

# スペクトル分解を用いた 3 次元メッシュへの電子透かしの埋め込み

大 淵 竜 太 郎<sup>†</sup> 高 橋 成 雄<sup>††</sup>  
宮 澤 貴 彦<sup>†</sup> 向 山 明 夫<sup>†</sup>

電子透かしとは、対象となるデータに何らかの構造を付加することにより、そのデータにメッセージを埋め込む技術で、その応用には著作権保護、改ざん検出などがある。本論文では、3次元ポリゴンメッシュに対しそのメッシュスペクトルの領域で電子透かしの埋め込み手法を提案する。本手法では、まず、メッシュの頂点の接続関係のみから得られるラプラシアン行列を求め、それを固有値分解して得られた固有ベクトルを正規化したものに頂点座標値を射影することでメッシュのスペクトルを求める。次いで、このメッシュスペクトルを透かし情報に基づいて変更して透かしの埋め込み。本手法で得られる電子透かしは相似変換（回転、並行移動、一様スケールリング）に対して頑強で、メッシュのスムージングや頂点座標へのランダムノイズの重畳、さらにはメッシュの部分的切り取りに対してもある程度の頑強性を示す。

## Watermarking 3D Polygonal Meshes Using Mesh Spectral Analysis

RYUTAROU OHBUCHI,<sup>†</sup> SHIGEO TAKAHASHI,<sup>††</sup> TAKAHIKO MIYAZAWA<sup>†</sup>  
and AKIO MUKAIYAMA<sup>†</sup>

Digital watermarking embeds structures called watermark into the target data, such as image and 3D polygonal models. The watermark can be used, for example, to enforce copyright and to detect tampering. This paper presents a new watermarking technique that adds watermark into a 3D polygonal mesh in the mesh's spectral domain. Our watermarking algorithm starts with eigenvalue decomposition of a Laplacian matrix, which is a matrix derived only from the connectivity of the polygonal mesh. Mesh spectra can then be obtained by projecting coordinates of vertices onto the set of normalized eigenvectors. Watermarks are embedded by modifying the magnitude of the spectra. The watermarks embedded by using this method are resistant to similarity transformation, random noise added to vertex coordinates, mesh smoothing, as well as to partial resection of the meshes.

### 1. はじめに

データ埋め込み、または電子透かしと呼ばれる技術は、情報を表現する watermark (透かし) と呼ばれる構造体を、透かし埋め込みの対象となるコンテンツデータに付加する。ここで、透かしの存在が埋め込み対象コンテンツの本来の目的 (たとえば人による表示・鑑賞) を阻害しないこと、かつ、透かしがコンテンツから容易には除去できないこと、が要件である。透かしに埋め込まれた情報 (「透かし情報」) は、説明の追加、改ざんの検出、あるいは正規の購入者の認証など、そのコンテンツを何らかの形で管理する目的で用

いられる。

電子透かしにはいくつかの分類法があるが、その中でも重要なものに頑強性と公開性がある。著作権保護などを目的とし、極力妨害に耐えて残ることを目標とするのが頑強な透かし (robust watermark) である。これに対し、積極的に透かしが壊れることで改ざんなどを検出する、脆弱な透かし (fragile watermark)<sup>18)</sup> も存在する。透かしの取り出しに、透かしの入ったデータ (「透かしデータ」) と、透かしの入っていない元データ (「被覆データ」) の両方を必要とするのが秘密透かし (private watermark) である。これに対し、透かしデータだけで透かしを取り出せる手法を公開透かし (public watermark) と呼ぶ (秘密透かし、公開透かしの透かしの取り出しは、それぞれ non-blind detection および blind detection とも呼ばれる)。公開透かしの方が実用上の制限が少ないが、頑強な透かしの実現が秘密透かしに比べてより困難な場合が多い。

<sup>†</sup> 山梨大学工学部コンピュータ・メディア工学科  
Computer Science Department, Yamanashi University

