

# 3

## インターネット流通を目指した 圧縮と透かしの技術

大淵 竜太郎 山梨大学工学部コンピュータメディア工学科  
ohbuchi@acm.org  
<http://www.kki.yamanashi.ac.jp/~ohbuchi/>

### ■インターネットと3次元モデル■

鈴木氏の稿に登場したHappy Buddhaのメッシュモデルは約100万個の面、ほぼその半数の頂点からなる。このような精緻なモデルをインターネット経由で配布したり販売したりするといった何が起こるだろう。

まず気づくのはそのデータ量である。座標のx, y, zそれぞれを32bit浮動小数点数で持つと、頂点座標値だけで12MByteである。このメッシュを頂点リスト表現で表すと頂点接続性をインデックスで持つことになるが、このインデックスのデータ量は頂点座標値のデータ量と同等以上なのが普通である。無圧縮だと、頂点の座標と接続性をあわせて25MBを超える。色やテクスチャ座標を付加するとデータ量はさらに増える。次に思いつくのは、販売したデータのセキュリティに関する事項である。「データを暗号化して配布し、代金を払った相手のみに鍵を渡す」、のでは盗用防止には不十分である。渡された鍵を使って平文化し元のモデルになった後は自由に横流しができるからである。また、そのデータが何かの設計図だとすると、悪意を持った微小な改変がその設計プロジェクトを破綻させるかもしれない。何らかの仕組みで、盗用防止、改ざん検出、秘密保持その他の操作を実現する必要がある。

### ■メディアデータ型と操作■

少し前までは、いわゆるマルチメディアデータのデータ型といえば、テキスト、音、静止画像、そして動画像であると考えられていた。しかし、インターネットの発展、特にWeb3Dの普及などをきっかけに、3次

元モデルも立派なマルチメディアデータ型の1つとして認められつつある。

あるマルチメディアデータ型を抽象データ型として見たとき、それぞれに対応する操作の集合が用意されていることに気づく(図-1)。データ型として最低限必要なのは入力、出力、編集である。これに加え、特にインターネットの普及に伴い、圧縮、索引付加、ハイパーリンク付加、内容による検索、さらには各種のセキュリティ関連(たとえば、暗号化、著作権の主張や改ざんの検出)の操作の重要性が増してきた。これらの操作はテキストに対してはかなりよく理解されており、さらに法的・社会的インフラストラクチャ(著作権法など)も存在して実際に運用されている。しかし、データ型が音や静止画像、動画像になると理解度が下がってゆく。たとえば、テキストのハイパーリンクや全文検索は当たり前だ。しかし、「この動画のいつの何(領域や物体)からいつの何」へのリンク、「あんな形がこう動くシーン」の検索などの操作の実現は格段に困難である。さらに、本稿の対象である3次元メッシュモデルでは、研究の歴史が浅いこともあって動画像以上に分からないことが多い。

これまで主に研究されてきた3次元モデルデータに対する操作は、最低限必要な3つの基本操作、入力、出力、編集である。たとえばいわゆる形状モデリングは入力・編集に、仮想現実表示は出力に当たる。しかし、最近、圧縮、索引付け、内容による検索、そしてセキュリティ関連の操作に対する研究が始まりつつある。以下、本稿では、ポリゴンメッシュデータ型を対象とし、インターネット時代になって初めてその重要性が増した操作の中から、圧縮および電子透かしの技術について紹介する。

