グラフィクスとCAD 100－4
（2000．9．7）

細分割曲面の連続的多重解像度表現

金井 崇
慶應義塾大学 環境情報学部
〒252-8520 神奈川県藤沢市遠藤 5322
E-mail: *kanai@sfc.keio.ac.jp*

本報告は，細分割曲面の連続的多重解像度表現 (PSS 表現) について提案するものである．これはいわばプログレッシブメッシュの細分割曲面版であり，任意の解像度の細分割曲面を格納し取り出すことのできる表現である．またここでは，高密度な三角形メッシュからの簡略化手法を用いた PSS 表現への高速変換手法や，ネットワーク環境下でのプログレッシブ転送への応用について議論する．

Continuous Multiresolution Representation of Subdivision Surfaces

TAKASHI KANAI
Keio University, Faculty of Environmental Information
5322 Endo, Fujisawa-city, Kanagawa, 252-8520, Japan.
E-mail: *kanai@sfc.keio.ac.jp*

In this paper, we propose a continuous multiresolution representation of subdivision surfaces called *progressive subdivision surfaces* (PSS representation). This representation is just a 'subdivision surface' version of progressive meshes (PM representation), and thus inherits various good properties of PM representation. Here we propose an algorithm for converting PSS representation from dense triangular meshes based on QEM-based simplification. We also discuss an application to progressive transmission for broadcasting virtual environments over the network like the Internet.

1 はじめに

細分割曲面 [15] は，制御メッシュと呼ばれる多面体を規則的に分割することで滑らかな曲面形状を表現することが可能である．中でも Loop によって提案された細分割曲面 [10] (図 1) は，三角形メッシュを制御メッシュとして，任意位相の C^1 連続な曲面を定義できる．Loop 細分割曲面の制御メッシュは，三角形メッシュと同じように定義できるので，データの圧縮や転送への利用も期待できる．さらに，三角形メッシュに対してこれまで提案されている，様々なアルゴリズムを転用することもできる．

本報告は，細分割曲面の連続的多重解像度表現 (*progressive subdivision surface representation*, 以下 PSS 表現と呼ぶ) について提案するものである．ここでは，Loop 細分割曲面 (の制御メッシュ) を Hoppe により提案されたプログレッシブメッシュ表現 [5] (以下 PM 表現と呼ぶ) と同様の形式で表現するものである．本表現は，PM 表現の良い性質を受け継いでいる：すなわち，エッジ消去 / 頂点分割の繰り返し操作により，任意の解像度の細分割曲面を高速に取り出すことが可能である．これより例えば，頂点

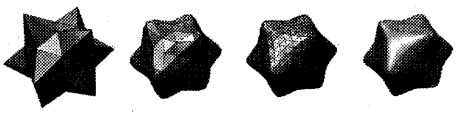

図 1: Loop 細分割曲面

数や誤差を基準として細分割曲面の粗密を調節することが可能となる．また適応的細分割により，粗密を領域毎に変えられるので，レンダリングの際にも有効である [6] .

2 PM 表現および PSS 表現

Hoppe により提案された PM 表現 [5] は，エッジ消去 (*edge collapse*) とその逆操作である頂点分割 (*vertex split*) の二つの操作 (図 2) を基本とした，任意位相の三角形メッシュに対する連続的多重解像度表現である．エッジ消去オペレータ $ecol(e)$ は，二つの隣接頂点 v_1, v_2 を一つの頂点 $\bar{v}$ に統合する位相操作を施す．初期メッシュ $\hat{M} = M^n$ に n 組のエッジ消去オ

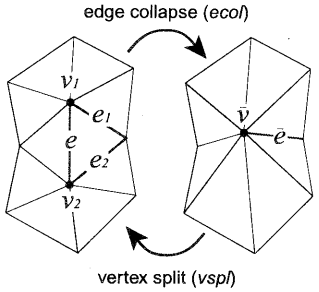

図 2: エッジ消去と頂点分割

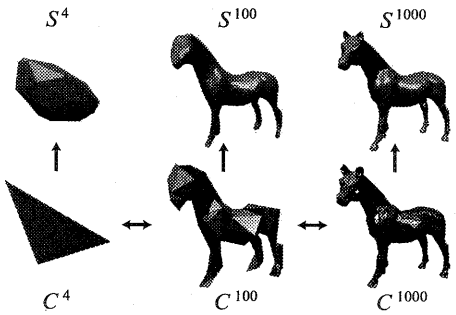

図 3: PSS 表現 (下段：制御メッシュ，上段：細分割曲面)