<sup>††</sup> 東京大学大学院総合文化研究科広域システム科学系  
Department of General Systems Studies, Graduate  
School of Arts and Sciences, The University of Tokyo

これまでの電子透かしの研究のほとんどは「古典的」マルチメディアデータ型、たとえば文字文書、静止画像、動画像、および音声データに対する埋め込みを中心としていた。これらについては非常に多くの論文といくつかの著書<sup>8),19)</sup>が発表されている。しかし最近、VRML や MPEG4、さらには多様な CAD データなどの 3 次元 (3D) データがその重要性を増してきたのを反映し、3D モデルを対象とする電子透かしの研究が行われるようになった。

3D モデルは形状 (ポリゴンメッシュやパラメトリック曲面など) やその属性 (頂点の色やテクスチャ座標)、動き (MPEG4 のアニメーションパラメータ) などの多種のデータオブジェクトからなり、それぞれを対象とする透かしアルゴリズムが発表されている。3 次元モデルの電子透かしについては大淵の解説<sup>20)</sup>がある。

形状は、3D モデルの最も重要な属性であり、3D モデルを対象とする電子透かしの研究においても最も重要な透かし埋め込みの対象である。3 次元の形状を対象とする電子透かしの多くはポリゴンメッシュの形状を埋め込みの対象としている<sup>3),6),11)~13),15),17),18)</sup>。これらは頂点座標値が頂点の接続性、あるいはその両方を変更して透かしを埋め込んでいる。このほか、機械 CAD など広く用いられているパラメトリック曲面を対象とし、その形状をまったく変更せずに透かしを埋め込む手法<sup>14)</sup>も報告されている。

本論文で提案する電子透かし手法では、まず、ポリゴンメッシュで定義される 3 次元形状の頂点接続性より定義されるメッシュ Laplacian (ラプラシアン) 行列をスペクトル分解する。次いで、こうして得られたスペクトルを、埋め込む透かし情報に応じて変更することで、スペクトル領域において電子透かしを付加する。本論文の透かしは妨害に耐えて残ることを目指した頑強な透かしである。また、取り出しに被覆メッシュを必要とするので秘密透かしに分類される。

本論文の手法で得られる電子透かしはメッシュに加えられる相似変換 (回転、並行移動、一様スケージングの組合せ) に対して頑強である。また、埋め込みデータを反復し拡散して埋め込むことで頂点座標へのランダムノイズの重畳に対してある程度の耐性を持ち、透かしを埋め込む「周波数帯域」を選ぶことにより、形状のローパスフィルタリングに相当するメッシュのスムージングに対してもある程度の耐性を持つ。さらに、メッシュを小領域に分割し、分割数だけ繰り返して同じ透かしを埋め込むことにより、メッシュの部分的切り取りに対する耐性も持たせることができる。

本論文の構成は以下のとおりである。まず、2 章で

は、メッシュのスペクトル分解と、それに基づいた電子透かしの埋め込みおよび取り出しアルゴリズムを述べる。3 章では実装と実験の結果を紹介し、4 章で関連研究を、5 章でまとめと今後の課題を述べる。

## 2. メッシュのスペクトル分解による電子透かし

本論文で述べる電子透かし手法は、頂点座標と頂点の接続関係により形状が定義される 3 次元ポリゴンメッシュに対し、頂点座標の変換領域で透かしを付加する。用いられる変換は、ポリゴンメッシュの接続関係より定義されるメッシュラプラシアン行列を基にしたポリゴンメッシュ形状のスペクトル分解である。

スペクトル分解の結果は、だまかにいって、小さな固有値 (低い周波数に対応) とその固有ベクトルがメッシュの概形を、大きな固有値 (高い周波数に対応) とその固有ベクトルがメッシュの詳細を、それぞれ表現すると考えられる。本論文の電子透かし手法では、スペクトル係数の振幅を透かし情報に応じて変更して透かしを埋め込み、変更したスペクトル係数から透かしの入った形状を再構成する。本透かし手法は秘密透かしであり、取り出しには透かし入りの透かしメッシュと透かし付加前の被覆メッシュが必要である。透かしの取り出しは、透かしメッシュと被覆メッシュをともにスペクトル分解し、そのスペクトル係数を比較して行う。