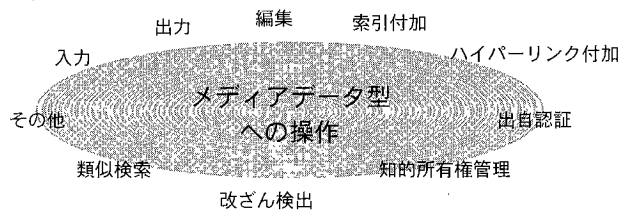


図-1 各メディアデータ型には、それに対応する操作の集合が用意されている

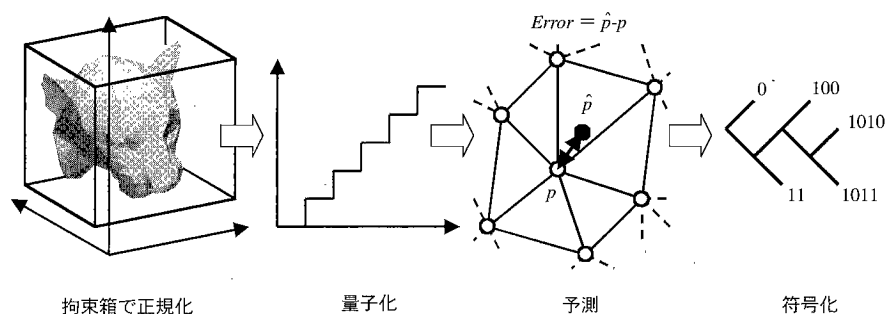


図-2 頂点座標値を圧縮する一般の手順

### ■ 3次元モデルの圧縮 ■

3次元メッシュを圧縮・伸長する場合、元のデータがそのまま欲しい場合と、使用上問題がなければ(たとえば、見かけが同じならば)多少近似されていてもよい場合がある。前者を無損失圧縮、後者を損失圧縮と呼ぶ。メッシュを圧縮し転送する場合、まとめて送る一括転送と、まず概形を送り後から徐々に詳細を追加してゆく漸近転送がある。ほとんどの方がJPEG圧縮された静止画像の漸近転送を見たことがあると思う。さらに、圧縮したいデータが時間変化する場合、これを最小の時間遅れで逐次に圧縮・転送したい場合があるが、これを本稿では逐次圧縮・転送と呼ぶ。逐次圧縮・転送は、たとえば携帯電話の音声伝送では当たり前のように使われている。

3次元メッシュモデルは頂点座標値、頂点接続性、および頂点や面などの属性値により定義される。頂点座標と頂点接続性についてその圧縮技術をみてみよう。属性は頂点に付属することが多く、隣接する頂点やテクスチャ座標が近いなど、頂点座標と類似のコヒーレンスを持つことが多い。したがって本稿では頂点座標と一緒に扱うこととする。

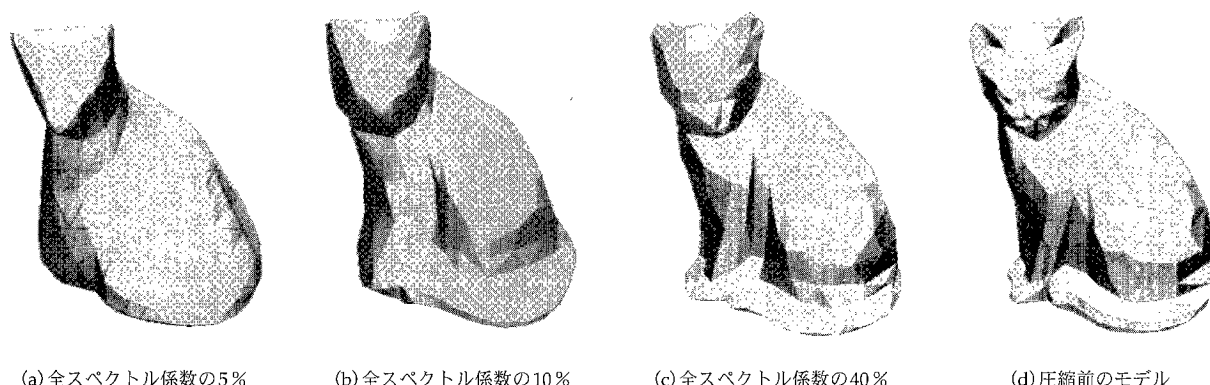
#### ■ 頂点座標や属性値など数値列の圧縮

頂点座標の圧縮は、一般に、図-2に示すような手順で

行われる。まずモデルの座標値を拘束箱で正規化した後、整数値に量子化する。すると座標軸ごとに16bit以下、ほとんどの場合は12bit程度で十分である(レーザーレンジスキャナなどで取り込まれたモデルの座標の精度はよくて12bitである)。量子化された座標値は、次に何らかの手法で予測を行う。予測値と実際の値との誤差は元の座標値そのものより小さい整数で表現できるので、効率よく符号化できるためである。符号化そのものはHuffman符号や算術符号などのエントロピー符号化技術を用いる。

