

3次元データへの情報の埋め込み

大瀨 竜太郎, 増田 宏, 青野 雅樹
Ryutarou Ohbuchi, Hiroshi Masuda, and Masaki Aono
ohbuchi@acm.org, masuda@trl.ibm.co.jp, aono@acm.org

日本アイ・ビー・エム東京基礎研究所
IBM Tokyo Research Laboratory

要旨

近年 Internet や CD-ROM などのデジタルメディアが普及し, 貴重なコンテンツの複製, 編集, 配布などが容易となった. この傾向は, データそのものに情報を埋め込んでコピーライトの通達や保護, 盗用の牽制, 等を実現しようとする情報埋め込み技術の研究を促した.

本論文では, 従来の情報埋め込みの研究では注目されていなかった 3次元ポリゴンモデルへの透かしの埋め込みに技術ついて, その背景, 基礎, およびいくつかのアルゴリズムを述べる. 本論文で述べるアルゴリズムは, 点, 線, 多面体などからなる広い意味での 3次元ポリゴンモデルを対象とし, その頂点座標値, 頂点間の接続関係(トポロジ), またはその両方を変更することにより情報を埋め込む.

1. はじめに

近年 Internet や CD-ROM, Digital Video Disk などのデジタルメディアが普及し, テキスト, イメージ, 動画, 音声, さらに Virtual Reality Modeling Language (VRML) に代表される 3次元(3D)モデルなどからなる貴重なコンテンツの複製, 編集, 配布などが容易となった. しかし, これは同時に, これらコンテンツの不法な複製, 編集, 配布をも容易とし, 問題となりつつある.

こうした傾向は, データそのものに情報を埋め込んでコピーライトの通達や保護, 盗用の牽制, あるいは在庫管理等に利用しようとする研究を促した. こうした情報埋め込みの技術は steganography, digital watermarking (デジタル透かし), data hiding (データ隠蔽), data embedding (データ埋め込み)等と呼ばれ, また情報を付与するためにデータオブジェクトに埋め込まれた可視または不可視の構造はしばしば (digital) watermark ((デジタル)透かし)と呼ばれる.

図 1は恐竜のポリゴンモデルに対して情報埋め込みを施した例を示す. 情報埋め込みの前後でモデルの見分けはつかない. 埋め込んだ “Copyright (C) DINOSAURS INC. <CR> Model #triceratops003 <CR> Contact http://www.dinosaurs.org/.”というテキストは, 情報を disembed (掘り出す)機能を付加した特別なブラ

ウザを使えば必要に応じて表示できる. この透かしは, 3Dモデルを実際にシーン構築に使う場合に想定される, モデルの一部切り出しやアファイン変換といった処理を施しても簡単には消えない. こうした埋め込み情報は, 例えば, ユーザの Web ブラウザを著作権管理をする Web サイトに接続し, ユーザに対してモデルの使用料の支払いを促すために用いることができる.

これまでのデータ埋め込みの研究は, 埋め込みの対象を, テキスト文書, 静止画, 動画, あるいは音声データに限定していた [Berghe196, Kobayashi96]. 本論文では, 3Dモデル, 特にポリゴンの集合からなる幾何形状への情報埋め込み技術を述べる. 3Dモデルに対する情報埋め込みの研究は, 筆者の知る限り, 本研究が初めてである.

本論文では, まず第 1.2 節で過去の手法を簡単に紹介し, 2章ではポリゴンモデルへのデータ埋め込みの基礎を, また 3章では我々の開発した具体的アルゴリズムを実行例とともに述べ, 4章でまとめと将来の展望を述べる.



図1. 本シーンの 2体の恐竜のモデルにはそれぞれ異なるコピーライト告知が埋め込んである. この情報はモデルをクリックすることにより掘り出し, 表示できる.

1.1. これまでの手法

これまで数々のデータ埋め込み手法が研究されてきた(例えば [Tanaka90, Walton95, Braudway96, Cox95, Zhao96]). しかし, 筆者らの知る限り, 埋め込みの対象はテキスト, 静止画, 動画, および音声情報に限られており, Virtual Reality Modeling Language (VRML) [ISO97] に代表されるような3Dモデルを対象とする情報埋め込みは今まで研究されていない.

3Dモデルにコピーライト等の情報を付加する手法としては, これまで, そのモデルの3Dモデルデータフォーマットの持つ, コメントやアノテーションのフィールドが使われてきた. しかし, 意図のあるなしに関わらずこの2つのフィールドは容易に取り去ることができる. 例えば3Dデータ間のフォーマット変換プログラムはしばしばコメントやアノテーションを取り去る. したがって, これらのフィールドを用いて3Dモデルに情報を付加する従来からの手法は, コピーライト保護や告知の目的には向いていない.

