2

表示・編集・圧縮のための多重解像度表現技術

金井　崇　　慶應義塾大学環境情報学部
kanai@sfc.keio.ac.jp
http://www.sfc.keio.ac.jp/~kanai/

■ネットワーク時代の新たな形状表現■

前稿に登場した1体約100万個の面を持つメッシュモデルであるHappy Buddhaを100体並べ，パーソナルコンピュータ(PC)上で表示して見たいと思ったとき，一体どのようにすればよいのだろうか？　まともに表示しようと思うならば，面の数は全部合わせて約1億ポリゴンになる．よって現存のPCでは，よほど大容量の主記憶(メモリ)を積まない限り，表示することでさえおぼつかないであろう．たとえ表示できたとしても，それをさまざまな視点から見たり動かしたりするのは，ものすごく気の遠くなるような作業となるに違いない．

しかしたとえば，視点から遠く離れたところにあるものに関しては，それほど多くの面の数は必要なく，より少ない数で簡単にしたHappy Buddhaでも事足りてしまう．このように，モデルの解像度を段階的に変えることを，詳細化レベル(level of detail, LOD)とか，多重解像度(multiresolution)と呼んでいる．初期のLODでは，あらかじめ複数の解像度のモデルを用意しておき，遠近に応じてモデルを置き換えるという方法がとられてきた．このLODをより進化させた考え方として，「それなら，複数の解像度が得られるように，モデルの表現そのものを変えてしまおう」という考えから生まれたのが，モデルの多重解像度表現(multiresolution representation)である．

この新しい表現方法は，1990年代半ばに最初のブレイクスルーが起こって以来，コンピュータグラフィクス(CG)および関連する分野に確実に食い込むほどの勢いを放っている．CGやCAD/CAMモデリングでは，これらの表現を利用したモデル編集・変形技術が次々と生み出されている．また，大容量データの処理を必須とする地理情報システム(GIS)や医療情報の分野，または，限られた装置の中で最大限のパフォーマンスを要求されるコンピュータゲームなどの分野でもとみに注目されている．

最近では特に，インターネットの急速な発展と相まって，ネットワーク時代の新たな形状表現としても広まりつつある．VRML/Web3DやMPEG4など，ネットワーク環境下での利用を想定した標準化の動きは，3Dモデルの圧縮や転送，それに著作権保護目的の透かしの埋め込みなど新しい技術を創出した．その中で，多重解像度表現はこれらの技術と大変相性が良いことから，基盤技術の1つとしても確立しつつある．

ここでは，モデルの代表的な表現である三角形メッシュ(以下メッシュと呼ぶ)の多重解像度表現とその基礎技術について，もう少し詳しくみていくことにする．

■多重解像度表現のための基礎技術■

本章ではまず，メッシュの多重解像度表現を考える上での基礎技術として，簡略化とパラメータ化について概説する．特に簡略化技術については，多重解像度表現を構築する上で不可欠な技術である．

■メッシュの簡略化

メッシュの簡略化は元々，面の数の多いメッシュからなるべく形状の特徴を保存しつつ，面の数を削減するCGモデリング技術である．非常に多くの論文が発表されているのと，多くの商用CGソフトウェアにはすで

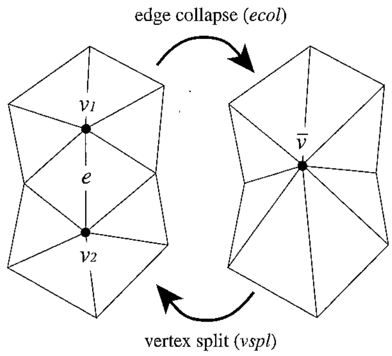

図-1 エッジ消去と頂点分割

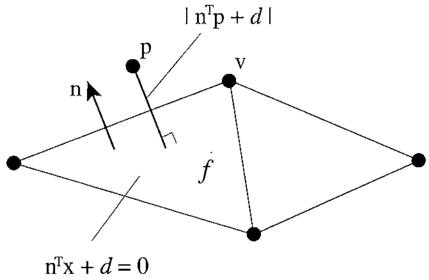

図-2 QEMの定義

に実装されていることを考えると，技術的には成熟しているといえるであろう．ここでは，その代表的ないくつかの手法について触れたい．簡略化技術を考える上での重要なファクタとして，次の3つが挙げられる．

