1

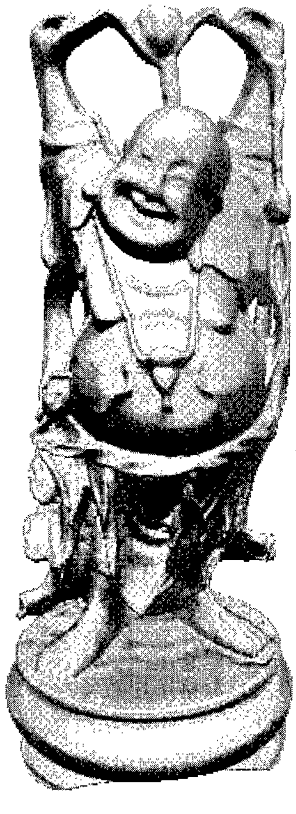
3次元メッシュモデルの生成と表現

鈴木 宏正　東京大学大学院工学系研究科精密機械工学専攻
suzuki@cim.pe.u-tokyo.ac.jp
http://www.cim.pe.u-tokyo.ac.jp/~suzuki/

■メッシュモデルの展開■

本特集では，コンピュータグラフィクス(CG)の世界で，通常ポリゴンと呼ばれるものに関して解説する．ポリゴンは，アニメーションやビデオゲームでは言うに及ばず，VR(Virtual Reality)，CAD(Computer Aided Design)など，3次元形状を表現するのに広く応用されており，インターネットの3次元コンテンツとしても中心的な存在となっている．

その意味では，特に新鮮味を感じないかもしれないので，まず，**図-1**を見ていただきたい．これは，CGで生成したHappy Buddhaという像である．スタンフォード大学のLevoy教授を中心としたグループのプロジェクトの成果である．このプロジェクトでは，3次元物体をレーザースキャナという装置で測定し，その測定データから精細な多面体データを作っている[5]．最近，ミケランジェロの彫刻の測定を行っており，ミケランジェロの鑿(のみ)の跡まで忠実に再現する見事な多面体モデルを生成している[3]，☆1.



メッシュデータはStanford大学Levoy教授の研究室から入手

図-1　測定データによる彫刻のモデル

多面体というと，算数でおなじみの直方体や正20面体などのような形を連想される人も多いと思うが，図-1の多面体は，面の数が100万個もある．**図-2**は部分を拡大したもので，微小な三角形の面がびっしりと並んでいるのが分かるだろう．このような多面体をメッシュという．特にすべての面が三角形のものを三角形メッシュという．

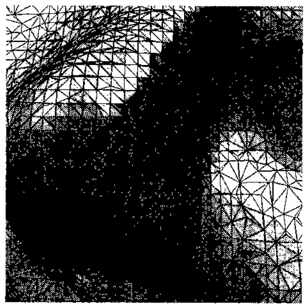


図-2　メッシュの拡大図

測定技術の発展や，後述する細分割曲面と呼ばれる技術などによって，非常に詳細なメッシュが生成されるようになってきており，それを扱うための新しいモデリング技術に関する研究がCGの分野で活発に行われている．特に，インターネット上での3次元メディアとしても重要な位置を占めており，そのための圧縮や透かしなどに関する研究開発も活発である．この特集では，本稿も含めて3つの記事によって，その概要を紹介