ペレータを適用することで，粗いメッシュ M^0 に簡略化される．一方その逆操作である頂点分割オペレータ $vspl(\hat{v})$ は，頂点 $\hat{v}$ を二つの v_1, v_2 に分割し，その間にエッジ e を挿入する位相操作を施す．メッシュ M^0 に n 組の頂点分割オペレータを適用することで，もとのメッシュ $\hat{M}$ を復元できる．まとめると，初期メッシュ $\hat{M}$ と簡略化された粗いメッシュ M^0 の間には，以下のような関係が成り立つ:

$$M^n \overset{ecol_{n-1}}{\rightarrow} \ldots \overset{ecol_1}{\rightarrow} M^1 \overset{ecol_0}{\rightarrow} M^0,$$
$$M^0 \overset{vspl_0}{\rightarrow} M^1 \overset{vspl_1}{\rightarrow} \ldots \overset{vspl_{n-1}}{\rightarrow} M^n.$$

PM 表現は $(M^0, \{vspl_0, \ldots, vspl_{n-1}\})$ で表される．PM 表現のデータ構造 [7] では，頂点の親子関係により頂点の木階層を作ることができる．これらの階層関係は，PM 表現の頂点分割の記録を一度読み込むことで，トップダウン的に生成される [6] ．レンダリング時にはこの親子関係を辿ることで，必要に応じてメッシュの粗密を変化させる選択的詳細化を，実時間で行なうことができる．

我々の提案する PSS 表現も，基本的には PM 表現と同様の表現である．細分割曲面の最も粗い制御メッシュ C^0 と，頂点分割オペレータ群 $vspl_i$ により，より詳細化された制御メッシュ C^i が定義される．個々の制御メッシュ C^i からは，細分割ステップによって細分割曲面 S^i が定義される．図 3 に PSS 表現の例を示す．任意の解像度を持つ細分割曲面は，PSS 表現による頂点の木階層を使って，その親子関係を辿ることで得られる．

レンダリング等の処理の効率化のためには，制御メッシュ C^i だけでなく，細分割曲面 S^i (実際には C^i に数回の細分割ステップを施した細分化メッシュ) を保持しておく必要がある[1]．これには，PM 表現のデータ構造 [7] において，個々の面 (Face) 毎に各細分割レベルの頂点を二次元配列もしくは四分木構造を用いて保持することで，効率的に格納できる [15]．頂点分割の際には，制御メッシュの接続性が変わるので，その度に細分割メッシュの各細分割レベルの頂点座標を更新する必要がある．ただその更新は，図 2 において，新しく生成される頂点 v_1, v_2 の近傍の面だけを対象とすれば良い．エッジ消去の場合も同様である．

3 PSS 表現の構築

本節では PSS 表現の構築方法について議論する．PSS 表現を直接ユーザが手動で生成するのは困難なので，ここでは密な三角形メッシュから PSS 表現を自動的に構築する方法を提案する．

三角形メッシュ（もしくは点群）から細分割曲面を生成する方法は過去にいくつか提案されている [8, 13] が，いずれも PSS 表現に必要な階層構造を作ることができない．我々のとった方法は，PM 表現の構築方法の多くがそうであるように，メッシュの簡略化にもとづくものである．ここでは Garland らの提案した QEM (Quadric Error Metric) による簡略化手法 [4] を拡張している．