1. 処理速度：高速なものでは，PentiumIIクラスのCPUで数万ポリゴンのメッシュの簡略化に約数秒というアルゴリズムも発表されている．
2. 品質：もとのメッシュにいかに近い形状であるかということの他に，折り目や角などの不連続線をいかに保存できるか，ということが挙げられる．
3. 属性の考慮：メッシュに付加される形状以外の属性(たとえば色やテクスチャ座標)への考慮を，アルゴリズムの中に含めるかどうか，が焦点となる．

簡略化を行うための1つのキーとなる操作が，エッジ消去(edge collapse)と呼ばれる局所的な位相操作である．これは，図-1にあるように，あるエッジeとその両端の頂点v_1，v_2の隣接する一領域において，eを消去し2つの頂点を結合して新たな頂点$\bar{v}$を作る．この操作を連続的に実行することで，メッシュの簡略化が行われるわけである．

このとき，新しく作られる$\bar{v}$の座標を求める方法が，その簡略化アルゴリズムの品質と実行速度を決める．Hoppeのとった方法[4]では，もとのメッシュの頂点をすべて保存しておき，位相操作の各段階でその近傍にある頂点座標を集め，最小自乗法により最適値を決定している．あくまで，もとのメッシュの頂点座標だけを頼りにしたわけだが，ただこの方法だと，簡略化が進むにつれて計算に必要なもとのメッシュの頂点の数が飛躍的に増えてしまう．たとえば，1つの$\bar{v}$の座標を決定するのに，数百から数千の頂点座標を集めねばならず，計算コストの増大を引き起こす原因となる．

これに対しGarlandらの提案する簡略化手法[3]では，この計算を，メッシュの各頂点における隣接面との距離の自乗和(Quadric Error Metric, QEM)の最小化に帰着することで，計算を簡素化している．図-2にあるように，1つの三角形面を含む平面は，平面の法線ベクトルnを使って，代数的に$n^Tx+d=0$と表せる．すると，ある点pとの距離は$|n^Tp+d|$で表すことができる．頂点vのまわりの面に対するこの距離の自乗を集めたものを，その頂点のQEMとして定義する．すなわち，

$$\begin{aligned} Q^v(\mathrm{v}) &= \sum_{f \ni v} area(f)\,|\mathrm{n}_f^T\mathrm{v}+d_f|^2 \\ &= \mathrm{v}^T\mathrm{A}\mathrm{v}+2\mathrm{b}^T\mathrm{v}+c, \end{aligned} \qquad (1)$$

と2次形式で表せる．ここでAは3×3の対称行列，bは列ベクトル，そしてcはスカラーである．また$area(f)$は面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．これより，各頂点vには$Q^v=(\mathrm{A}, \mathrm{b}, c)$を構成する6＋3＋1＝10個の浮動小数点値が貯蓄され，これがメッシュの形状をかわりに表現する．エッジ消去のときには，2つの頂点が持つQEM Q^{v1}，Q^{v2}を加算したものを，新しい頂点に引き継ぐ．QEMを用いることで，実行速度が飛躍的に改善されるだけでなく，品質の面でも，Hoppeらの簡略化アルゴリズムとほぼ変わりない結果が得られることが確認されている．

色情報や法線，テクスチャ座標が付加したメッシュの簡略化は，これらの属性に関する評価をQEMに付加したり，属性の不連続線(たとえばテクスチャの境界線など)付近でのエッジ消去にペナルティを施すことで対処できる[5]．

■メッシュのパラメータ化

形状表現の1つに，Bスプライン曲面等のパラメトリック曲面表現がある．詳細な説明は教科書[6]に譲るが，これらの表現では，2次元パラメータ空間上の一点(u, v)が，3次元空間への写像関数Sにより，曲面上の一点

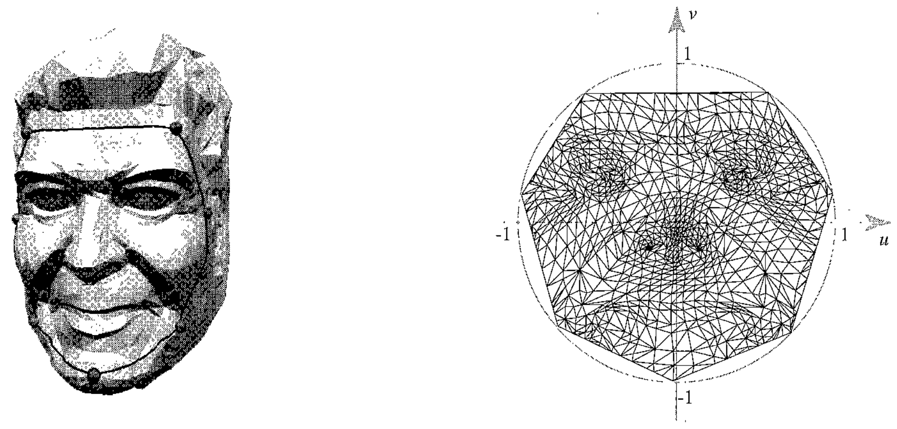

図-3　パラメータ化

$S(u, v)$として定義される.