### 2.1 メッシュのスペクトル分解

メッシュスペクトルは、頂点の接続情報で定義されたメッシュラプラシアン行列に基づく。この行列を固有値分解し、得られた固有ベクトルを正規化して基底ベクトルを得る。この基底ベクトルに、メッシュの頂点座標を成分ごとに射影して得られた係数がメッシュスペクトルである。

メッシュのラプラシアン行列にはいくつかの定義があるが<sup>1),2)</sup>、我々は Bollobás<sup>2)</sup> のメッシュラプラシアン [別名 Kirchhoff (キルヒホフ) 行列]  $K$  を用いた。以後、本論文では、Bollobás のメッシュラプラシアン行列をキルヒホフ行列と呼ぶ。

キルヒホフ行列  $K$  は

$$K = D - A. \quad (1)$$

で定義される。 $A$  はポリゴンメッシュの頂点の隣接行列で、以下のように定義される。

$$A_{ij} = \begin{cases} 1 & \text{頂点 } i \text{ と } j \text{ が隣接,} \\ 0 & \text{その他.} \end{cases} \quad (2)$$

$D$  は対角行列で、その対角要素  $D_{ii} = d_i$  は頂点  $i$  の次

数である．Karni らが形状圧縮に用いた  $L = I - HA$  がある<sup>7)</sup>．ここで  $A$  は上記の式 (2) と同じで，対角行列  $H$  は対角要素に頂点  $i$  の次数の逆数  $H_{ii} = 1/d_i$  を持つ．

我々はキルヒホフ行列  $K$  を透かしの埋め込みに用いた．この選択に際し，我々は，2つの行列で求まるスペクトルの振舞いを見るための実験を行った．我々は Karni らの頂点座標値圧縮アルゴリズム<sup>7)</sup> を実装し，複数のメッシュについて圧縮率を変えたときの形状近似誤差  $\varepsilon$  の変化を調べた．その結果，モデルや圧縮率で前後するものの， $K$  と  $L$  でほぼ同様の傾向を示した．ここで  $\varepsilon$  とはもとのモデルと圧縮されたモデルの頂点座標値の2乗平均誤差である．実験の結果，2つの行列の振舞いがほぼ同様のため，我々は，より安定で効率の良い固有値分解アルゴリズムが存在するキルヒホフ行列  $K$  を使用することとした（対称行列の  $K$  には非対称行列の  $L$  よりも安定で効率的な固有値分解手法を適用できる）．

$n$  個の頂点からなるメッシュから定まる  $n \times n$  のキルヒホフ行列  $K$  を固有値分解すると， $n$  個の固有値とそれに対応する  $n$  個の  $n$  次元固有ベクタ  $w_i$  ( $1 \leq i \leq n$ ) が得られる．この固有ベクタを正規化すると基底ベクタ  $e_i$  が得られる．

$$e_i = w_i / \|w_i\| \quad (1 \leq i \leq n). \quad (3)$$

$n$  個の頂点座標  $v_i = (x_i, y_i, z_i)$  ( $1 \leq i \leq n$ ) を，その  $x, y, z$  各成分ごと， $i$  番目の基底ベクタ  $e_i$  に射影すると，メッシュのスペクトル  $r_i = (r_{s,i}, r_{t,i}, r_{u,i})$  ( $1 \leq i \leq n$ ) が得られる．ここで  $s, t, u$  はメッシュスペクトル領域での直交座標で，空間領域の  $x, y, z$  に対応する．