3.1 QEM にもとづくメッシュの簡略化

初期メッシュ $\hat{M}$ の各面 f に対し，二次関数 $\mathcal{Q}^f(\mathbf{v})$ を，ある点 $\mathbf{v}$ から面 f を含む平面への距離の二乗として，$\mathcal{Q}^f(\mathbf{v}) = (\mathbf{n}^T\mathbf{v} + d)^2$ と定義する．ここで $\mathbf{n}$ は面 f の法線ベクトルである．もとのメッシュの各頂点に，その頂点に隣接した面の二次関数の，面の面積による重み付き和 $\mathcal{Q}^v(\mathbf{v})$ を割り当てる．この関数は

$$\begin{aligned}\mathcal{Q}^v(\mathbf{v}) &= \sum_{f \ni v} area(f) \cdot \mathcal{Q}^f(\mathbf{v}) \\ &= \mathbf{v}^T\mathbf{A}\mathbf{v} + 2\mathbf{b}^T\mathbf{v} + c, \qquad (1)\end{aligned}$$

のようにまとめることができる [4]．ここで $\mathbf{A}$ は 3 × 3 の対称行列，$\mathbf{b}$ は列ベクトル，そして c はスカラーである．また $area(f)$ は三角形面 f の面積を示す．面の面積による重み付けは，三角形の大きさにより QEM の値が変化することを防ぐためである [3] ．

[1] Pulli らは，細分化メッシュを保持しないですむメモリ効率の良い高速な細分割曲面のレンダリング手法を提案している [11]．しかし，制御メッシュの隣接する三角形をペアとし，四辺形として保持する必要があるため，頻繁に位相操作を必要とする本表現にそのまま適用するのは難しい．

これより，各頂点 v には $\mathcal{Q}^v = (\mathbf{A}, \mathbf{b}, c)$ を構成する $6+3+1=10$ 個の浮動少数値が貯蓄される．ここで $\mathcal{Q}^v(\mathbf{v})$ はアルゴリズムの開始時はすべて 0 となることに注意．

$\hat{M}$ の各エッジ e に対し，エッジ消去オペレータ $ecol(e)$ (図 2) 適用後の頂点 $\bar{v}$ の座標を求める．すなわち，エッジの両端点 v_1, v_2 の持つ QEM の和 $\mathcal{Q}^{\bar{v}} = \mathcal{Q}^{v_1} + \mathcal{Q}^{v_2}$ を最小にする頂点座標 $\mathbf{v}^{min}$ を見つける．これは，$\mathcal{Q}^{\bar{v}}$ の勾配 $\nabla\mathcal{Q}^{\bar{v}}$ がゼロになるときの座標であり，以下の線形方程式を解くことで得られる:

$$\mathbf{A}\mathbf{v}^{min} = -\mathbf{b}. \tag{2}$$

$\mathcal{Q}^{\bar{v}}(\mathbf{v}^{min})$ が，エッジ e のコストとなる．各エッジに発生するコストを計算し，コストの小さいものから順にエッジ消去を実行する．エッジ消去を実行した後は，$\bar{v}$ に隣接するエッジのコストを随時更新する．

3.2 頂点—エッジによる QEM の管理

Garland らの簡略化アルゴリズムの中では，QEM は頂点によって管理されている．そして，アルゴリズムの各段階で最適な頂点を求めるために，エッジ消去を施した後に新たに計算される頂点だけを評価関数に代入して評価している．ただ我々の目的では，最適な頂点の算出に，その頂点だけでなく近傍の形状 (頂点) をも評価したい．そのために，QEM の別の管理方法を用意する．

本手法での QEM の管理は，メッシュのエッジと頂点で行う．$\hat{M}$ の各エッジ e に対し，以下の式の値を持つことにする:

$$\mathcal{Q}^e = \mathcal{Q}^{f_l} + \mathcal{Q}^{f_r}. \tag{3}$$

ここで f_l, f_r は e に隣接する面を示す．アルゴリズムの開始時に各エッジ e に対し $\mathcal{Q}^e$ を算出する．アルゴリズムの途中での各エッジ消去に対し，消去されるエッジ e の持つ QEM は，新しく生成される頂点 $\bar{v}$ に格納する．よって，エッジ消去を繰り返すことにより頂点に QEM が蓄積していく．これを一般化すると，図 2 において，各エッジ消去に対し

$$\begin{cases} \mathcal{Q}^{\bar{v}} &= \mathcal{Q}^e + \mathcal{Q}^{v_1} + \mathcal{Q}^{v_2} \\ \mathcal{Q}^{\bar{e}} &= \mathcal{Q}^{e_1} + \mathcal{Q}^{e_2} \end{cases}, \tag{4}$$