パラメータ化とは，ちょうどこの逆写像，すなわち3次元空間上の曲面（ここではその多面体近似であるメッシュ）上の点から2次元パラメータ空間上への写像関数S^{-1}を決定するための技術である．このパラメータ化の応用範囲は非常に広い．たとえばテクスチャマッピングにおけるテクスチャ座標の計算や，メッシュから自由曲面への変換，3Dモーフィングなどへの応用が考えられる．そしてこのパラメータ化の技術も，メッシュの多重解像度表現の構築に一役買っている．

代表的なパラメータ化の例として，**図-3**にFloaterによって提案された凸結合によるパラメータ化[2)]の結果を示す．これは，図左の頭部モデルの顔の領域を，図右の2次元パラメータ空間の凸多角形上に展開したものである．このパラメータ化手法は，境界上の頂点の写像と，内部の点の写像の2つのステップにより構成される．まず，図左の球で示される頂点を図右の凸多角形の頂点として置き，続いて境界上のその他の頂点を多角形の辺上に置く．内部頂点のパラメータ座標は，ある頂点が隣接する頂点群の凸結合になるように自動的に計算される．この計算は疎の連立一次方程式を解くことで得られる．

■メッシュの多重解像度表現とその特徴■

本章では，2つの主なメッシュの多重解像度表現について，その特徴および構築方法について述べる．

■プログレッシブメッシュ

Hoppeが提案したプログレッシブメッシュ（Progressive Mesh，以下PM表現と呼ぶ）[4)]は，メッシュの多重解像度表現としては最も有名であり，その表現の緻密性と美しさから"究極のLOD"（Ultimate LOD）と称されるほどである．

Hoppeは，簡略化アルゴリズムによって利用されるエッジ消去（*ecol*）が逆操作可能であることを見つけ，これを頂点分割（vertex split, *vspl*）とした．そして，この可逆的な2つの操作（図-1）を基本とする，任意接続性の三角形メッシュに対する連続的多重解像度表現を定義した．

PM表現は最も粗いメッシュとn組の頂点分割操作を用いて（M^0, $\{vspl_0, \ldots, vspl_{n-1}\}$）として表現される．**図-4**は，あるメッシュ（頂点数49,919）をPM表現として表したものである．この図の中で，頂点数が100の最も粗いメッシュM^{100}から頂点数が49,919の最も細かいメッシュM^{49919}へ，またはその中間の任意の頂点数のメッシュへは，必要な数だけの頂点分割（あるいはエッジ消去）を使って求めることができる．たとえば，M^{100}からM^{1000}へは900回の頂点分割操作を施すことで得られる．

このPM表現を構築するには，メッシュの簡略化の際にその履歴をすべて頂点分割記録に変換して保存しておく．データを読み込む際には，最も粗いメッシュM^0とその履歴を逆順に読み込めばよい．

1回の頂点分割記録にはどの程度のデータが必要だろうか？ **図-5**において，ある階層のメッシュM^iから1回の頂点分割$vspl_i$を経て，次の階層のメッシュM^{i+1}が生成されるものとする．このとき，まずM^i側の情報として，分割する頂点$\bar{v}$の番号と，隣接する面のペア(f_1, f_2)，(f_3, f_4)が必要である．後者は，頂点分割の際にどの面とどの面の間に面が生成されるかを記録するものである．またM^{i+1}側の情報として，新しく生成される2つの頂点v_1, v_2の座標値v_1, v_2が必要となる．