スペクトルから頂点座標値へ逆変換するには，以下式 (4) のように，固有ベクタ  $e_i$  をスペクトル係数  $r_{s,i}, r_{t,i}$  および  $r_{u,i}$  倍して線形和を計算する．こうすると，もとの形状の頂点座標値  $v_i = (x_i, y_i, z_i)$  の成分  $x_i, y_i$ ，および  $z_i$  を得られる．

$$\begin{aligned} & (x_1, x_2, \dots, x_n)^T \\ &= r_{s,1}e_1 + r_{s,2}e_2 + \dots + r_{s,n}e_n, \\ & (y_1, y_2, \dots, y_n)^T \\ &= r_{t,1}e_1 + r_{t,2}e_2 + \dots + r_{t,n}e_n, \\ & (z_1, z_2, \dots, z_n)^T \\ &= r_{u,1}e_1 + r_{u,2}e_2 + \dots + r_{u,n}e_n. \end{aligned} \quad (4)$$

メッシュラプラシアン行列の固有値分解は，大まかにいって，機械部品などのメッシュ表現に対して振動モード解析を施すことに相当する．固有値はメッシュが変形する1つの振動モードの「周波数」に，また（正規化した）固有ベクタはメッシュの変形を表現す

る基底ベクタに相当する．ある固有値のスペクトル係数が変わると，対応する固有ベクタに応じてメッシュが変形し，頂点座標値が変化する．

## 2.2 透かしの埋め込み

本論文の透かし手法は，メッシュの接続性から導かれるキルヒホフ行列を固有値分解して求めたスペクトルを変更して透かしを埋め込む．このような順序付けされた数値を変更して透かしを埋め込む場合，これまでしばしば帯域拡散通信 (spread spectrum communication) の考え方が採用されてきた<sup>8)</sup>．かく乱に強くかつ単位周波数あたりの電力が低いという帯域拡散通信の特徴は，種々の妨害に対し頑強かつ不可知でなければならない電子透かしの要求と合致しているからである．我々の手法も帯域拡散通信の手法を採用しており，具体的には Hartung らが用いた手法<sup>5)</sup>に近い．

我々の透かしアルゴリズムが埋め込む透かしデータは  $m$  次元のビットベクタ  $a = (a_1, a_2, \dots, a_m)$  (各ビットは  $\{0, 1\}$  の値をとる) である．各要素  $a_j$  はそれぞれ chip rate (拡散率)  $c$  の回数複製され， $a$  は長さ  $m \cdot c$  の埋め込みシンボルベクタ  $b = (b_1, b_2, \dots, b_{mc})$  に変換される．

$$b_i = a_j, \quad j \cdot c \leq i < (j+1) \cdot c. \quad (5)$$

同じビットを  $c$  回繰り返して埋め込むことでランダムノイズなどに対する耐性を高める効果がある．

次いで， $b$  は  $\{-1, 1\}$  の値をとる透かしシンボルベクタ  $b' = (b'_1, b'_2, \dots, b'_{mc})$

$$b'_i = \begin{cases} -1 & \text{if } b_i = 0, \\ 1 & \text{if } b_i = 1. \end{cases} \quad (6)$$

の列に変換される．

頂点数  $n$  のポリゴンメッシュをスペクトル分解して得られたスペクトル係数を  $r_i = (r_{s,i}, r_{t,i}, r_{u,i})$  ( $1 \leq i \leq n$ )，透かし埋め込み鍵 (整数値)  $k_w$  を種として生成された  $\{-1, 1\}$  の値をとる2値の擬似乱数系列を  $p_i$ ，変調振幅を  $\alpha$  ( $\alpha > 0$ ) とする．透かし情報によって変更されたスペクトル係数の  $s$  成分  $\hat{r}_{s,i}$  は次のように計算される．

$$\hat{r}_{s,i} = r_{s,i} + b'_i \cdot p_i \cdot \alpha. \quad (7)$$

同様の操作を  $t, u$  成分についても行うと，透かし情報が埋め込まれたスペクトル係数  $\hat{r}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$  が得られる．さらに，このスペクトル係数  $\hat{r}_i$  を式 (4) を用いて逆変換すると透かしの入ったポリゴンメッシュモデルの頂点座標値  $\hat{v}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$  が得られる．