という QEM の更新を行なう．

3.3 細分割点を用いた評価関数の定義

本手法では，最適な頂点座標を計算するための評価関数を，もとのメッシュの頂点ではなく，もとのメッシュから得られる曲面上の点を評価するように定義する．ただしここでは，厳密な曲面上の点ではなく，もとの制御メッシュに二回細分割ステップを施した頂点を用いる．何故ならば，制御メッシュを二回細分割すれば，極限曲面にほぼ近い形状が得られるからということと，そのほうが計算がより簡便になるからである．厳密な曲面上の点を用いることも可能 [12] であるが，任意位相の制御メッシュに対して適用するには，少なくとも二回細分割することが条件になり，また定式化がより複雑になる．

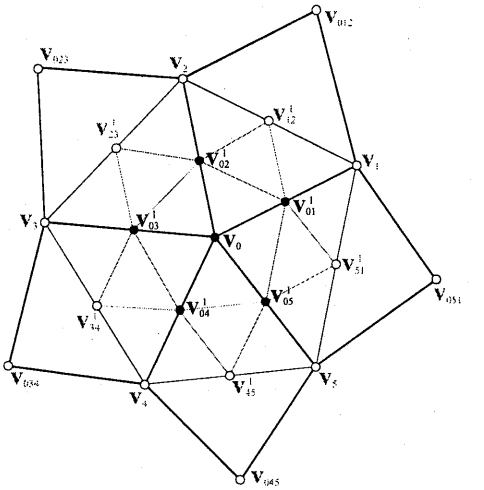

図 4: 頂点 $\mathbf{v}_0$ を中心とした細分割頂点の番号付け

ここで後の議論のために，エッジ消去後に新しく生成される頂点 $\bar{v}$ (ここでは $\mathbf{v}_0$ とする) の近傍頂点の番号付けについて，図 4 を用いて説明する．図中の頂点の記号のうち，右上添え字は細分割の回数を示す．例えば，一回細分割ステップを施した頂点の座標は，新たに生成される頂点も含めて $\mathbf{v}^{(1)}$ と表す．また，右下添え字は頂点 $\mathbf{v}_0$ に隣接する頂点やエッジに対して，半時計周りのオーダで番号付けすることを基本とする．例えば，$\mathbf{v}_j$ $(j = 1 \ldots \kappa_0)$ は，$\mathbf{v}_0$ の価数 κ_0 の 1-ring 近傍の頂点を示す．$\mathbf{v}^{(1)}_{0j}$ は，一回細分割ステップを施すことによってエッジ e_j の中間に生成される奇頂点である．この頂点は，$\mathbf{v}_0$ および $\mathbf{v}_j$ の線形式で表せる．

我々の定義する評価関数は次式のような形式で表現される:

$$\mathcal{Q}(\mathbf{v}_0) = \mathcal{Q}^{v_0}(\mathbf{v}^{(2)}_0) + \sum_j^{\kappa_0} \mathcal{Q}^{e_j}(\mathbf{v}^{(2)}_{0j}), \tag{5}$$

ここで，$\mathbf{v}^{(2)}_0$ は $\mathbf{v}_0$ に二回細分割ステップを施した頂点を，$\mathbf{v}^{(2)}_{0j}$ は $\mathbf{v}^{(1)}_{0j}$ にさらに一回細分割ステップを施した頂点を示す．式 (5) の右辺の第一項は，頂点 $\mathbf{v}_0$ 付近の形状を評価したものであり，その頂点の持つ $\mathcal{Q}^{v_0} = (\mathbf{A}_0, \mathbf{b}_0, c_0)$ が使用される．また第二項は，$\mathbf{v}_0$ に隣接するエッジ e_j の付近の形状を評価したものであり，したがって e_j の持つ $\mathcal{Q}^{e_j} = (\mathbf{A}_{0j}, \mathbf{b}_{0j}, c_{0j})$ が使用される．