☆1 http://graphics.stanford.edu/projects/mich/

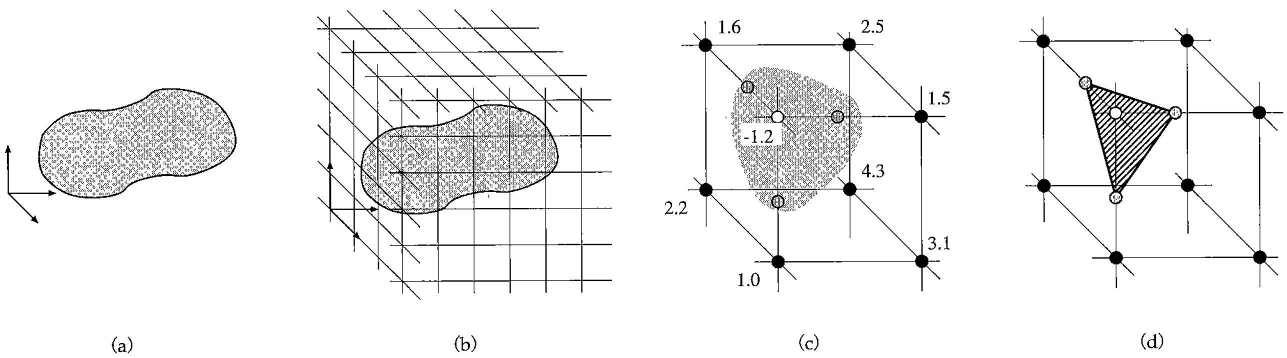


図-3　マーチングキューブ法

する．本稿では，メッシュモデルの中でも特に応用範囲の広い，すべての面が三角形の三角形メッシュを中心にして，その表現や処理の基礎となる技術的な事項と，それらに関連した代表的な話題について紹介する．個別のより詳しい内容は，他の2稿に譲る．

■メッシュの生成（マーチングキューブ法）■

このようなメッシュはどのようにして測定データから生成されるのだろうか．そこでは，マーチングキューブ法というアルゴリズムが大活躍している[1), 4), ☆2]．これを説明するために，図-3 (a)のような物体を考えよう．この物体の表面を表す三角形メッシュを生成することを考える．そのために，まず物体の置かれた3次元空間に図-3 (b)のような格子を置く．そして各格子点において，その点から物体までの距離を計算し，格子点の値とする．このとき，物体の外部の点では距離に-1を掛けて負の値にしておく（符号付距離）．実際には物体の形は未知なので，このような値は，上述のレーザースキャナによるデータから生成する．X線CTやMRIのような断層撮影装置によって得られる3次元画像あるいはボリュームデータからも生成することもできる．

そして，この格子データから，距離値が0になるような曲面を作ることができれば，その曲面は物体の形を表すことになるだろう．これを等値面探索という．そのために，図-3 (c)のように格子の中の8個の格子点からなる立方体を考える．また，各格子点の値が図のような値になっているとする．これは，白丸の点だけが物体の外側にある状態である．空間で値が連続的に変化することを仮定すれば，0の曲面（ゼロ等値面）は，正の値を持つ格子点と負の値を持つ格子点の間を通過する．そして，その位置はそれらの値を内分することによって計算することができる．そのように考えると，ゼロ等値面は図-3 (c)のように立方体の辺と交差しているはずであり，これらの交点を結んで図-3 (d)のような三角形を作ると，この三角形はゼロ等値面の一部を近似しているものと考えられる．このようにして，8個の格子点の値を使って，立方体の中に三角形を生成できる．これを格子内のすべての立方体について行うと物体の表面を表す三角形メッシュを容易に生成できる．これをマーチングキューブ法という．Levoy教授らのプロジェクトでは，この格子間隔を非常に小さくとることによって精細なモデルを生成している．

☆2 これ以外にも測定データからメッシュを生成する方法はある．

■メッシュの表現（STRIP表現）■

三角形メッシュを表現するには，最も簡単には，俗に“トライアングルスープ”と呼ばれる方法を使って，各三角形の3頂点の座標値を並べればよい．より一般的には，頂点リスト方式と呼ばれる多面体を表現するデータ構造を用い，面，辺，頂点間の接続関係を含めて表現する．一方，以下で示すように，面数の大きいメッシュを扱うための，より効率的な表現としてSTRIPがある．

図-4 (a)のような三角形メッシュを画面に表示することを考えよう．そのためには，図-4 (b)のようにメッシュの三角形を1つずつ表示してゆけばよい．3次元CGで広く使われているOpenGLなどのAPIでは，アプリケーションプログラムから頂点の座標がグラフィクスパイプラインへと転送される．この図からも分かるように，三角形ごとに3頂点を与えると同じ頂点を何度も転送することになり処理効率を低下させることになる．