座標値圧縮の鍵となるのは予測である。Taubinらที่ใช้た手法では頂点のspanning tree(全域木)を深さ優先探索し、頂点座標を1次元の数値列とする。これに対し音声圧縮で用いられるLinear Predictive Coding(線形予測符号化)などを適用する。プログレッシブメッシュを拡張したアルゴリズムを用いる場合には、頂点分割操作で作られる頂点の座標を予測するのに、その頂点に隣接する複数の頂点座標の平均(など)を予測値とする(図-2の例)。もちろん、尖点や稜線の付近では頂点座標予測がうまく働かない場合もあり、その部分の圧縮率は下がる。

座標値の予測は不規則サンプルの集合からの近似補間問題ととらえることもできる。たとえばKarniとGotsmanは、メッシュの接続性から求まるLaplacian行列を固有値分解し、得られた固有ベクトルに頂点座標



(a) 全スペクトル係数の5% (b) 全スペクトル係数の10% (c) 全スペクトル係数の40% (d) 圧縮前のモデル

図-3 メッシュスペクトルによる損失圧縮の例。スペクトル係数の (a) 5%, (b) 10%, (c) 40% を使って座標値を復元

を射影してメッシュスペクトル係数を求めた。メッシュスペクトル係数は規則サンプルの場合のフーリエ係数に相当し、メッシュスペクトル係数と基底である固有ベクトルからメッシュ形状を近似できる。この手法は頂点座標を損失圧縮する。頂点接続性は何らかの無損失圧縮手法を使用する。図-3はKarniらの手法で圧縮を行った例である。スペクトル係数を固有値の小さい方から全体の5%、10%、および40%使って復元したのが(a)、(b)、(c)で、これらが、元のモデル(d)を近似していることが分かる。

色やテクスチャ座標、法線ベクトルなどの属性値も、隣接する頂点の間で近い値を持つなど、頂点座標と同様なコヒーレンスを持つ場合が多い。したがってこれらの数値の圧縮にも頂点座標値と同様な手法が使えることが多い。ただ、接続性の圧縮を行う場合、変更された接続性と属性との間で矛盾が生じないようにする必要があり、これはメッシュの簡単化の場合も同様である。

### ■頂点接続性の圧縮

頂点接続性の表現に頂点リスト方式を使う場合、すべての頂点に番号を振り、その3つ組で1つの三角形を表現する。これには、メッシュ全体の頂点個数がVなら三角形1つ当たりの位相の表現に $3\log_2 V$ bitを要する。たとえば、あるメッシュがV=4,096個の頂点からなるとすると、接続性の表現に三角形当たり36bitを要する。圧縮前の座標値や属性に匹敵するデータ量である。

ほとんどのポリゴンメッシュは、局所的には平面に埋め込みのできるグラフと考えることができる。このような平面グラフ圧縮のブレークスルーは、Taubinらが1996年ころ発表(論文は1998年)したメッシュ接続性の無損失圧縮手法であった。この手法は、まず三角メ

ッシュの頂点を全域木で覆い(図-4(a)) Vertex Tree(頂点木)を作る(図-4(b))。次に、この頂点全域木に含まれる辺で鉄を入れて元のメッシュを切り開くと、今度は三角形の木ができる(図-4(c))。頂点木も三角形木も効率よい符号化が可能である。三角形木の場合、根となる三角形とその初期辺の向きを決めて深さ優先探索をすると、三角形の接続関係は(1)L(次の三角形が左辺に隣接)、(2)R(次の三角形が右辺に隣接)、(2)B(右辺と左辺の両方に隣接;分岐)、(4)T(葉)、の4種類のシンボルの列によって記述できる(左右を決めるには面の裏表が決まる必要があることに注意)。こうしてシンボル列化された頂点木と三角形木をそれぞれ再生し、それらを組み合わせると元のメッシュの接続性が再現できる。ここまででも大変コンパクトな表現であるが、頂点木および三角形木を記述したシンボル列中の“LRLRLR”のような繰り返しをランレングズ符号化し、これを算術符号などでエントロピー圧縮すると、全体として非常に高い圧縮率で接続性を圧縮できる。実際には木が複数になるなどいくつかの例外があり、アルゴリズムは上記より多少複雑になる。

