

QoS を目指した 3 次元モデルの多重解像度ストリーミングメッシュ法の開発

6 U - 0 4

陳 炳宇 西田 友是
 東京大学

1. はじめに

近年、インターネットで効率的に 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があるとは限らない。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。その他、インターネットの実質上のネットワークバンド幅は安定ではないので、どの程度のデータ量のモデルを伝送するかが問題である。そのため、本研究ではインターネット伝送のための QoS (Quality of Service) を目指した多重解像度ストリーミングメッシュの伝送法を開発した。このシステムでは、最初に実質上のネットワークバンド幅に適した少ないデータ量で、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、高精度なモデルが必要となる場合に、段階的にいくつかの必要な情報を追加伝送して、高精度なモデルを提供できる。また、すべての追加情報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。

2. メッシュの表示法

3 次元モデルは通常三角形メッシュにより表示される。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、 m は頂点数、 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) d_f 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) $s_{v_i, f}$ のセットである。ここで、頂点 v_i に隣接する面 f の 1 つのコーナー $\{v_i, f\}$ はウェッジと定義される。

もし、ある面 $\{j, k, l\}$ のエッジ $\{j, k\}$ が他の面と共有しなければ、そのエッジは境界エッジと定義される。もし、あるエッジ $\{j, k\}$ が境界エッジであり、このエッジと隣接する 2 つの面 f_l と f_r が異なる離散的な属性 (すなわち、 $d_{f_l} \neq d_{f_r}$) をもち、こ

のエッジに隣接するウェッジが異なるスカラー属性 (すなわち、 $s_{v_j, f_l} \neq s_{v_j, f_r}$ あるいは $s_{v_k, f_l} \neq s_{v_k, f_r}$) をもつならば、そのエッジはシャープエッジと定義される。

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

図 1: メッシュの表示法。

もし、あるエッジ $\{j, k\}$ の端点が両方ともに、それに隣接したウェッジのスカラー属性が同じ (すなわち、 $s_{v_j, f_l} = s_{v_j, f_r}$ または $s_{v_k, f_l} = s_{v_k, f_r}$) である場合、その両端点のスカラー属性の差がそのエッジの重みと定義される。もし、その重みがある閾値よりも大きいならば、そのエッジはベースエッジと定義される。

3. メッシュの簡略化と再構成

3 次元メッシュの簡略化はまず、シャープエッジを探す。次に、閾値を用いて、ベースエッジを探す。そして、シャープエッジとベースエッジの端点を除去できない頂点として指定する。その他の頂点と、これらの頂点と連結したエッジは、除去できる頂点とエッジである。また、ベースエッジを探すとき計算したエッジの重みをプライオリティーキューに置く。そうすることで、小さい重みがあるエッジから優先的に除去されることができる。その次に、除去できるエッジの両端点を試しに結合してみる。もし問題がなければ、そのエッジの除去の情報を、再構成するのに必要なパッチとして保存する。そして、図 2 のように除去できるエッジの端点を結合して、近傍のローカルな三角形を変形する。除去可能な頂点とエッジを除去することで、最初の簡略化したメッシュが得られる。再び、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、それらをそれぞれ除去する。このようなループを繰り返すことで、メッシュ簡略化は完成する。

The Development of a QoS Oriented Multi-resolution Streaming Mesh for 3D Models

Bing-Yu Chen and Tomoyuki Nishita

The University of Tokyo

E-mail: {robin, nis}@is.s.u-tokyo.ac.jp

Web: <http://nis-lab.is.s.u-tokyo.ac.jp/~{robin, nis}/>

¹ 本論文では、オリジナルのモデルを再構成する必要な追加情報をパッチと呼ぶ。

メッシュを簡略化するとき、図2のような頂点結合というオペレーションを用いて頂点が1つ減少したメッシュになる。逆に、頂点を1つ増加したメッシュにする場合には、伝送したパッチにより、分割する頂点が分かって、図2のような頂点分割というオペレーションを用いる。また、ある頂点を分割して追加する頂点の、位置と属性はそのパッチに含まれている。