そこで，転送される頂点数を減らす方法が考慮されている．その代表的なものが，STRIPである．これはこの図のように三角形が帯状に並んでいる場合に適用で

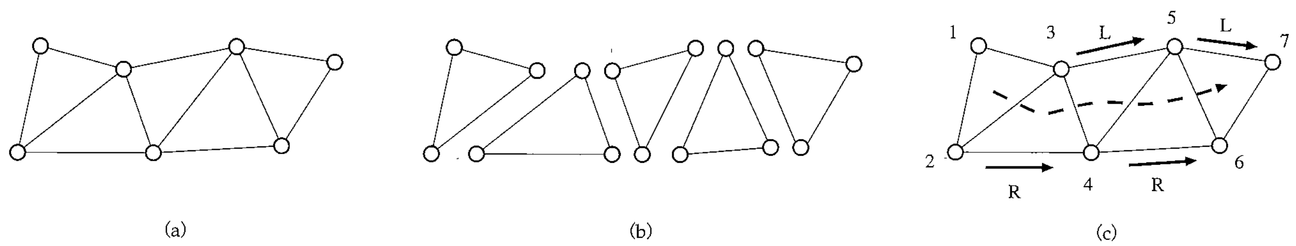


図-4　STRIPデータ構造

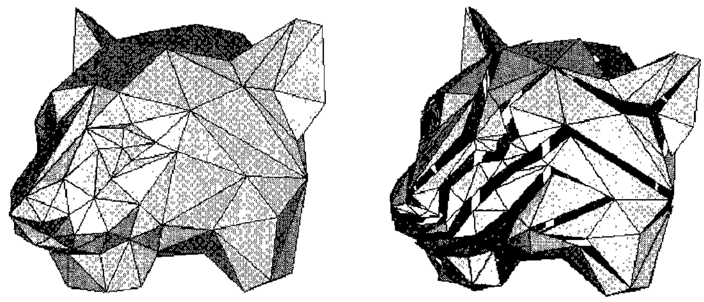


図-5　STRIP化

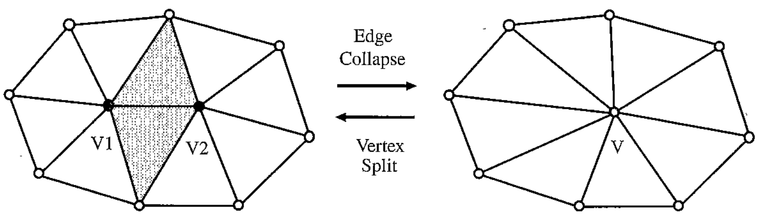


図-6　Edge CollapseとVertex Split操作

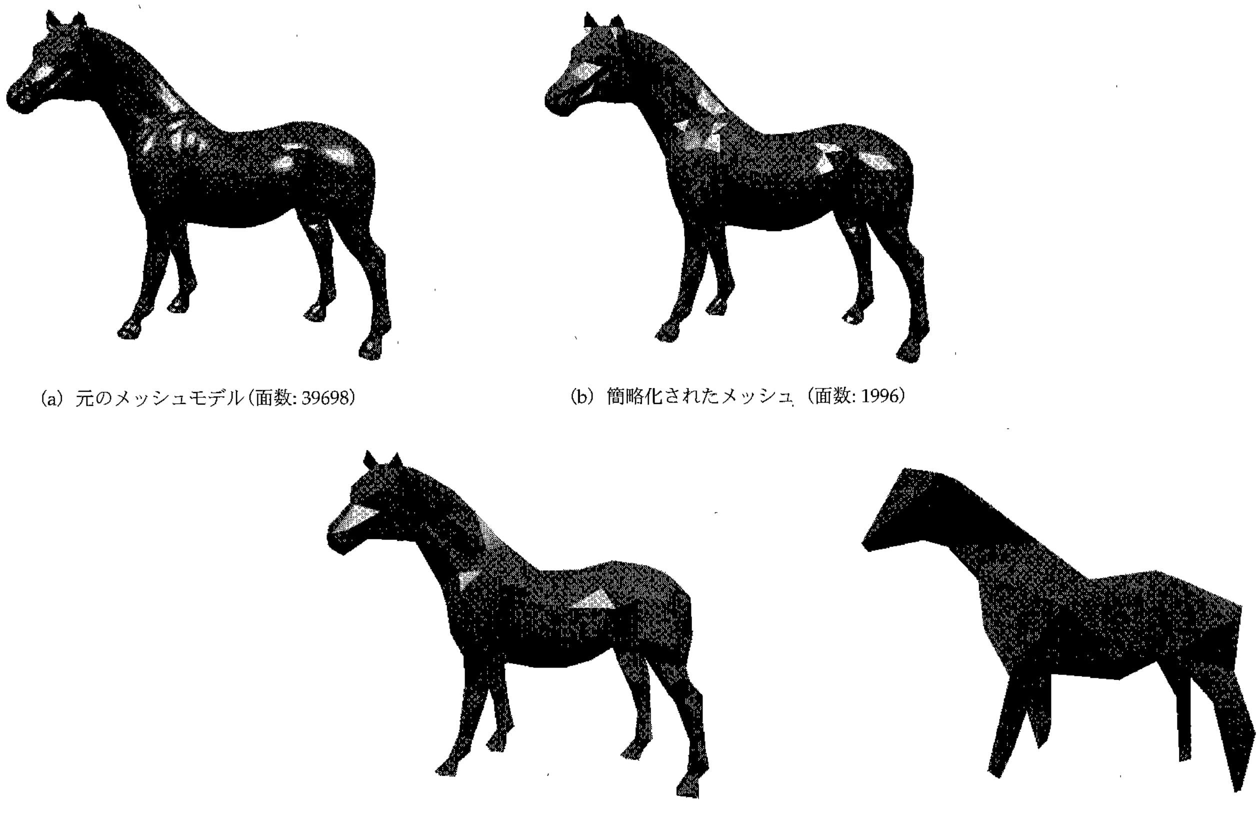
きるもので，図-4(c)に示すように，一番端の三角形の頂点を1，2，3とすると，2番目の三角形は，三角形1，2，3のうちの後ろから2つの頂点2，3を用い，さらにそのうちの右側の頂点から辺を延ばして新しい頂点4とすることで定義される．さらに次は左側から辺を延ばし，頂点5とする．したがって1，2，3，4，5，6，7という頂点番号の列から，三角形の並びを生成することができる．これは右，左，右，左と規則正しく進む場合であるが，そうでない場合にも対応できるように拡張することができる．また，一般のメッシュも，図-5に示すように“切れ目”を入れて，STRIPに変換することができる．

三角形を1つずつ表示する場合は，三角形1つにつき，3個の頂点を転送する必要があったが，STRIPを使うと，最初の三角形以外は，三角形1つにつき1個の頂点で定義されることになる．この性質を活かしてメッシュデータの圧縮を行うことができる．圧縮は，大規模なメッシュデータをインターネットによって転送したり，アーカイブ化するのに重要な技術であり，さまざまな高度な方法が開発されている[6]．これについては，大渕氏の稿に譲る．

■多重解像度表現■

図-1に示すような大規模なメッシュを表示することを考えよう．ズームアップすると図-2のように，各三角形は識別できるが，遠い視点からメッシュを表示すると，1つの三角形が1ピクセル以下になってしまう．その場合，詳細なモデルを持つ必要はなく，もっと粗い(面数の少ない)モデルでも十分となる．このように，異なる詳細度でモデルを表現し，必要に応じて面数を変えられるメッシュの表現方法を多重解像度表現(multi-resolution representation)という．