圧縮しようとするモデルによって異なるが、Taubinらの手法によると三角メッシュの接続関係を三角形当たり平均2.2bitで表現できた。この圧縮手法は後に漸近圧縮へ拡張され(Progressive Forest Split(PFS)圧縮方式)、さらに非多様体の扱いなどを加えた後、MPEG-4 SNHC標準(<http://www.cselt.it/mpeg>)に採用された。

その後、Taubinらの手法に刺激され、よりすぐれた圧縮率を持つ無損失の頂点接続性圧縮アルゴリズムが多数登場しているが、その詳細は文献7)に譲る。

Taubinらのアルゴリズムに限らず、ほとんどの接続性圧縮アルゴリズムは、頂点接続性のグラフが平面に埋め込めるとして、そのようなグラフの特徴を利用し

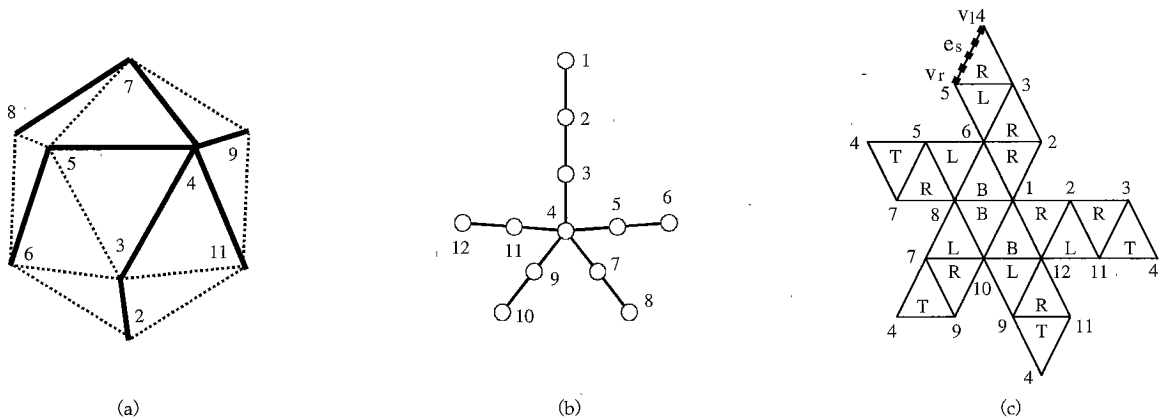


図-4 (a) 正20面体を (b) 全域木 (実線) で覆い (数字は頂点番号), 頂点木に含まれる辺で20面体を切り開くと (c) の三角形木が得られる. (c) の三角形木を開始辺  $e_s$  から始めて深さ優先探索すると, 木を符号化した符号列が得られる. 頂点木および三角形木両方の符号列から元の20面体が復元できる

ている. しかしこの仮定が成り立たない場合もある. たとえば, 互いに連結していない要素が複数ある (非連結), ドーナツのように穴が開いている (球と同相でない), クラインの壺のように面の裏表の区別がない (向き付け不可能), 連結要素が点1つや辺1つでつながっている (非多様体), などの場合である. 実は, 巷の Web3D (VRML) モデルではこれが普通である. これら「性質の悪い」モデルを圧縮する場合, 何らかの操作を用いてこれらのメッシュを平面に埋め込めるグラフに (可逆) 変換した後に圧縮すればよい. 変換操作を圧縮データの一部に記録しておけば, 元のメッシュが再現できるはずである. こうした変換は接続性圧縮の前処理かもしれないし, 圧縮処理アルゴリズムに組み込まれている場合もある.

#### ■漸近圧縮・転送

圧縮で忘れてならないのは漸近転送である. すぐに想像できるように多重解像度表現, LODや簡単化と関係があり, “究極のLOD” と呼ばれるプログレッシブメッシュなどは漸近転送に拡張できる. 漸近転送アルゴリズムの研究は最近特に盛んで, Hoppeのプログレッシブメッシュを改良したPajarolaらの手法, あるいは頂点除去によるLiとKuoらの手法などがある. 詳しくは文献7)に含まれる参考文献を参照してほしい.