2次元静止画像などでは埋め込み対象となる数値が少なくとも数千個は存在するため  $c$  を大きくとることができる．しかし我々の場合，スペクトル係数の数が

数百個（たとえば、猫のモデルで 353 個）から 1000 個程度なので、 $c$  をあまり大きくすることはできない。整数値の透かし鍵  $k_w$  は擬似乱数系列の種で、取り出しの際にも必要となるため、公開するか、または、何らかの手段を用い、埋め込み側と取り出し側の間で秘密裏に受け渡しをする必要がある。変調振幅  $\alpha$  は、埋め込むポリゴンメッシュの axis-aligned Bounding Box (Bbox) を計算し、その辺の長さの最大値  $\phi$  に対する振幅率  $\beta$  により  $\alpha = \phi \cdot \beta$  として指定する。変調振幅  $\alpha$  は、モデルの見かけが変わらない程度に小さく、かつノイズやスムージングなどの妨害で透かしが壊れないように大きく、選ぶ必要がある。

透かしの埋め込み対象となるメッシュの頂点数が  $n$  個なら、このメッシュをスペクトル分解して得られる固有値、固有ベクタ、およびスペクトル係数はそれぞれ  $n$  個で、単純には  $n$  ビットが埋め込める計算である。しかし、 $n$  個のスペクトル係数すべてを透かしの埋め込みに用いるわけではない。固有値の小さい側から 5 個の固有ベクタは、後述のように、相似変換を加えられたモデルともとのモデルの対応をとる変換を決定するために用いる。したがって、透かし埋め込みの対象となるのはそれ以降（より高周波）の  $(n-5)$  個のスペクトル係数である。

これら使用可能なスペクトル係数をどのように使うかは、後述のように、ランダムノイズの重畳やスムージングなどに対する頑強性のうちのいずれを重視するかに依存する。たとえばランダムノイズに対する頑強性を上げるには拡散率  $c$  をできるだけ大きくする。頂点数が 1030 個の場合、相似変換の補正に使う係数を除くと、32 ビット長の透かしデータを埋め込むならば、最大の拡散率（透かしの繰返し回数）は 32 となる。逆に、形状のスムージングにある程度耐える透かしを実現するためには、低周波成分、つまり小さな固有値に対応するスペクトル係数を中心に透かしの埋め込みが必要がある。

### 2.3 透かしの取り出し

本論文の透かしは秘密透かしであり、透かしの取り出しに際し、透かしの埋め込まれたポリゴンメッシュ（埋め込み後にノイズ重畳などの妨害をされた可能性もある）と、透かし埋め込み前のオリジナルのメッシュの双方が必要である。

取り出し処理では、相似変換を加えられた透かしメッシュを被覆メッシュと幾何学的に重ね合わせる必要がある。これには、まず、各メッシュの概形を、その最初の 5 個（固有値の小さい方から 5 個）のスペクトル係数と固有ベクタを用いて再構成する。次いで、

各概形について、その頂点座標から 3 行 3 列の共分散行列を求める。この共分散行列を固有値分解して得られた固有ベクタがその形状の主軸である<sup>4)</sup>。概形の頂点の重心が重なるよう平行移動し、主軸の向きが揃うように回転し、大きさが合うようにスケールすると、2 つのメッシュが重なり合う。前述のように、透かしの埋め込みは 6 つ目以降のスペクトル係数を変更して行うので、透かしの存在はメッシュの重ね合わせには影響を与えない。