このような多重解像度表現を構成するには，大きく分けて2つの方法がある．それは，だんだん粗くする方



(a) 元のメッシュモデル(面数: 39698)　(b) 簡略化されたメッシュ (面数: 1996)

(c) 簡略化されたメッシュ (面数: 596)　(d) 簡略化されたメッシュ (面数: 96)

メッシュデータはStanford大学Levoy教授の研究室から入手

図-7　メッシュの簡略化

法と，だんだん細かくする方法である．まず，前者は，詳細なモデルから始めて，その面の数を少しずつ減らしていくもので，モデルが段階的に簡単になるので，簡略化(simplification)と呼ばれる処理である．面数を減らす1つの方法は，図-6に示すように，中央の辺を潰して，左から右へとメッシュを修正するものである．そうすると，三角形の数が2つ減る．この操作をEdge Collapse操作と呼ぶ．これを繰り返していくと，面の数はどんどん減っていく．図-7にその例を示す．簡略化では，このような膨大なメッシュモデルに対して，いかにしてメッシュの形を保存したまま面の数を減らすか，ということがポイントになる．これについては金井氏の稿に譲る．

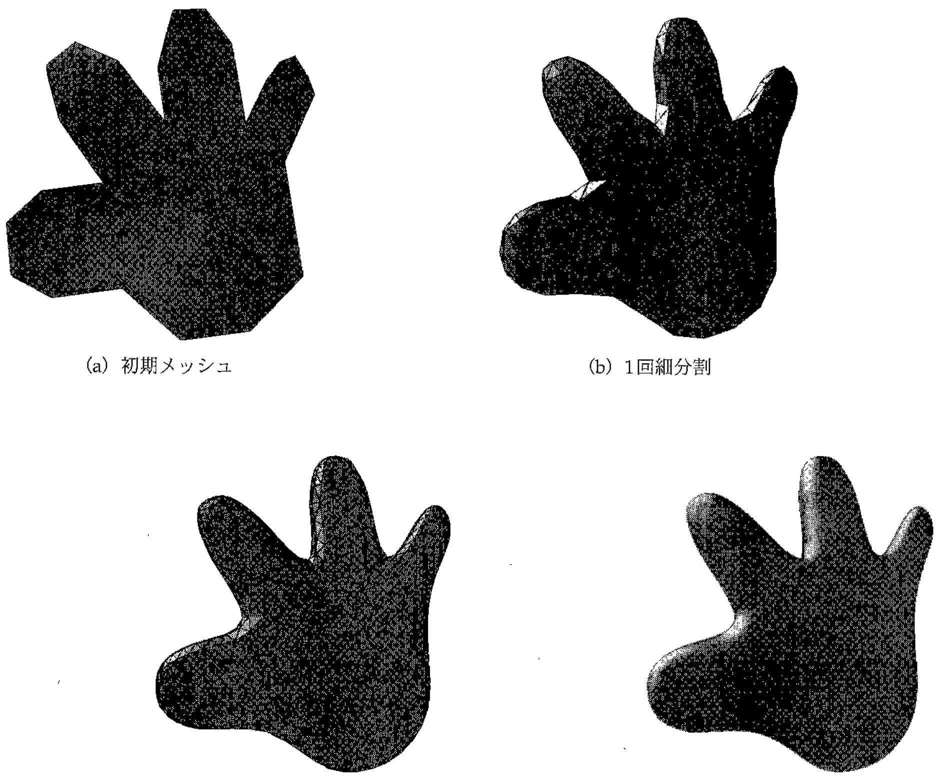
一方，粗いメッシュから細かいメッシュにしていく方法に，細分割曲面(subdivision surface)がある[7]．細分割曲面は，多面体をどんどん分割してゆくことによって形を滑らかにしてゆく方法である．図-8はその一例で，Loop細分割曲面といわれるものである．多面体を分割し，分割後に頂点の位置を形状が滑らかになるように調整する(スムージング)ということを繰り返す．