最適化計算に関しては，もとのメッシュ上の頂点 $\mathbf{v}_0$ だけを変数として考える．すなわち，$\mathcal{Q}(\mathbf{v}_0)$ が最小となるような $\mathbf{v}_0$ を求めたい．そこでここでは，$\mathbf{v}_0$ に注目した式の整理を行う．$\mathbf{v}_0^{(2)}, \mathbf{v}_{0j}^{(2)}$ は以下のようにそれぞれ $\mathbf{v}_0$ の一次関数の形で表すことができる:

$$\mathbf{v}_0^{(2)} = \alpha_0 \mathbf{v}_0 + \mathbf{e}_0, \quad \mathbf{v}_{0j}^{(2)} = \alpha_{0j} \mathbf{v}_0 + \mathbf{e}_{0j}. \tag{6}$$

ここで α_0, α_{0j} は $\mathbf{v}_0$ の係数であり，$\mathbf{e}_0, \mathbf{e}_{0j}$ は，残りの頂点ベクトルすべてを集めたものである．式 (6) を用いて式 (5) を展開し整理すると，

$$\mathcal{Q}(\mathbf{v}_0) = \mathbf{v}_0^{\mathrm{T}} \bar{\mathbf{A}} \mathbf{v}_0 + 2\bar{\mathbf{b}}^T \mathbf{v}_0 + \bar{c}, \tag{7}$$

のようにまとめることができる．これは $\mathbf{v}_0$ に関する二次形式の関数となる．$\mathcal{Q}$ を最小にする頂点の座標 $\mathbf{v}_0^{min}$ は $\nabla \mathcal{Q} = 0$, すなわち

$$\bar{\mathbf{A}} \mathbf{v}_0^{min} = -\bar{\mathbf{b}}, \tag{8}$$

を解けば良く，これは式 (2) と同様線形の連立方程式となる．

3.4 アルゴリズム

PSS 表現構築アルゴリズムの手順について説明する．オリジナルの簡略化アルゴリズム [4] の若干の拡張が必要になる．

1. 初期メッシュ $\hat{M}$ の各エッジ e に対し，Q^e を計算して 10 個の係数を格納する．各頂点に対する Q^v は空にしておく．

2. $\hat{M}$ の各エッジに対し，エッジ消去したときの新しい頂点に対する最適位置 $\mathbf{v}_0^{min}$ を計算する．すなわち，式 (7) の係数 $\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{c}$ を求め，式 (8) を解く．コスト $\mathcal{Q}(\mathbf{v}_0^{min})$ を計算し，最小コストを先頭とするヒープの中に昇順に収める．

3. コストの小さい順にエッジ消去を実行し，式 (4) のルールに従って頂点，エッジのもつ $\mathcal{Q}$ を更新する．$\mathbf{v}_0$ を含むすべてのエッジのコストを更新する．

オリジナルの簡略化アルゴリズムと違うのは，一つに，頂点，エッジに格納される QEM と，最適配置計算用に使う評価関数とが異なるため，コストの評価の各段階で後者を逐次計算しなければならない点である．よってこれがオリジナルのアルゴリズムよりも計算時間が余計にかかる主な要因となる．またもう一つに，頂点に加えてエッジにも QEM を保持するため，メモリ消費量が大幅に増加することが挙げられる．エッジの数は頂点の数に比べおよそ 3 倍であるから，単純な計算ではオリジナルに比べ約 4 倍のメモリを消費することになる．

他の簡略化手法を見てみると，例えば Lindstrom らの方法 [9] では，QEM を保持せず簡略化途中の形状から QEM を直接計算することで，メモリ消費量の効率化を図っている．しかしながら，本手法においてこの方法を取ることは困難である．何故ならば，簡略化の途中のメッシュは細分割曲面の制御メッシュであり，もとの形状からはかけ離れた形状となっている．そのため，簡略化の途中の段階の形状を誤差評価することができない．