#### ■実用化へ

MPEG-4標準にはメッシュの圧縮および顔などのアニメーションデータの圧縮が含まれる. またJava言語の3次元APIであるJava3Dにはメッシュの圧縮が含まれる (<http://java.sun.com/products/javamedia/3D>). その他, Virtue社 (<http://www.virtue.com>) や MetaCre-

ations社 (<http://www.metastream.com>) はそれぞれ圧縮データフォーマットやアーカイブソフトウェア, プラグインビューアなどを製品としている.

#### ■電子透かし■

データ埋め込み, または (電子) 透かしと呼ばれる技術は, watermark (透かし) と呼ばれる何らかの構造体をコンテンツに付加する (図-5参照). このとき, 透かしが埋め込み対象となるコンテンツ本来の目的 (たとえば人による表示と鑑賞) を阻害しないように, かつ, コンテンツから透かしを除去できないようにする. 透かしは, 説明の付加, 改ざんの検出, 無許可複製データの再生禁止, あるいは正規の購入者の認証など, 何らかの形でコンテンツを管理するために用いることができると期待されている.

電子透かしにはいくつかの分類法がある. まず透かしはその可知性により分類できる. 透かしの対象となるデータの使用目的に影響を与えない透かしが不可知な透かしである. これまでの透かしはほとんどが不可知な透かしであるが, コンテンツの無許可の再配布を牽制する目的で, 可知な透かしを積極的に使った例もある. 注意したいのは, 3次元モデルの可知性の定義はそれ以外のメディアデータ型の場合より複雑な点である. たとえば静止画の電子透かしの場合, 画面で表示したものを人が見て透かしの存在に気づかなければよい. 3次元モデルでは, 観測者が人であっても, 照明やカメラのパラメタ, レンダリング手法などにより観測法が変わる. さらに, その3次元モデルを機械部品の製造に用いる場合など, 透かしによる微小な座標値の変

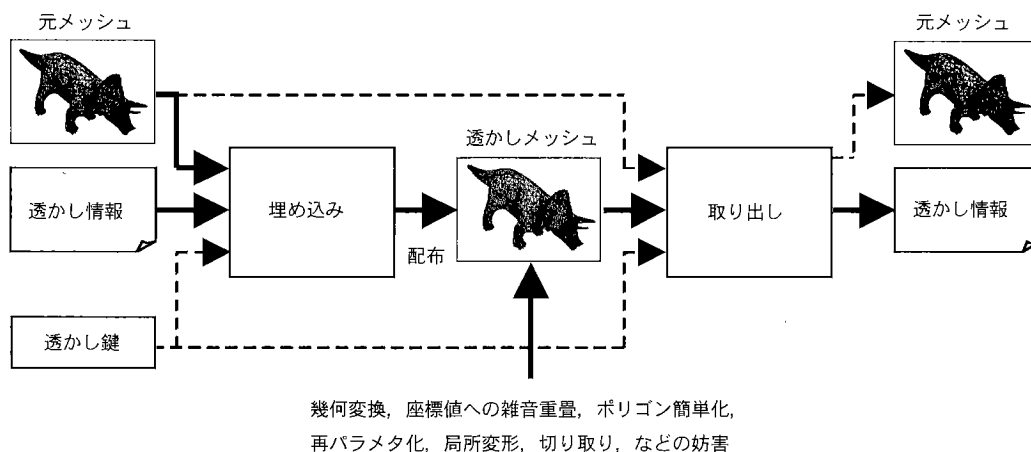


図-5 電子透かしの基本モデル

化がその機械部品の働きを失わせる(エンジンがかからないなど)こともあり得る。

電子透かしは頑強性により頑強な透かしと脆弱な透かしに分類できる。前者は被覆データに加えられる意図的・非意図的な変更にも耐えねばならない。脆弱な透かしは、意図的な変更によって透かしが改変または破壊されることを積極的に利用し、改ざんの検出を行う。頑強な透かしは妨害に耐えることを要求され、著作権の主張などに使用する。ここで非意図的な変更とは、コンテンツの通常の使用において予期されるもの、3次元モデルならばたとえばアフィン変換などを指す。意図的な変更とは、透かしを変更あるいは破壊するなどの目的で意図的に加えられるものを指す。別な分類として、取り出しに元の(透かし埋め込み前の)データが必要な透かしを秘密透かしと呼び、そうでないものを公開透かしと呼ぶ。後者の方が利用しやすいが、技術的にはより困難である。