メッシュの重ね合わせが終わると、透かしの入ったメッシュから計算されたスペクトル係数  $\hat{r}_i = (\hat{r}_{s,i}, \hat{r}_{t,i}, \hat{r}_{u,i})$  とオリジナルのモデルから計算されたスペクトル係数  $r_i = (r_{s,i}, r_{t,i}, r_{u,i})$  の差分をとり、その差分に埋め込みに使用したのと同じ擬似乱数系列  $p_i$  を掛け、その結果を拡散率  $c$  個分総和する。この操作をスペクトルの  $s, t, u$  の 3 つの成分についてそれぞれ行い、その総和をとる。

$$\begin{aligned} q_j &= \frac{1}{3} \sum_{l \in \{s,t,u\}} \sum_{i=j \cdot c}^{(j+1) \cdot c - 1} (\hat{r}_{l,i} - r_{l,i}) \cdot p_i \\ &= \frac{1}{3} \sum_{l \in \{s,t,u\}} \sum_{i=j \cdot c}^{(j+1) \cdot c - 1} b'_i \cdot \alpha \cdot p_i^2. \end{aligned} \quad (8)$$

擬似乱数系列が同期しており、かつ、透かしメッシュに対して加えられた各種の妨害を無視できるとすると、

$$q_j = c \cdot \alpha \cdot b'_i \quad (9)$$

となる。ここで  $q_j$  は  $\{-\alpha c, \alpha c\}$  の 2 値いずれかをとり、 $c$  と振幅  $\alpha$  はつねに正の数である。したがって以下のように  $q_j$  の正負を判定することによって、埋め込まれた透かしデータビットベクタ  $\mathbf{a} = (a_1, a_2, \dots, a_m)$  の  $j$  番目の要素  $a_j$  を取り出せる。

$$a_j = \text{sign}(q_j). \quad (10)$$

実際には透かしの入ったメッシュに対し各種の変更が加わっている可能性がある。加えられる変更には、それが意図的な妨害であるか否かを別にして、乱数値重畳、メッシュのスムージング、ポリゴン単純化、リメッシング、相似変換やアフィン変換などの全域的幾何変換、メッシュの一部切り取り、あるいは局所変形などが考えられる。埋め込まれた透かしがどれだけの種類のどれだけの強度の妨害に耐えなければならないかは透かしの使用目的に依存する。たとえば著作権保護を目的とする場合にはできるだけ多くの種類の、できるだけ強度の高い妨害に耐えることが要求される（つまり、頑強な透かしが要求される）場合が多い。

次節で述べる領域分割に基づいた埋め込みの繰返しを行ったと仮定すると、本論文で述べる手法は、相似

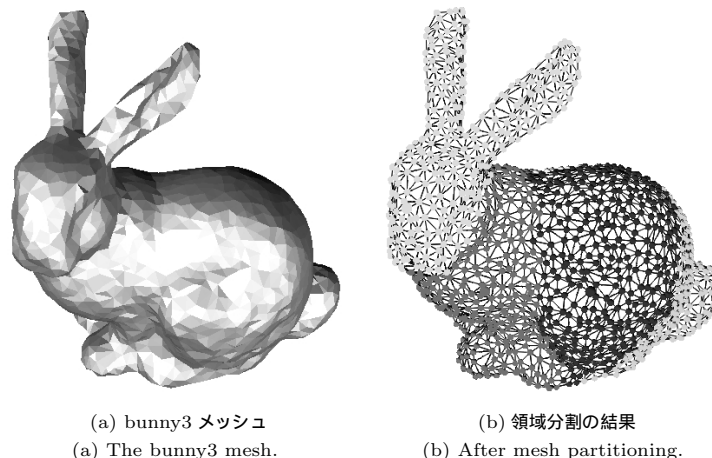


図 1 比較的大きな bunny3 メッシュ (頂点数 2218, 面数 4432) を 6 つの小領域に分割した例  
Fig.1 In this example, the bunny3 mesh (2218 vertices, 4432 faces) is partitioned into 6 sub-meshes.