2. データ埋め込みの基本的手法

2.1. データ埋め込みの要件

以下3つがデータ埋め込みの重要な要件である.

(1) **Unobtrusive (邪魔にならない)**: 透かしは, 埋め込み対象となるデータオブジェクト(例えば静止画)の本来の使用目的(例えば表示して見る)を妨げてはならない. 多くの透かしは不可視であるが, その使用目的によっては可視な透かしが有効な場合もある [Braudway96].

(2) **Robust (頑強である)**: 透かしは, それが意図的なものかどうかに関わらず, 予期される妨害, 例えばアフィン変換などに耐えて残らねばならない. ここで要求される頑強さとは, 第三者に内容を読み取られないようにするのは異なることに注意してほしい. 第三者に内容を読まれないためには従来の暗号化技術(例えば [Schneier96]を参照)を用いればよい.

3Dモデルはどのような妨害を考慮する必要があるのだろうか? データフォーマット変換は頻繁に使われるし, その際に浮動小数点数の表現誤差や切り捨て誤差が導入される. モデルを含むシーンの作成の際, モデルの回転や平行移動はもちろんのこと, 一様あるいは非一様なスケール, 場合によっては歪みも加わる場合がある. モデルの一部を変形したり(例えば人体のモデルの腕を曲げる), その一部を切り取る事たまにある. 最近では, 表示や転送の高速化の為にポリゴン単純化も行われる.

理想的には, モデルの価値に支障が出ない程度の妨害を想定する場合, 透かしは, 上に挙げたものを含む多種多様な妨害全てに耐えるべきである. 現実的には, 想定する使用目的とそれに伴って予期される大多数の妨害に耐えればよいだろう. もし妨害が意図的なものなら, 妨害(モデルの価値を損なわない範囲)による透かし破壊のコストが, 護ろうとするモデルのコストと同等以上になればよい.

(3) **Space efficient (空間使用効率が良い)**: 透かしは, 現実的な量の情報を, 限られた大きさのオブジェクト(例えば, ポリゴン数を大きさの指標とする)に埋め込みねばならない. ただし, 一般に, 空間使用率は頑強さと相反する関係にある.

2.2. データ埋め込みの対象

3Dシーンはいろいろなデータタイプのオブジェクトを含みうる. VRML 2.0を例に取ると, インデクスつきポリゴンによる幾何形状の記述, 定義済みの形状(円錐等), 頂点法線ベクタ, 頂点カラー, テクスチャマッピング用の静止画や動画, あるいはJava™による行動記述, 等の複数の型のオブジェクトを含みうる. 我々は, 3Dシーンに含まれる各種データタイプのうち, 幾何形状の記述がデータ埋め込みの対象として最も適していると主張する. 幾何形状は3Dシーン記述の中で最も重要で, 従って最も取り除かれにくい為である.

幾何形状の記述法にも, ポリゴン集合, ボクセル列挙, 曲面で囲まれたソリッド, など数多くある. 本論文で紹介する研究では, 数ある幾何形状の表現法の中で, 点や線も含めた広い意味でのポリゴンの集合を埋め込みの対象(出力)とした(図2参照).

埋め込みの入力としては, 本論文では, ポリゴンに加え, Non-Uniform Rational B-Splines (NURBS) 等で表現された曲面も考慮する. これら曲面はデータ埋め

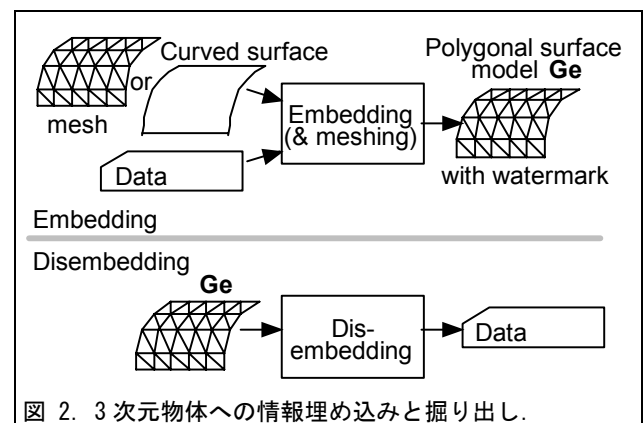


図 2. 3次元物体への情報埋め込みと掘り出し.