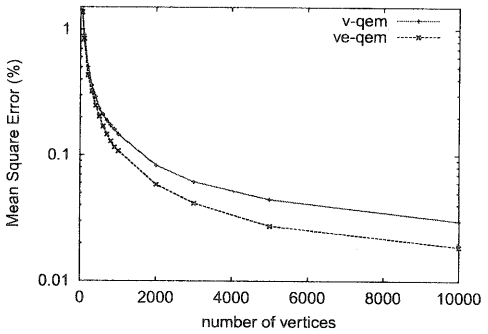

図 6: 細分割曲面の頂点数毎のもとのメッシュに対する誤差

3.5 生成結果

図 5 に，スタンフォード兎モデルに対する本手法の適用例を示す．図 5(b)-(c) はオリジナルの簡略化アルゴリズムにより得られる簡略化メッシュを，図 5(d)-(e) にはオリジナルの簡略化アルゴリズムで用いられている QEM の管理手法（頂点による管理）および評価関数により得られる制御メッシュを，そして図 5(f)-(g) では本手法で提案している QEM の管理方法（頂点－エッジによる管理）および評価関数による制御メッシュを示す．計算時間は，PentiumIII 500MHz で図 5(b)-(c) の簡略化メッシュの生成に約 1 分，制御メッシュの生成に約 2 分かかっている．制御メッシュの生成の方が簡略化メッシュの生成に比べて約 2 倍の計算時間を要するのは，最適計算におけるマトリクス $\bar{\mathbf{A}}$ とベクトル $\bar{\mathbf{b}}$ の算出の際に，計算する頂点の数が増えることによるオーバーヘッドが大きいことが原因である．

図 6 は，細分割曲面のもとのメッシュに対する誤差を示したグラフである．制御メッシュ作成の過程において，いくつかの頂点数に対し誤差を計算してプロットしている．横軸には頂点数を表示し，縦軸には誤差をメッシュの大きさに対する % で対数表示している．誤差評価ツールとして IRI-CNR Metro tool[2] を用い，その中の平均自乗誤差 (L^2−norm) を表示している．グラフには，オリジナルの QEM 管理方法 (v-qem) および本手法による管理手法 (ve-qem) の二つの結果を載せている．