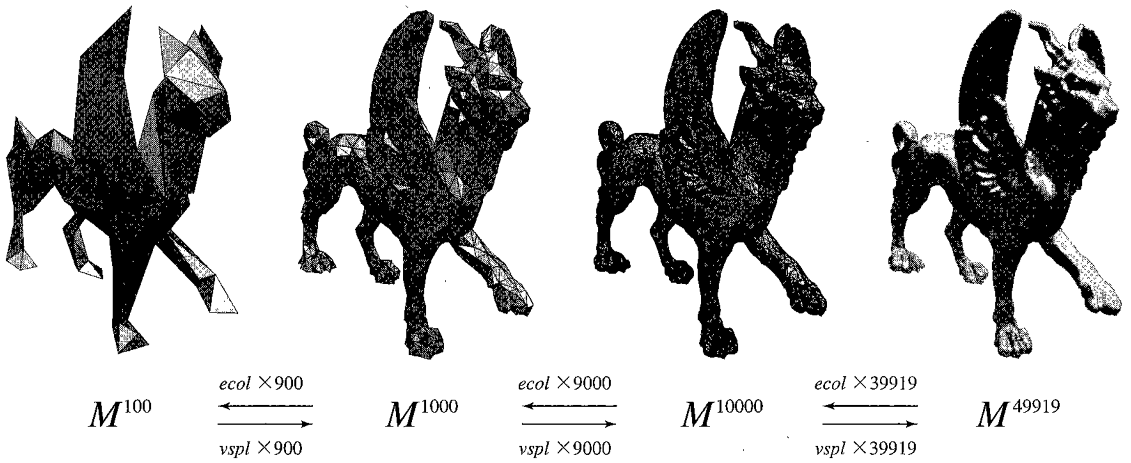

図-4　PM表現による多重解像度表現

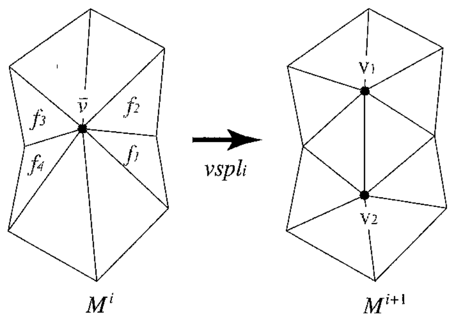

図-5　頂点分割に必要なデータ

■細分割接続性による多重解像度表現

もう1つのメッシュの多重解像度表現として代表的なものは，細分割接続性(subdivision connectivity)を用いた表現である．細分割接続性とは，メッシュを規則的に分割することにより得られる接続性のことである．中でも有名なのは4対1細分割と呼ばれる分割方法で，**図-7**のように，1つの三角形面のエッジに頂点を1つずつ増やし，この頂点を使って4つの小さい面に分割する．この分割方法では，解像度を詳細化レベル(LOD)により定義する．すなわち，ベースメッシュ(base mesh)と呼ばれる最も粗いメッシュをレベル0とすると，レベルが1つ上がるごとに面の数が4倍になるような詳細化メッシュが生成される．このとき，詳細化メッシュの形状を決定するための付加的な要素として，詳細化係数(detail cofficient)と呼ばれる係数が使われる．この係数は一般的にはベクトルとして与えられる．ただ，もとの頂点を保存する場合—これを補間細分割(interpolative subdivision)と呼ぶ—は，図の点線の矢印で示される係数は定義する必要がない．また前稿の細分割曲面は，この種類の表現の特別な場合として位置付けることができる．ただ細分割曲面では，詳細化係数は隣接頂点座標の線形和により自動的に計算できるので，わざわざそのためのデータを用意しなくてもすむ．

さて，この細分割接続性による多重解像度表現をより一般的な任意接続性のメッシュに適用するには，何らかの変換を行う必要がある．この変換操作は再メッシュ化)と呼ばれる．**図-6**に，任意接続性のメッシュから細分割接続性による多重解像度表現に変換した結果を示す．この表現は，代表的な再メッシュ化手法であるEckらの方法[1]を使って作成している．この方法では，まずもとのメッシュをなるべく少数の三角形面からなる領域に分割する．この三角形面領域から構成される疎のメッシュをレベル0のベースメッシュとし，レベルを1つずつ上げてゆくことで詳細化する．このとき，分割された三角形領域ごとにパラメータ化を行い，このパラメータを用いて，細分割後の頂点座標(詳細化係数)をもとのメッシュの面上に乗るように計算する．図では，詳細化レベルが0，2，4の各段階における結果を示している．