細分割の計算は非常に単純で，必要な精度に応じて「いつでも好きなだけ」分割することができる．何回分割しても，多面体であることには変わりがないのであるが，数学的には無限回の細分割を行うと滑らかな曲面に収束する．

細分割曲面は，現在非常に活発な研究分野となっている．実用面でも，1997年に米国のPixarという会社が細分割曲面技術を用いてGeri's GameというCGアニメーションを作成しアカデミー賞ショートフィルム部門賞を受賞した．これが火付け役となった[2]．その後，A Bug's LifeやToy Story 2☆3などでも利用されており，CGの分野では曲面モデルの主役になろうとしている．

そのように注目されるのには，多重解像度表現に加えて，もう1つの大きな理由がある．それは，図-8(d)のような曲面を表そうとすると，従来のパラメトリック曲面では，全体を1つの曲面で表すことができず，いくつものパッチと呼ばれる矩形の曲面を貼り合わせて表現する必要があった．そのため，パッチとパッチの境界で滑らかに接続せず，折れてしまったり，またせっかく滑らかに接続するように曲面を生成しても，アニメーションのために一部の曲面を変形すると，その接続性が壊れてしまい，再度調整する必要があった．それに対して，細分割曲面は全体を1つの曲面で覆うこ

☆3 http://www.pixar.com/shorts/shorts.html



(a) 初期メッシュ　(b) 1回細分割　(c) 2回細分割　(d) 極限曲面

図-8 Loop細分割曲面

とができ，(いくつかの点を除いて)いたるところで滑らかな曲面を定義できるので，そのような心配がまったくなく，連続性を保ったまま自由自在に変形を行うことができるのである．

■メッシュモデリングについて■ ■ もっと知りたい人へ ■

本稿では，メッシュモデリングをベースとした，いくつかの話題について，その背景とともに簡単な紹介を行った．これ以外にも，メッシュデータの知的所有権のための透かし技術や，強調やモーフィングといった編集技術などさまざまなテーマについて，非常にアクティブに研究が行われている．

また，これまで3次元のモデリングに関しては，主として，アニメーションのキャラクターデザインを対象としたCGの分野と，機械設計を対象としたCADの分野，そして，画像からの3次元情報の復元を対象としたコンピュータビジョンの3分野で，研究が行われてきた．近年急速に，この3分野が融合しており，メッシュに限らず，ユニークな技術が展開しており，見逃せない領域となっている．

本分野の研究に関する文献を見るには，毎年夏に行われるACM(米国計算機学会)のSIGGRAPHという学会のチュートリアルの資料(Course Note)が手っ取り早い．それらを参考文献に示した．なお，しばらく前からCD-ROMでの配布も行われている．

謝辞　慶應義塾大学金井崇先生，筆者の研究室の院生，三谷純君，竹内真悟君には，本文の図の作成をお願いした．ここに感謝します．

参考文献

1) Bloomenthal, J. eds.: Introduction to Implicit Surfaces, Chap. 4, Surface Tiling, Morgan Kaufmann Pub (1997).
2) DeRose, T., Kass, M. and Truong, T.: Subdivision Surfaces in Character Animation, In SIGGRAPH 1998 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, pp.85-94 (1998).
3) Levoy, M. et al.: The Digital Michelangelo Project: 3D Scanning of Large Statues Conference Proceedings, Annual Conference Series, ACM SIGGRAPH 2000, pp.131-144 (2000).
4) Lorensen, W. E. and Cline, H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, In Computer Graphics (SIGGRAPH 1987 Proceedings), Vol.21, pp.163-169 (1987).
5) 3D Photography, ACM SIGGRAPH 2000, Course Note, No.19 (2000).
6) 3D Geometry Compression, ACM SIGGRAPH 2000, Course Note No.38 (2000).
7) Subdivision for Modeling and Animation, ACM SIGGRAPH 2000, Course Note No.23 (2000).

(平成12年9月7日受付)