込みと同時にメッシュ分割し, ポリゴンの集合として出力

する。

もちろん、幾何形状の記述以外のオブジェクトも埋め込みの対象となりうる。頂点法線ベクタは、それがモデルの使用に不可欠の場合には変更が難しく、埋め込みの対象として有望である。頂点カラーや頂点ごとのテクスチャ座標も埋め込みの対象になる。また、テクスチャとして張る静止画や動画には従来のデータ埋め込み技術が適用できる。しかしこれら非幾何形状データは切除や変更が比較的容易で、また実際にモデルを使用する頻繁に変更される(例えば色を塗り替えるなど)。したがって本論文ではこれらは埋め込み対象として考察の対象に含めない。

2.3. 埋め込みの基本操作とプリミティブ

ポリゴンよりなる 3 次元物体では、3 次元物体の頂点の**座標**、または頂点間の結合関係つまり**トポロジ**、のいずれかまたは両方を変更することで透かしを埋め込む。本論文では埋め込み対象にこれらの変更を加える単位(例えば、同一直線状の 2 つの線分の長さの比)を**埋め込みプリミティブ**と呼ぶ。

2.3.1. 座標プリミティブ

座標プリミティブを用いた情報埋め込みでは、以下のような種々の(複合)量を変えて、究極的には点や頂点の座標を変更し、情報を埋め込む。下に列挙したような、あるクラスの座標変換に不変な量を埋め込みプリミティブとして使うと、そのクラスの変換に対して頑強な埋め込みを実現できる。

1. 以下のすべての変換で変わる。
 - a. 点の座標.
2. 平行移動と回転に不変。
 - a. 直線の長さ.
 - b. 多角形の面積.
 - c. 多面体の体積.
3. 平行移動, 回転, 一様スケーリングに不変。
 - a. 相似な三角形を決める 2 つの無次元量(例. 2 角).
4. アフィン変換に不変。
 - a. 1 直線状の 2 線分の長さの比.
 - b. 平行な 2 多角形の面積の比.
 - c. 2 つの多面体の体積の比.
5. 射影変換に不変。
 - a. 直線状の複比(例えば [Farin96] を参照)

一般に、座標の変更が目立たぬよう、これらの量の変更は小さく抑える事が殆どである。

2.3.2. トポロジプリミティブ

トポロジへの埋め込む場合は、例えば、4角形を 2 つの 3 角形に分割する 2 つの場合 \square と \square を 2 つのシンボルに対応させればよい。また、 \square と \square のようにメッシュの細かさを変えてシンボルやパターンを埋め込むことも出来る。

2.4. 埋め込みプリミティブの順序付け

実用的な量の情報を埋め込むためには、上で述べたような埋め込みプリミティブの集合を考え、その要素間で順序付けをし、順序の付いたプリミティブの集合全体でデータをエンコードする事が重要である。一度順序が付いた埋め込みプリミティブの集合に情報を埋め込む手法は**シンボル列埋め込み**と**パターン埋め込み**の 2 種類に分類出来る。シンボル列埋め込みでは、英文アルファベット 1 文字や 2 進数の各桁の数字(1 と 0)などのシンボルを各埋め込みプリミティブに対応させ、全体として順序のついた記号列を埋め込む。プリミティブの配置の多くは 1 次元順序であるが、2 次元以上の順序もありえる。パターン埋め込みは、可視化するとヒトが視覚的に認識できるパターンを、多くは 2 次元順序を付けた埋め込みプリミティブ集合に埋め込む。

シンボル列埋め込みであろうとパターン埋め込みであろうと、プリミティブの間で順序をつける方法には以下の 2 つの手法がある。

- a. **トポロジによる順序付け**:埋め込みプリミティブのトポロジ的な隣接関係(例えば、3 角メッシュ上の頂点の隣接関係)を使ってプリミティブ間の順序を付ける。例えば、3 角メッシュ上の頂点の minimal spanning tree (最小展開木)を使う。本手法は座標及びトポロジプリミティブに適用でき、その順序は座標変換に不変であるが、トポロジを変える操作、例えばメッシュの単純化や切り取り等により順序は壊れる。
- b. **量による順序付け**:埋め込みプリミティブのあらゆる量(例えば、2 つの多面体の体積比)の間の大小関係を用い、プリミティブをソートして配置する。本手法は座標プリミティブにのみ適用できる。予期される座標変換で不変なプリミティブを用いないと順序が変わる。

