構成である。また、ストリーミングメッシュの簡略化したメッシュでも形状と特徴が認識できる。

頂点結合と頂点分割という簡単なオペレーションは 3 次元モデルの簡略化と再構成の主なプロセスである。そして、簡略化したメッシュの頂点はオリジナルのメッシュの頂点のサブセット (頂点の位置も同様) であり、メッシュのトポロジーも同様である。なお、オリジナルの 3 次元モデルを再構成するのに必要なパッチのサイズは小さいので、インターネットでの伝送に適している。さらに、複雑な計算の必要がないので、高速にストリーミングメッシュを生成できる。

謝辞

本研究で使用している 3 次元モデルは Georgia Tech. の Large Models Archive⁶ と Microsoft Research の Hoppe 氏が公開した FTP サイト⁷ からダウンロードした。皆様に感謝いたします。

参考文献

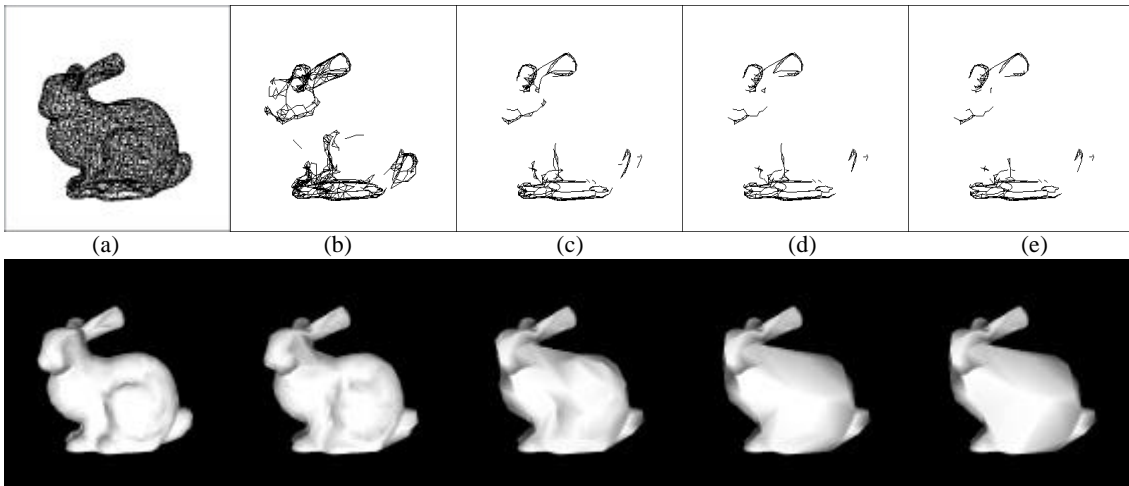
- [1] The Source for Java™ Technology. Sun Microsystems, Inc., <http://java.sun.com>, 2001.
- [2] P. Alliez and M. Desbrun. Progressive Compression for Lossless Transmission of Triangle Meshes. In *ACM SIGGRAPH 2001 Conference Proceedings*, pages 195-202, 2001.
- [3] B.-Y. Chen and T. Nishita. jGL and its Applications as a Web3D Platform. In *ACM Web3D 2001 Conference Proceedings*, pages 85-91, 2001.
- [4] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks and W. Wright. Simplification Envelopes. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 119-128, 1996.
- [5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. In *ACM SIGGRAPH 95 Conference Proceedings*, pages 173-182, 1995.
- [6] M. Garland and P. S. Heckbert. Simplifying Surfaces with Color and Texture Using Quadric Error Metrics. In *IEEE Visualization 98 Conference Proceedings*, pages 263-269, 1998.