変換に対して頑強で、さらに乱数値重畳、メッシュのスムージング、およびメッシュの一部切り取りに対してある程度の頑強性を持つ。頑強性については 3.3 節で実験結果とともに詳しく述べる。

#### 2.4 大規模メッシュ処理のための領域分割

我々の現在の実装では、ラプラシアン行列を Householder 法により相似変換した後で反復法を適用し、すべての固有値と固有ベクトルを求めている。メッシュの大きさ (頂点数) が数百程度まではこの手法が適用可能であるが、それを超えると、固有値と固有ベクトルを求めるのに莫大な計算時間を要する。そこで、我々は、大規模メッシュ (頂点数  $10^4 \sim 10^7$ ) に透かしを埋め込む場合、Karni ら<sup>7)</sup>が行ったように、メッシュを数百程度以下のポリゴンからなる領域に区分し、それぞれの領域に対して個別に固有値分解と透かしの埋め込みおよび取り出しの処理を施すことにする。

Karni らは、MeTiS<sup>9)</sup>と呼ばれるソフトウェアを用い、分割された領域がほぼ同じ数の頂点を含み、かつ領域の境目に存在する稜線がなるべく少なくなるように領域分割を行っている。しかし、MeTiS はメッシュ頂点の接続性のみに着目して領域分割を行うため、その分割は形状の凹凸などの幾何特徴を必ずしも反映しない。我々は、多少領域ごとの頂点数のばらつきや、領域境界に位置する稜線数の増加を許すが、その代わりに形状特徴を反映できるような領域分割を実装した。

我々の方法ではまず、メッシュの特徴領域となるであろう部分の中心にある頂点を、特徴点として、メッシュ全体になるべく等間隔に配置する。現在、これら特徴点の選択と配置は人手で行っている。次に、その

特徴点を出発点として、それぞれの特徴領域の勢力範囲を、1 つずつ隣接する頂点へと拡張していく。領域分割は、この領域の拡張操作で、メッシュ上のすべての頂点が覆い尽くされたときに完了する。

領域ごとの固有値計算と透かしの埋め込みおよび取り出しの処理は、それぞれの領域内の頂点群に関して定義されるラプラシアン行列を用いて行われ、領域間にまたがる稜線は無視される。いうまでもないが、透かしの埋め込みと取り出しで同一の領域分割が必要である。したがって、取り出し処理のために、領域分割のもととなる特徴点を透かし埋め込み前のメッシュと一緒に保存しておく。

領域分割を用いると、大規模メッシュの処理を可能にするだけでなく、メッシュの部分的切り取りに対する耐性を高めることもできる。これには、分割されたそれぞれの複数の領域に同一の透かしの埋め込んでおく。メッシュの切り取りが加えられても、切り取りの加わった領域と隣接せず、したがって切り取りの影響のない領域を使えば、固有ベクトルを用いたメッシュの重ね合わせおよび透かし前後のメッシュの比較による透かしの取り出しが可能である。

図 1 は、比較的大きなウサギのポリゴンメッシュモデル bunny3 (頂点数 2218, 面数 4432) (図 1(a)) を 6 つの小領域に分割した結果 (図 1(b)) を示している。この例では小領域の大きさは頂点数で 334 から 494 の範囲となった。