量、トポロジいずれによる順序付けを行った場合でも、ユニークでかつ予期される妨害に対して不変な順序付けの為の**初期条件**を見つけることが重要である。初期条件を見つける手法には多くのやり方がありうるが、その一つを後に第 3.2 節に述べる。

順序付けの手法はまた、その局所性から、**グローバル順序付け**、**ローカル順序付け**、**添え字順序付け**の3つに分類できる。

- グローバル順序付け:**一つの埋め込み対象に含まれるすべての埋め込みプリミティブを順序付ける。一般に、他の2手法に比べてこの手法の空間利用効率は高いが、順序が壊されやすい弱点がある。
- ローカル順序付け:**一つの埋め込み対象に含まれる埋め込みプリミティブを複数の独立なサブセットに分割し、それぞれのサブセット内で順序を付ける。各サブセットは、埋め込みプリミティブ間の近さ(例えばユークリッド空間的近さ)をもとに作られる。
- 添え字順序付け:**ローカル順序付けに類似するが、そのサブセットが非常に小さい(例えば数プリミティブ程度)。このサブセットを *macro embedding primitive* (マクロ埋め込みプリミティブ)と呼ぶ。マクロ埋め込みプリミティブには、データをエンコードするシンボルとペアで、そのシンボルの順序を示す添え字も同時に埋め込む。この添え字によりシンボルの集合が順序付けされる。

後者2手法の近傍の近傍の測度としては、トポロジ的(例えば3角形メッシュにおける頂点の隣接性)または量的(例えば面積の比の値やユークリッド距離)のいずれを用いてもよい。

3. 埋め込みアルゴリズム

本章では、基礎技術、すなわち埋め込みプリミティブや順序付けに関する技法を組み合わせで作られたアルゴリズムをその実行例と共に紹介する。

ここで述べるアルゴリズムはすべて非多様体モデラのカーネル部 [増田 96]をもとにして実装された。このカーネルは radial edge 構造 [Weiler86] を用いて頂点、辺、面、および領域の間のトポロジカルな関係を記述しており、その関係に対する操作を効率よく行うことができる。

3.1. 相似三角形4つ組みへのシンボル列埋め込み

本節で述べるアルゴリズムは、相似な3角形を決める二つの無次元量に情報を埋め込む。例えば図3で $\{b/a, c/a\}$, $\{S/a^2, b/c\}$, 及び $\{t1, t2\}$ 等が相似3角形を決める2つの無次元量である。Sは3角形の面積、t1, t2は角度、また a, b, cは3辺の長さである。このアルゴリズムを本論文では Triangle Similarity Quadruple (TSQ)アルゴリズムと呼ぶが、これは、図4に示すような、

隣接する3角形の4つ組みをマクロ埋め込みプリミティブとして使うためである。これら4つの3角形の、2つにデータ、1つに添え字、残りの1つにマーカを埋め込む。マーカにはその4つ組みがマクロ埋め込みプリミティブである事を示す特別の値を埋め込む。図5はTSQアルゴリズムで透かしを入れたモデルであるが、マクロ埋め込みプリミティブを成す3角形の4つ組みは濃灰色、その他の3角形は薄灰色でレンダリングしてある。

本アルゴリズムによって埋め込まれた透かしはモデルの平行移動、回転、及び一様スケーリング(x, y, z各軸の拡大・縮小率が同じ)を含む変換に対して頑強である。さらに添え字順序付けを使い、同一データを繰り返して埋め込むと、部分の切り取りや局所変形に対してもかなり頑強さを増す。しかし、この透かしは、より一般的な変換(例えばずれ変形を含む変換)を加えたり、頂点座標値に乱数を重畳したりすると壊れる。

埋め込みアルゴリズムは、入力である3角形メッシュをスキャンし、未だ埋め込みに使われていない、互いに隣接する3角形の4つ組みをマクロ埋め込みプリミティブ候補として選ぶ。ここで、チェックを行い、あるマクロ埋め込みプリミティブ候補の3角形の1つ以上が埋め込