二つの方法を比較すると，本手法で提案している頂点－エッジ管理方法による結果の方が，同じ頂点数の

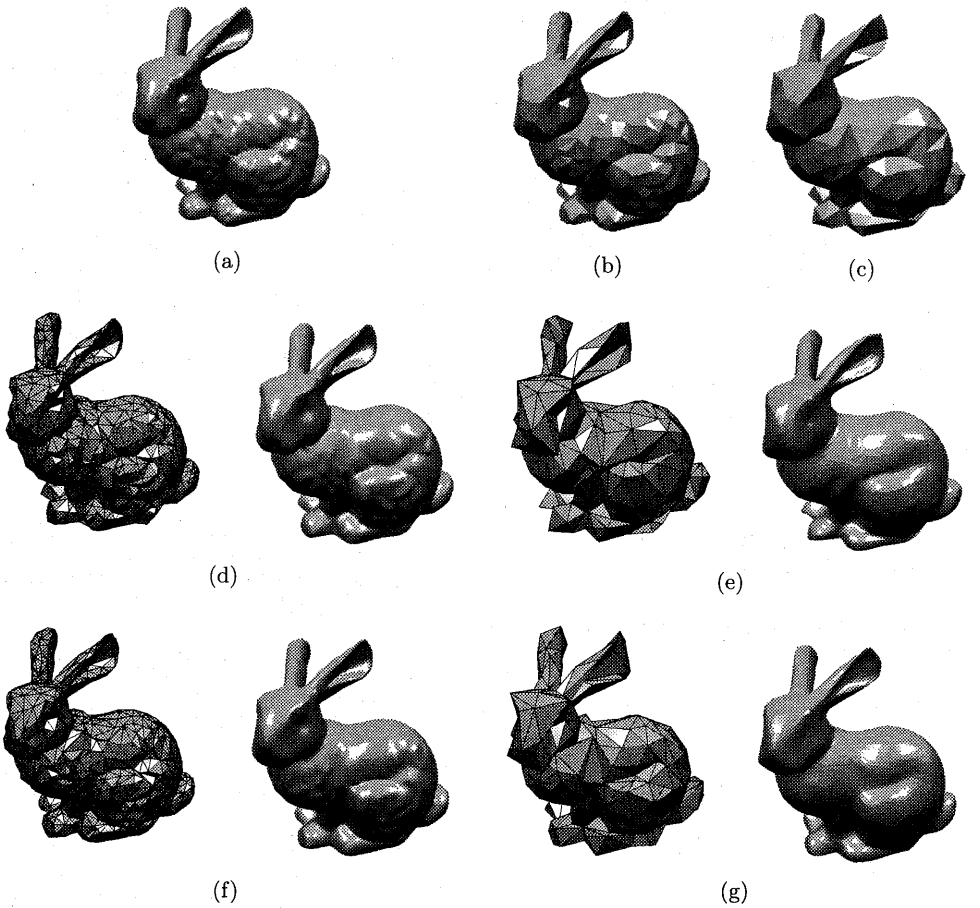

図 5: PSS 表現の構築例. (a) スタンフォード兎モデル (頂点数 35,947, 面数 69,451). (b)-(c) Garland らの簡略化アルゴリズム [4] による簡略化メッシュ ((b) 頂点数 1,000, (c) 頂点数 300) . (d)-(e) 頂点による QEM の管理 (v-qem) にもとづく制御メッシュ ((d) 頂点数 1,000, (e) 頂点数 300). (f)-(g) 頂点—エッジによる QEM の管理 (ve-qem) にもとづく制御メッシュ ((d) 頂点数 1,000, (e) 頂点数 300). なお (d)-(g) において, 左側に制御メッシュ, 右側に細分割曲面 (細分割メッシュ) を示す.

細分割メッシュに対する誤差が小さいことがわかる. このことから, 本手法により, もとのメッシュに対してより高い近似精度で細分割曲面を生成できることが確認できる. ただ, 頂点が非常に少なくなると (<100), 誤差が急激に大きくなるためにその優位性はそれほど目立たない.

4 プログレッシブ転送に関する考察

ここでは, PSS 表現の利用法の一つである, プログレッシブ転送 (*progressive transmission*) への応用について論じる. プログレッシブ転送は, クライアント／サーバー方式の分散型システムでの 3D モデルの表示に有効である. インターネットなどのネットワークで繋がれたサーバーの仕事としては, まずモデルの多重解像度表現を用意する. またもう一つには, クライアントの要求に対して, まず最初に最も粗いモデルを転送し, その後モデルの詳細化に関する記録を順々に転送する. クライアントは, 送られて来た情報をもとにモデルを再構築し表示する. このような技術は, 画像に関してはすでにプログレッシブ JPEG 等で一般的に利用されている.

PSS 表現は, 細分割曲面を多重解像度として階層的に表現したものであるが, 構築方法 (3節) で述べたように, その出発点は三角形メッシュとなる. 従って, PSS 表現は三角形メッシュに関するプログレッシブ転送に利用できる. すなわち, 最初に最も粗い制御メッシュ C^0 を送り, その後一連の頂点分割の記録 $vspl_i$ を転送する. すべてのデータを送ると, 一つの完全な三角形メッシュデータ $\hat{M}$ が得られる.

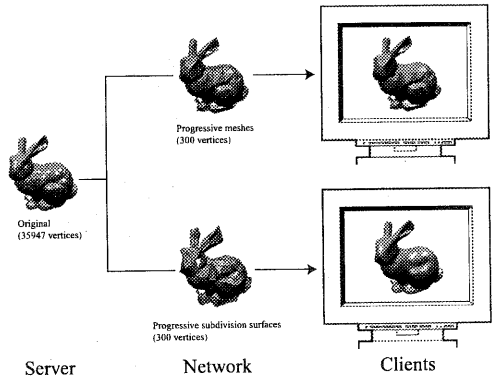

図 7: PM 表現と PSS 表現によるプログレッシブ転送

ここで，3D モデル（ここでは一般的な任意位相の三角形メッシュを考える）のプログレッシブ転送について，その方式の主な評価事項として以下の 5 つを挙げ，本表現の特徴を述べる．