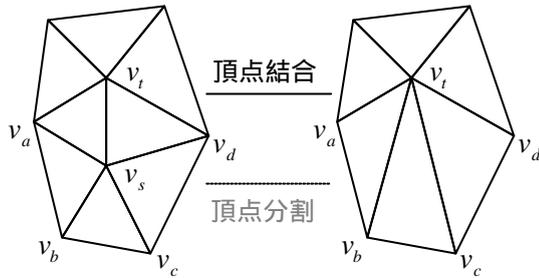


図2：頂点結合と頂点分割の例。

4. QoS を目指した伝送法

インターネットでストリーミングメッシュを伝送するとき、それぞれのパッチを個別に伝送すれば、ユーザーはすぐもっと細かいメッシュの形状が見えるが、ネットワークオーバーヘッド[1]が大きくなるので、オリジナルのメッシュを再構成するまでの時間がかかる。逆に、すべてのパッチをカプセル化して伝送する場合に、ネットワークオーバーヘッドが小さくなるが、カプセル化されたパッチをダウンロードするには時間がかかる。また、ユーザー側とサーバーの間の実質上のネットワークバンド幅が未知なので、あらかじめ伝送予定のパッチ数を設定することもできない。

そのため、ネットワーク分野の QoS[2]の概念を参考にして、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、実質上のネットワークバンド幅によって適した異なった解像度の3次元モデルのメッシュデータをサーバーから提供する。このシステムでは、最初に簡略化されたメッシュを伝送して、同時にサーバーとクライアントの間の実質上のバンド幅を計算する。その後、計算されたバンド幅によって適したパッチ数を伝送する。そして、ユーザーはバンド幅によって、異なった解像度の3次元モデルを得られる。また、伝送予定の異なった解像度の3次元モデルがランタイムに生成されるので、あらかじめサーバー上で異なった解像度の3次元モデルを別々に保存する必要がない。

5. 結果

図3は3次元モデル bunny のオリジナルのメッシュ、簡略化された結果と再構成中の中間メッシュである。簡略化と圧縮に要した時間の合計は 681ms である。実験したプラットフォームの CPU は Intel

Mobile Pentium III 850MHz、開発言語は Java のみである。また、Java の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 jGL[3]を利用している。

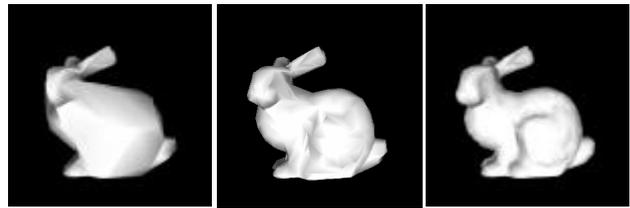


図3：3次元モデル bunny の(a)オリジナルのメッシュ（面数：2915、頂点数：1494）、再構成中の中間メッシュ（面数：1145、頂点数：609）と(c)簡略化された結果（面数：531、頂点数：302）。

モデル	伝送時間 (ms)
(1)オリジナルのメッシュ	10,956
(2)簡略化されたメッシュ	2,143
(3)図3の(b)まで	4,545
(4)図3の(c)まで	11,182

表1：PHS(64K)の回線を用いて、(1)オリジナルと(2)簡略化されたメッシュの伝送時間の比較。また、(3)図3の(b)と(4)図3の(c)までの伝送時間も測定した。

回線の種類	伝送された3次元モデル		
	時間(ms)	面数	頂点数
100BaseT	195	2,915	1,494
10BaseT	761	1,707	890
PHS(64K)	2,143	531	302

表2：回線によって異なった3次元モデルの伝送例。

表1と表2は多重解像度ストリーミングメッシュと QoS を目指した伝送法の実験結果である。100BaseT の回線を利用しているユーザーは最初にオリジナルのメッシュと同様なモデルを得られる。64K の PHS を利用しているユーザーは最初に一番粗いメッシュを得る。10BaseT の回線を利用しているユーザーは最初に適した簡略化されたメッシュが得られる。

参考文献

- [1] RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. The Internet Society, 1999.
- [2] White Paper - Introduction to QoS Policies. Stardust Technologies, Inc., 1999.
- [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] H. Hoppe. Progressive Meshes. In *ACM SIGGRAPH 93 Conference Proceedings*, pages 99-108, 1996.
- [5] 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.