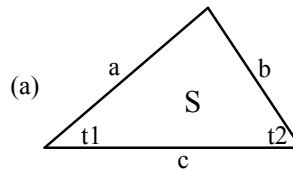


図3. $\{b/a, c/a\}$, $\{S/a^2, b/c\}$, $\{t1, t2\}$ 等の無次元量ペアが相似3角形を決める。

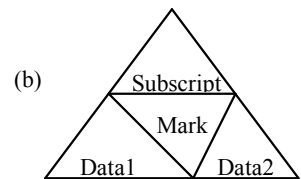


図4. ここで用いたマクロ埋め込みプリミティブは3角形4つより成る。



図5. TSQアルゴリズムでデータを埋め込んだモデル。マクロ埋め込みプリミティブをなす3角形の4つ組みは濃い灰色で示されている。

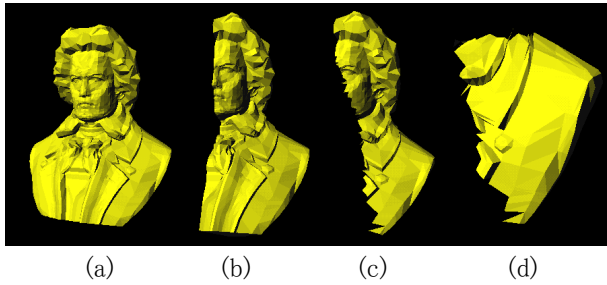


図6. 長さ 132 バイトのメッセージを 6 回繰り返して埋め込んだベートーベンの胸像のモデル (3 角形数 4889) を適当な平面で切り取っていった。

	3角形数	データ
(a)	4889	オリジナル. (132 バイトを 6 回反復して埋め込んだ)
(b)	2443	オリジナルが残っている. (132/132 バイトが残った)
(c)	1192	部分的に破壊された. (102/132 バイトが残った)
(d)	399	部分的に破壊された (85/132 バイトが残った)

表1. 図 6 に示すモデルの部分の切り取りによるデータのロス.

モデル	3 角形数	埋め込めた	埋め込み
Cow	5804	1062	19.8
Triceratops	5604	966	9.9
Beethoven	4889	873	9.7
IBM mesh	2996	486	9.8
Face	1406	210	2.3
Stegosaur	1023	180	2.5
Sphere	959	132	1.4

表2. TSQ アルゴリズムでいろいろなモデルに埋め込めるデータ量と埋め込み処理時間.

みに適さないと判断されると(例えば, 3 角形が縮退している, 小さすぎる, などの理由)その候補は棄却される. データ, 添え字, マーカの埋め込みは, このテストをパスしたマクロ埋め込みプリミティブを成す各 3 角形の頂点 (達¹) を動かし, ひいては相似 3 角形達を決める無次元量のペアを変更して行く.

掘り出しアルゴリズムは, 3 角形メッシュをスキャンし, まずマクロ埋め込みプリミティブ達を示すマーカー値が

¹ 本論文では, 以降, ある対象が複数なことを示したい場合, それが人か物かに関わらず「達」を付ける.

埋め込まれた 3 角形達を探す. 次いで見つかったマクロ埋め込みプリミティブそれぞれからデータと添え字を読み出す.

図 6a は TSQ アルゴリズムを使って 132 バイトのメッセージを 6 回反復して埋め込んだベートーベンの胸像のモデル (3 角形数 4889, 頂点数 2655) である. 表 1 に見られるように, モデルを約半分にした段階ではメッセージは保たれたが, 約 4 分の 1 にすると 132 バイト中 102 バイトだけが壊されずに残った. ただし, 添え字順序付けが用いられているため, メッセージの一部が壊された場合でも残った文字は正しい位置に残る事が多い.

表 2 は TSQ アルゴリズムでいくつかのモデルに透かしを入れた場合の, 埋め込めるデータ量と埋め込み処理に要する時間を示す. 掘り出しに要する時間は埋め込みに要する時間とほぼ等しいので表には含めなかった. 実行時間は IBM AIXTM 4.1.4 オペレーティングシステムのもとで 100MHz の PowerPC 604 で実行して測定した.

埋め込めるデータ量は頑強さとのトレードオフで変わるので, この表は参考として見て欲しいが, この程度のデータ量が埋め込めればかなり使い道が有りそうだ. 実行時間はほぼモデルの 3 角形数に比例した. この程度の速度でも用途によっては十分使えるだろう. 現在プログラムの実行速度最適化は行っていないが, もし行えば速度向上が期待される.

3.2. 4 面体の体積比へのシンボル列埋め込み

本節では, 2 つの 4 面体の体積の比を用いてシンボル列を埋め込む Tetrahedral Volume Ratio (TVR) アルゴリズムを述べる. 本アルゴリズムは, 埋め込みプリミティブ達を, トポロジを用いてグローバル順序付けする. 用いた埋め込みプリミティブがアファイン変換に対し不変なため, TVR アルゴリズムで埋め込まれた透かしはアファイン変換に耐える. しかし TVR アルゴリズムはトポロジの変更 (例えば切り取り) や, 乱数値の座標値への重量により破壊される. 本節の最後に, 多少手を加えて切り取りに対して頑強さを増したアルゴリズムも紹介する.