ここで記憶にとどめて欲しいのは、その他の各種セキュリティ技術と同様に、電子透かし単体では著作権保護などの現実の応用で効果を上げることは難しい点である。実際に何が 필요한かは応用ごとに異なるが、透かしと組み合わせて使う電子署名や透かしを入れたデータを預託するための信用できる預託機関などのインフラストラクチャ、そしてこれらインフラストラクチャの社会的・法的な認知など、満たされねばならない要件は数多い。

### ■ポリゴンメッシュを対象とする電子透かし

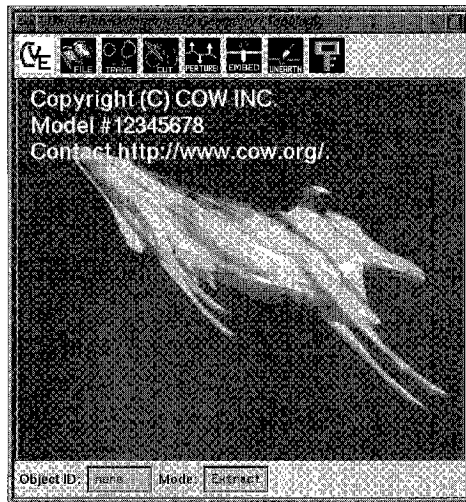
圧縮の章でみたように、メッシュには頂点座標、頂点接続性、頂点属性、の3要素があるが、これらすべてが電子透かしの埋め込み対象となる。さらに、これら

の3要素が時間変化する動的なメッシュも透かしの対象となる。

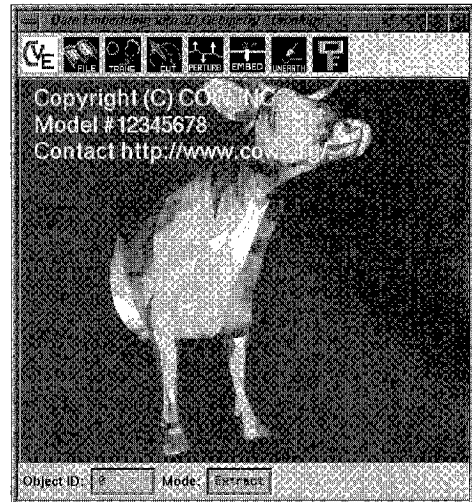
頂点座標値の変更による埋め込み：Ohbuchiら<sup>5)</sup>のアルゴリズムはポリゴンメッシュによって定義された形状の座標値を埋め込みの対象とする。座標値そのままでは、たとえば平行移動や回転のような日常の幾何変換をメッシュに加えると透かしが消えてしまう。そこでよく用いられる特定のクラスの幾何変換、たとえば相似変換やアフィン変換に対する不変量を利用して透かしの埋め込んだ(図-6参照)。これらの不変量は頂点座標から誘導されたもので、アフィン変換不変量の一例としては四面体の体積の比がある。部分的な切り取りに対し耐性はモデルの各所に繰り返して透かしの埋め込むことで実現した。この手法により埋め込んだ透かしは、頂点座標への乱数値重畳およびメッシュ全体に渡る頂点接続性の変化(ポリゴン簡単化あるいはedge flippingなど)により破壊された。

この他、Benedensらは面法線ベクタの集合という大局的特徴を変更して透かしの埋め込んだ<sup>1)</sup>。また、2次元静止画像において脆い透かしの提案したYeoとYeungらは3次元メッシュモデルでも最初に脆い透かしの試みている<sup>8)</sup>。