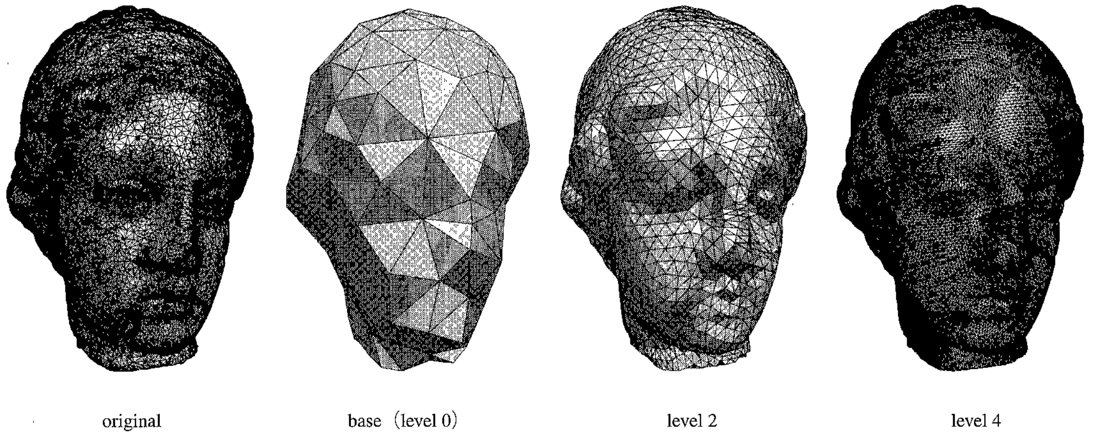

図-6　細分割接続性による多重解像度表現

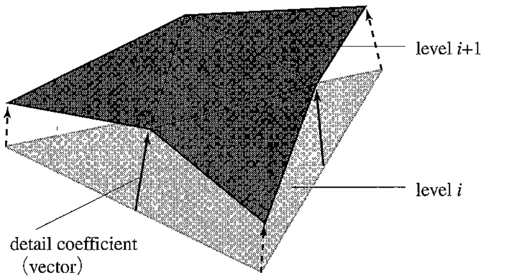

図-7　4：1細分割による細分割接続性と詳細化係数

	解　像　度	可　逆　性	データ量
PM表現	連続的	可　逆	大
細分割接続性	段階的	不可逆	小

表-1　PM表現と細分割接続性による表現との比較

■2つの表現の比較

上記の2つの表現の違いを比較すると(表-1参照のこと)，

- 解像度表現のタイプとしてみた場合，PM表現が頂点を1つずつ増減できる連続的な表現であるのに対し，細分割接続性による表現は，レベルを1つずつ上下させることにより詳細化／簡単化を行う段階的な表現である．
- 任意接続性を持つ一般的なメッシュに対して，PM表現はもとのメッシュを完全に復元できるのに対し，細分割接続性による表現は，一度再メッシュ化により近似変換する必要があるため，もとのメッシュには完全には戻らない（もちろん，最初からこの構造で記述されているものに関しては可逆性が成り立つが，そのようなメッシュはきわめて稀である)．
- PM表現はもとのメッシュの約2倍の頂点数および面数を必要とするので，その分必要なデータ量は増える．細分割接続性による表現では，メッシュの接続性が規則的であるため，明示的に記述する必要がないことから，おおよそ同じ面数のメッシュと比較すると，データ量はかなり小さくなる．

と，2つの表現にはそれぞれ一長一短を持つことが分かる．これまでの研究例でみてみると，PM表現は解像度の増減に関する自由度が高いことから，主に表示の効率化のために利用され，逆に，細分割接続性による表現は，その表現の簡便性から主にモデリング技術への利用例が多いようである．さらに，これらの表現技術がどのように活かされ，またネットワーク環境下で利用するにはさらにどのような技術が必要か，などの議論に関しては，後稿に譲りたい．

参考文献
1) Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W.: Multiresolution Analysis of Arbitrary Meshes, In Computer Graphics (Proc. SIGGRAPH 95), pp.173-182, ACMPress, New York (1995).
2) Floater, M. S.: Parameterization and Smooth Approximation of Surface Triangulations, Computer Aided Geometric Design, 14 (3), pp. 231-250 (1997).
3) Garland, M. and Heckbert, P. S.: Surface Simplification Using Quadric Error Metrics, In Computer Graphics (Proc. SIGGRAPH 97), pp.209-216, ACM Press, New York (1997).
4) Hoppe, H.: Progressive Meshes, In Computer Graphics (Proc. SIGGRAPH 96), pp.99-108, ACM Press, New York (1996).
5) Hoppe, H.: New Quadric Metric for Simplifying Meshes with Appearance Attributes, In Proc. IEEE Visualization '99, pp.59-66, ACM Press, New York (1999).
6) 鳥谷, 千代倉: 3次元CADの基礎と応用, 共立出版 (1991).

(平成12年9月1日受付)