TVR アルゴリズムは, 以下の手順を踏んで, 座標プリミティブ達に対してトポロジを用いた 1 次元のグローバル順序付けをし, シンボル列を埋め込む.

- (1) ある初期条件から初めて, 入力されたメッシュの頂点達の spanning tree (展開木) を求める. この展開木は頂点木 (vertex tree) と呼ばれる [Taubin96]. 初期条件は予期される外乱に対して頑強でなければならない. もし初期条件が変わると透かしは壊れる.

- (2) 頂点木を, 順序付けられた 3 角形達の列, triangle sequence (3 角形列)に変換する.
- (3) 3 角形列を順序を持つ 4 面体達の列 tetrahedron sequence (4 面体列)に変換する. まず, 適当に選んだ数個の3角形達の頂点達の重心を求め, これを共通の頂点として, 各 3 角形を 4 面体にする. これで 3 角形列が 3 面体列に成る. ただし, 共通の頂点(重心)を求めるのに使った数個の3角形は予め3角形列から取り除き, 4面体列に含めず, 従って頂点移動による埋め込みにも使わない.
- (4) 4面体列を, 順序を持つ比達の列に変換する. 各比は2つの4面体の体積達の比である. これには, 4面体列のうちどれか1つの4面体(例えば最初の4面体)を全ての比達の共通の分母として選ぶ. 次いでこの共通分母の4面体の体積で(4面体列の)残りの4面体の体積達を割って比の列を作る.
- (5) 比の列の比を微少に変えてシンボルを埋め込む. 比の値を変えるには分子となった4面体をなす頂点の座標達を動かす. この時, 埋め込みシンボル達の間で干渉があってはならない. つまり, ある4面体の体積の変更はその他の4面体達の体積に影響を与えてはならない.

図7はTVRアルゴリズムで透かしを入れたモデルである. 埋め込みに使った3角形達—濃い灰色でレンダリングされている—の間が空いているが, これは, 上記(5)で述べたように, 4面体同士の相互干渉を避けている為である.

最初の 2 ステップは少々入り組んでいるのでもう少し説明を加える. まず, この 2 ステップでは, 入力メッシュの全ての 3 角形が向き付けされている(メッシュが向き付け可能な多様体である)事が必要条件である. 以後, まず頂点木の生成法, およびこれを 3 角形列に変換する方法を説明し, その後で頂点木生成の初期条件の決めかたを述べる.

頂点木は, まず空白に初期化し, 初期頂点から初期トラバース方向へ入力 3 角形メッシュの頂点達のトラバースを開始する. トラバースの課程である頂点に到達すると, その頂点に隣接する頂点達を(一つ前の頂点と注目する頂点を結ぶ辺から始めて)反時計方向の順で調べ, その中で最初に現れかつ未だ頂点木に含まれない頂点を新たに頂点木に追加する. 次にその新たに追加した頂点を次の注目する頂点として繰り返す. 次にたどれる頂点が無くなってトラバースが行き詰まったら, 複数の選択肢があった頂点(1つ以上の隣接する頂

点が未だ頂点木に含まれないような頂点)にバックトラックしてトラバースを進める.

図 8の例では, 頂点木の根は頂点1で(頂点1は又分岐点でもある), トラバースは頂点 1-から順に 9 へ進んだ後で1にバックトラックし, 11-12 とたどって終了する.

頂点木の3角形列への変換は 3 角形を作る辺を登録する Triangle Bounding Edge (TBE; 3角形境辺)リストを成長させつつ行う. TBEを成長させるには, まず, 頂点木に含まれる頂点達を結ぶ辺達に TBE を初期化する. 次いで, 頂点木を根からトラバースし, 訪れた各頂点で, その頂点に隣接する辺達を時計周りに調べ, 選んだ辺が未だ TBE リストになければ, その辺を TBE リストに追加する. 3角形列は空白に初期化する. 次いで, TBE リストを成長させつつ, TBE リスト中の辺達と新たに TBE リストに追加した辺とによって初めて作られる3角形を, 3角形列に加える.



図7. TVR アルゴリズムで透かしを埋め込んだ例. 埋め込みに使われた 3 角形達は濃い灰色でレンダリングされている.