⁵ <http://nis-lab.is.s.u-tokyo.ac.jp/~robin/jSM>

⁶ http://www.cc.gatech.edu/projects/large_models

⁷ <ftp://ftp.research.microsoft.com/users/hhoppe/data>

- [7] M. Garland and P. S. Heckbert. Surface Simplification Using Quadric Error Metrics. In *ACM SIGGRAPH 97 Conference Proceedings*, pages 209-216, 1997.
- [8] P. S. Heckbert and M. Garland. Survey of Polygonal Surface Simplification Algorithms. In *ACM SIGGRAPH 97 Conference, Multiresolution Surface Modeling Course Notes*, 1997.
- [9] H. Hoppe. New Quadric Metric for Simplifying Meshes with Appearance Attributes. In *IEEE Visualization 99 Conference Proceedings*, pages 59-66, 1999.
- [10] H. Hoppe. Efficient Implementation of Progressive Meshes. In *Computer & Graphics*. Vol. 22, No. 1, pages 27-36, 1998.
- [11] H. Hoppe. Progressive Meshes. In *ACM SIGGRAPH 93 Conference Proceedings*, pages 99-108, 1996.
- [12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Mesh Optimization. In *ACM SIGGRAPH 93 Conference Proceedings* pages 19-26, 1993.
- [13] A. Lee, W. Sweldens, P. Schoröder, L. Cowsar and D. Dobkin. MAPS: Multiresolution Adaptive Parameterization of Surfaces. In *ACM SIGGRAPH 98 Conference Proceedings*, pages 95-104, 1998.
- [14] R. Ronfard and J. Rossignac. Full-range Approximation of Triangulated Polyhedra. In *Computer Graphics Forum (EUROGRAPHICS 96 Conference Proceedings)*, Vol. 15, No. 3, pages 67-76, 1996.
- [15] P. Sander, J. Snyder, S. Gortler and H. Hoppe. Texture Mapping Progressive Meshes. In *ACM SIGGRAPH 2001 Conference Proceedings*, pages 409-416, 2001.
- [16] W. I. Schroeder, J. A. Zarge and W. E. Lorensen. Decimation of Triangle Meshes. In *ACM Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, Vol. 26, No. 2, pages 65-70, 1992.
- [17] G. Turk. Re-tiling Polygonal Surfaces. In *ACM Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, Vol. 26, No. 2, pages 55-64, 1992.



Level 16: $f = 2915$ Level 12: $f = 1145$ Level 8: $f = 669$ Level 4: $f = 549$ Level 0: $f = 531$
 図8 : 3次元モデル bunny の各レベルの (a) ~ (e) ベースエッジと (f) ~ (j) 簡略化したメッシュ。

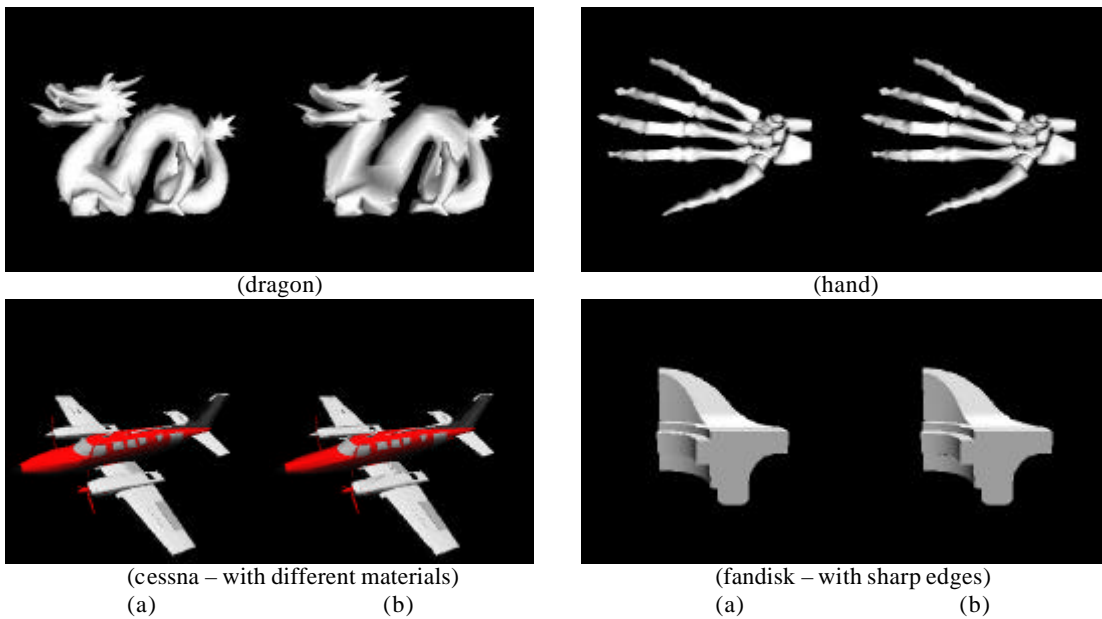


図9 : (a) オリジナルの3次元モデル、(b) 簡略化したメッシュ