頂点座標値への変換領域での埋め込み：金井らは三角メッシュをウェーブレット変換し、その変換領域の係数を操作して情報を埋め込む手法を提案した<sup>3)</sup>。変換領域で操作した係数を逆変換すると透かしの埋め込んだ形状が得られる。彼らの透かしはアフィン変換に耐え、かつ頂点座標に重畳されたランダムノイズに対してもある程度の頑強性を有した。必ずしも変換領域の手法とはいえないが、Praunらの手法<sup>6)</sup>の手法では、与えられたメッシュの形を、透かし情報で重み付けした空



(a)アフィン変換後



(b)切り取り後

図-6 Ohbuchiらの透かしは (a) アフィン変換に耐え、(b) 切り取りにもある程度耐えた

間領域の基底関数(正規性や直交性は持たない)で変形する。この手法は相似変換, 切り取り, ランダムノイズ重畳などの妨害に対しある程度の頑強性を有したが, 複雑なメッシュ(頂点数の多いメッシュ)に対してのみ適用可能という欠点がある。

頂点接続性の変更による埋め込み: 3Dモデルのトポロジ, つまり頂点の結合関係を変更して埋め込む透かしもある。単純には, 文字パターンの形にメッシュの大きさを変えたり, メッシュを切り絵のように文字などのパターンに切り抜くこともできる。また, 三角ストリップを切り, その三角形の隣接関係にビット列を埋め込むこともできる<sup>5)</sup>。これらの手法はすべての座標変換に耐えるが, 接続性を変更する操作により壊れる場合もある。

その他の手法: これらの他, メッシュに付随する属性, たとえばテクスチャ座標などを変更してデータを埋め込む手法も報告されている。また, MPEG-4の facial animation parameters(顔アニメーションパラメタ群; FAPs)ストリームの位相成分を変調することで動きに透かしを埋め込んだ例もある<sup>2)</sup>。大変興味深いのは, 透かしの入った顔アニメーションをレンダリングして得られた2次元動画(MPEG-2圧縮)をFAPs抽出用の画像処理プログラムに入力しFAPsを抽出することで, レンダリング後の2次元動画像から透かしの情報を取り出すことに成功した点である。

## ■もっと詳しく知るには■

スペースの都合上, メッシュ圧縮に関する論文は参考文献リストには列挙しない。しかし, 本稿で参照したメッシュ圧縮関連の文献はすべて, 著者名や発表年よりTaubinらの資料<sup>7)</sup>の中から探すことができる。3次元モデル透かしについてのまとまった情報源はないが, 文献などへのリンクが<http://www.kki.yamanashi.ac.jp/~ohbuchi/research/3dwmbib.html>にあるので参考にしてほしい。3次元モデル以外を対象とした電子透かしについては文献<sup>4)</sup>が詳しい。

### 参考文献

- 1) Benedens, O.: Geometry-Based Watermarking of 3D Models, IEEE CG&A, pp.46-55 (Jan./Feb. 1999)
- 2) Hartung, F., Eisert, P. and Girod, B.: Digital Watermarking of MPEG-4 Facial Animation Parameters, Computer and Graphics, Vol.22, No.4, pp.425-435, Elsevier (1998).
- 3) Kanai, S., Date, H. and Kishinami, T.: Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition, Proc. of the Sixth IFIP WG 5.2 GEO-6, pp.296-307, Tokyo, Japan (Dec. 1998).
- 4) Katzenbeisser, S. and Petitcolas, F. A. P.: Digital Watermarking, Artech House, London (2000).
- 5) Ohbuchi, R., Masuda, H. and Aono, M.: Watermarking Three-Dimensional Polygonal Models, Proceedings of the ACM Multimedia '97, pp.261-272 (1997).
- 6) Praun, E., Hoppe, H. and Finkelstein, A.: Robust Mesh Watermarking, Proc. ACM SIGGRAPH '99, pp.49-56 (1999).
- 7) Taubin, G. and Rossignac, J.: Organizers, 3D Geometric Compression, ACM SIGGRAPH '00, Course Notes #38 (2000).
- 8) Yoe, B-L. and Yeung, M. M.: Watermarking 3D Objects for Verification, IEEE CG&A, pp.36-45 (Jan./Feb. 1999)

(平成12年9月4日受付)