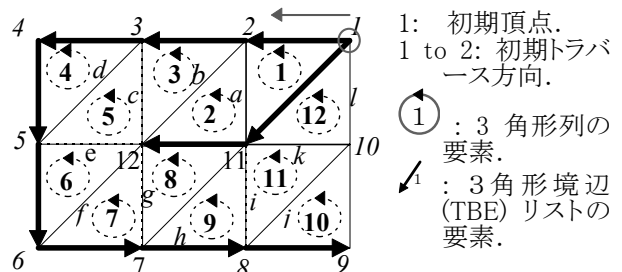


図8. 簡単な 3 角形メッシュ上に作られた頂点木, 3 角形境辺 (TBE) リスト及び 3 角形列の例.

図 8の例では, TBE リストに追加された辺には, 追加の順にアルファベットをふった. ただし, TBE リストの要素でも, その初期状態での要素, つまり頂点木をつな

ぐ辺には何もふっつてい無い。

ここで少々立ち戻り、頂点列生成の初期状態の決めかたを説明する。求めたい初期条件は初期頂点とそこからの初期トラバース方である。しかし、TVR アルゴリズムでは、以下の手順で向きの無い**初期辺**をもとめ、これを初期条件として使う。初期辺では初期トラバースの向きが確定せず(2方向いずれかが可能)曖昧さが残る。この曖昧さの解決はいわゆる trial-and-error 方式で行った。つまり、2つの初期トラバースの向きでデータを掘り出してみ、あらかじめ定めた先導シンボル列が掘り出せた向きが正しい向きであるとする。

初期辺を決めるには、与えられた3角形メッシュの全ての辺のそれぞれについて、その辺を挟む2つの3角形達で決まる4面体の体積を計算する(図 10参照)。こうして求めた体積達を比べ、その中で最大の体積を持つ4面体を作った辺を初期辺とする。勿論これだけでは計算エラーやノイズ等により誤った辺が初期辺として選ばれる可能性がある。この曖昧さの解決にも再び trial-and-error 方式を用いた。つまり、複数の初期辺の候補達を使ってデータを掘り出し、正しい先導シンボル列を掘り出せた初期辺を正しい初期辺とする。

表3はTVRアルゴリズムでいくつかのモデルに透かしを入れた場合の、埋め込めるデータ量と、埋め込み処理に要する時間を示す。表2同様、掘り出しに要する時間は埋め込みに要する時間とほぼ等しいので表には含めなかった。処理時間の測定は表2の場合と同じ条件で行った。

表2と比べるとわかるように、TSQ アルゴリズムとTVR アルゴリズムの間で埋め込めるデータ量や実行時間はほぼ同じである。実行時間については、メッシュのトラバースの手間が大勢を占めている。TVR アルゴリズム

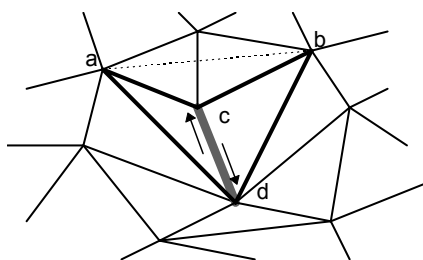


図10. 初期条件である初期辺を決めるため、4面体 a-b-c-d (初期辺 c-d を共有する2つの3角形 a-d-c と b-c-d で作られる)の体積を求める。辺 c-d の二つの矢印は、初期辺から考えられる2つの初期トラバース方向を示す。

ムにおいても、埋め込みデータ量、実行時間も多くの用途で実用に耐えると考える。

TVR 方による透かしに、グローバル順序付けの代

わりにローカル順序付けまたは添え字順序付けを採用し、メッセージを反復して埋め込むことにより、局所変形や部分の切取りに対しての頑強性を増す事が出来る。このようにして頑強性を増したアルゴリズムを Triangular Volume Ratio Cluster (TVRC)アルゴリズムと呼ぶ。TVRC アルゴリズムは入力メッシュを、トポロジカルな距離による近傍を用いて複数のサブメッシュに分け、それぞれのサブメッシュに対してTVR法を適用する。図9に示す例は牛のモデル(図9a)にTVRC法でメッセージを

モデル	3角形数	埋め込めたデータ量 [bytes]	埋め込みに要した時間[sec]
Cow	5804	1027	20.2
Triceratops	5604	650	7.2
Beethoven	4889	324	9.6
IBM mesh	2996	652	9.9
Face	1406	116	2.2
Stegosaur	1023	225	2.4
Sphere	959	216	1.5

表3. Tetrahedral Volume Ratio (TVR) アルゴリズムでいろいろなモデルに埋め込めるデータ量と埋め込みの処理時間。



(a) The original model.