可逆性 データがすべて転送されたとき，もとのモデルを完全に復元できる．

均一性 詳細化記録の小単位がすべて同一種類，同一量のデータである．

頑健性 データ転送の途中に損失があった場合（例えば，回線が途中で途切れた場合など）モデルを（不完全ながらでも）復元できる．

適応性 データを転送する方法（例えば順番や一部転送など）に大きな自由度を持つ．

視認性 データ転送の途中でももとのモデルに関する情報が視覚的に認識できる．

このうち上の 4 つの項目について，PSS 表現 は PM 表現と同じ特徴を持つ．

視認性に関しては，一般に，転送されたデータをもとに粗いモデルを詳細化する時，最初の方が形状の変化が劇的である．よって，データ量の少ない時に，いかにもとのモデルにより近い形状が再現できるかが重要である．PSS 表現では，転送途中の形状はもとのモデルを近似した細分割曲面として定義できる．図 7 にあるように，同じ量のデータを転送したとき，PSS 表現の方が PM 表現に比べ表現力が高い分，より視認性が高いと言える．特に，転送したデータ量の少ない時ほどその効果は大きく，低バンド幅の回線を介した表示に関して有効である．ただし，クライアント側でのモデル再構築の際，細分割曲面を構築し表示するので，その負担が PM 表現よりもより大きくなる．

5 おわりに

本報告では，細分割曲面の連続的多重解像度表現 (PSS 表現) について提案した．PSS 表現に関する概説と，その構築方法，そしてプログレッシブ転送への応用について述べた．PM 表現に比べ，特に視認性に関して PSS 表現の有効性を確認した．

なお，3 節で提案した PS 表現の構築方法は，三角形メッシュからの高速かつ高精度な細分割曲面変換手法としても位置付けることができる．今後の課題としては，境界や色などの属性情報が付加されたときの処理方法や，区分平滑細分割曲面 [1] への変換手法の開発が挙げられる．また，本手法をもとにした他の細分割曲面への当てはめ手法は，竹内らによって行われている [14] ．

参考文献

[1] Biermann, H., Levin, A., and Zorin, D. Piecewise smooth subdivision surfaces with normal control. In *Computer Graphics (Proc. SIGGRAPH 2000)* (2000), ACM Press, New York, pp. 113–120.

[2] Cignoni, P., Rocchini, C., and Scopigno, R. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2 (1998), 167–174.

[3] Garland, M. *Quadric-Based Polygonal Surface Simplification.* PhD thesis, Carnegie Mellon University, School of Computer Science, 1999.

[4] Garland, M., and Heckbert, P. S. Surface simplification using quadric error metrics. In *Computer Graphics (Proc. SIGGRAPH 97)* (1997), ACM Press, New York, pp. 209–216.

[5] Hoppe, H. Progressive meshes. In *Computer Graphics (Proc. SIGGRAPH 96)* (1996), ACM Press, New York, pp. 99–108.

[6] Hoppe, H. View-dependent refinement of progressive meshes. In *Computer Graphics (Proc. SIGGRAPH 97)* (1997), ACM Press, New York, pp. 189–198.

[7] Hoppe, H. Efficient implementation of progressive meshes. *Computers & Graphics 22*, 1 (1998), 27–36.

[8] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. Piecewise smooth surface reconstruction. In *Computer Graphics (Proc. SIGGRAPH 94)* (1994), ACM Press, New York, pp. 295–302.

[9] Lindstrom, P., and Turk, G. Evaluation of memoryless simplification. *IEEE Trans. Visualization and Computer Graphics 5*, 2 (1999), 98–115.

[10] Loop, C. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[11] Pulli, K., and Segal, M. Fast rendering of subdivision surfaces. In *Rendering Techniques '96 (Proc. 7th Eurographics Workshop on Rendering)* (1996), Springer-Verlag, Wien, New York, pp. 61–70.

[12] Stam, J. Evaluation of Loop subdivision surfaces. In *SIGGRAPH 99 Course Notes No.37 "Subdivision for Modeling and Animation"*. ACM SIGGRAPH, 1999.

[13] Suzuki, H., Takeuchi, S., Kanai, T., and Kimura, F. Subdivision surface fitting to a range of points. In *Proc. 7th Pacific Graphics International Conference (Pacific Graphics '99)* (1999), IEEE CS Press, Los Alamitos, CA, pp. 158–167.

[14] Takeuchi, S., Kanai, T., Suzuki, H., Shimada, K., and Kimura, F. Subdivision surface fitting using QEM-based simplification. 8th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2000), 2000. to appear.

[15] Zorin, D., and Schröder, P. Subdivision for modeling and animation. In *SIGGRAPH 2000 Course Notes.* ACM SIGGRAPH, 2000.