(b) Message embedded.



(c) Cut in half.

(d) Affine-transformed.

図9. (a) 牛のモデル (3角形数 5804) . (b) TVRC アルゴリズムでメッセージ (画面に重ねて表示してある) を埋め込んだ後. このメッセージは (c) 部分の切取りや (d) アフライン変換に耐えて生き残った.

埋め込み(図9b), その一部を切取ったり(図9c)全体にアフライン変換を掛けたり(図9d)してもメッセージが壊されずに残ることを示す。

4. まとめ

本研究で我々はデータ埋め込みの考え方を3次元ポリゴンモデルに対して適用した。現在まで、データ埋め込みの対象となるオブジェクトはテキスト、静止画、動画、および音声信号に限られていた。3次元形状モデルに対するデータ埋め込みは、筆者らの知る限り、本研究が最初と思われる。

本論文では、まず3次元形状モデルに対するデータ埋め込みの背景および要件を述べ、3次元形状モデルに含まれる各種のデータオブジェクトの中で幾何形状がデータ埋め込みの対象として最も適していると論じた。さらに、幾何形状の中でも、ポリゴンモデルを埋め込みの対象として選んだ。次いで、基本的な埋め込みの手法、頂点座標値や頂点間の接続関係を変える埋め込みプリミティブや、複数の埋め込みプリミティブを順序付ける手法を述べた。その後で具体的な埋め込みアルゴリズムをいくつかの実行例と共にのべ、3次元ポリゴンモデルに対する情報埋め込みが実現可能なことを示した。

筆者らは、本論文で紹介したアルゴリズムでも、その使用目的によっては使えると考える。例えば「このモデルのコピーライトは xxx にあります。ライセンス料を yyy までに払ってください。」といったコピーライト告知メッセージの表示には使えるだろう。しかし、その他の多くの用途にとって、本論文で紹介したアルゴリズムでは頑強さが不十分である。例えばモデルの盗用を証明する用途にはこれらのアルゴリズムは使えないだろう。

将来の研究課題としては、まず頑強さの向上があげられる。また、埋め込みの対象となるデータ型を曲面や頂点法線ベクタなどに広げる為の研究も必要であろう。

参考文献

- [Berghel96] H. Berghel and L. O’Gorman, Protecting Ownership Rights Through Digital Watermarking, *IEEE Computer*, July 1996, pp.101-103.
- [Braudway96] G. Braudway, K. Magerlein, and F. Mintzer, Protecting Publicly-Available Images with a Visible Image Watermark, *IBM Research Report*, TC-20336 (89918), January 15, 1996.
- [Cox95] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *Technical Report 95-10*, NEC Research Institute, 1995.
- [Farin96] G. Farin, *Curves and Surfaces for CAGD: a Practical guide*, 4th edition, Academic Press, 1996.
- [ISO96] ISO/IEC JTC1 SC24/N1596 CD #14772 Virtual Reality Model Language (VRML 2.0)

[Kobayashi96] M. Kobayashi, The Electronic Hare: Mathematics and Image Marking, *SIAM News*, October 1996.

[増田 96] 増田 宏, “非多様体形状モデルのための形状操作とその応用に関する研究”, 博士論文, 東京大学工学部 (1996) .

[O’Rourke94] J. O’Rourke, *Computational Geometry in C*, Cambridge University Press, 1994.

[Schneier96] B. Schneier, *Applied Cryptography*, Second edition, John Wiley & Sons, Inc., New York, 1996.

[Tanaka90] K. Tanaka, Y. Nakamura, and K. Matsui, Embedding Secret Information into a Dithered Multilevel Image, *Proc. 1990 IEEE Military Communications Conference*, pp. 216-220, 1990.

[Taubin96] G. Taubin and J. Rossignac, Geometry Compression Through Topological Surgery, *IBM Research Report*, RC-20340 (89924), January, 1996.

[Walton95] S. Walton, Image Authentication for a Slippery New Age, *Dr. Dobb’s Journal*, April 1995.

[Weiler86] K. Weiler, The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling, *Geometric Modeling for CAD Applications*, North Holland, pp. 3-36, May 1986.

[Zhao96] J. Zhao and E. Koch, Embedding Robust Labels into Images for Copyright Protection, *Proc. of the Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge, and New Technologies*, Vienna, August 1995.