### 3. 実験と結果

我々は 2 章で述べたアルゴリズムを C++ と GUI

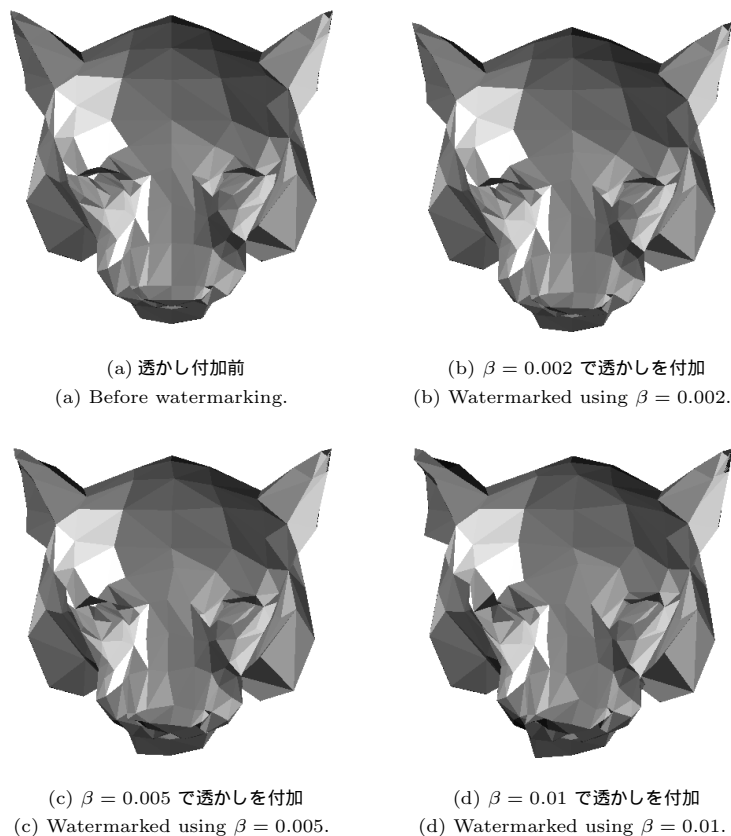


図 2 透かし埋め込みの振幅を変え、透かし付加の前後で形状の見かけを比較

Fig. 2 Appearance of the tiger model is compared before and after the watermarking using multiple embedding amplitudes.

ツールキット ftk (<http://www.ftk.org>) を用いて実装し、実験を行った。

### 3.1 透かし埋め込みの例

図 2 は虎のメッシュモデル tiger (図 2 (a)) の VRML モデル、頂点数 254、面数 504) に領域分割をせずに透かしを埋め込んだ例である。埋め込みの振幅率  $\beta$  は、座標軸に沿ったバウンディングボックスの最大長の 0.2% ( $\beta = 0.002$ , 図 2 (b)), 0.5% ( $\beta = 0.005$ , 図 2 (c)), 1% ( $\beta = 0.01$ , 図 2 (d)) の 3 種類を使用し、また拡散率  $c = 7$  を用いた。透かしを埋め込む処理には Pentium III 866 MHz の PC で約 20 秒かかり、そのほとんどが固有値分解の時間であった。取り出し時には参照形状の固有値分解を行う必要があるため、ほぼ 2 倍の時間がかかった。

埋め込み前 (図 2 (a)) と、 $\beta = 0.002$  で埋め込んだ後 (図 2 (b)) ではほとんど見かけに変化はない。より大きな振幅の  $\beta = 0.005$ ,  $\beta = 0.01$  で埋め込んだ場合では、特に振幅の最も大きい  $\beta = 0.01$  の場合において透かしによる形状品質の劣化が明らかに認められた。

表 1 透かし埋め込み処理の時間

Table 1 Execution time for our mesh watermarking algorithm.

Models	Number of Partitions	Number of vertices	Number of faces	Exec. Time
tiger	1	254	504	20 s
bunny1	1	646	1288	5m21 s
distcap	1	686	1368	6m19 s
bunny2	1	1197	2390	33m16 s
tiger	3	254	504	2 s
bunny3	6	2218	4432	6m34 s

実際に透かしによる形状品質の劣化がどれほどの振幅で認知されるかは、透かしの対象とするメッシュに依存する〔透かしによる形状品質劣化の認知は、このほかに、レンダリング手法 (ワイヤフレームかスムーズシェーディングかなど)、カメラパラメータ、照明の位置、面の色などにも影響を受ける〕。

### 3.2 領域分割による大規模メッシュの扱い

表 1 は透かしの埋め込みに要する時間をモデルの大きさ (頂点数) と領域分割の数で比較したもので

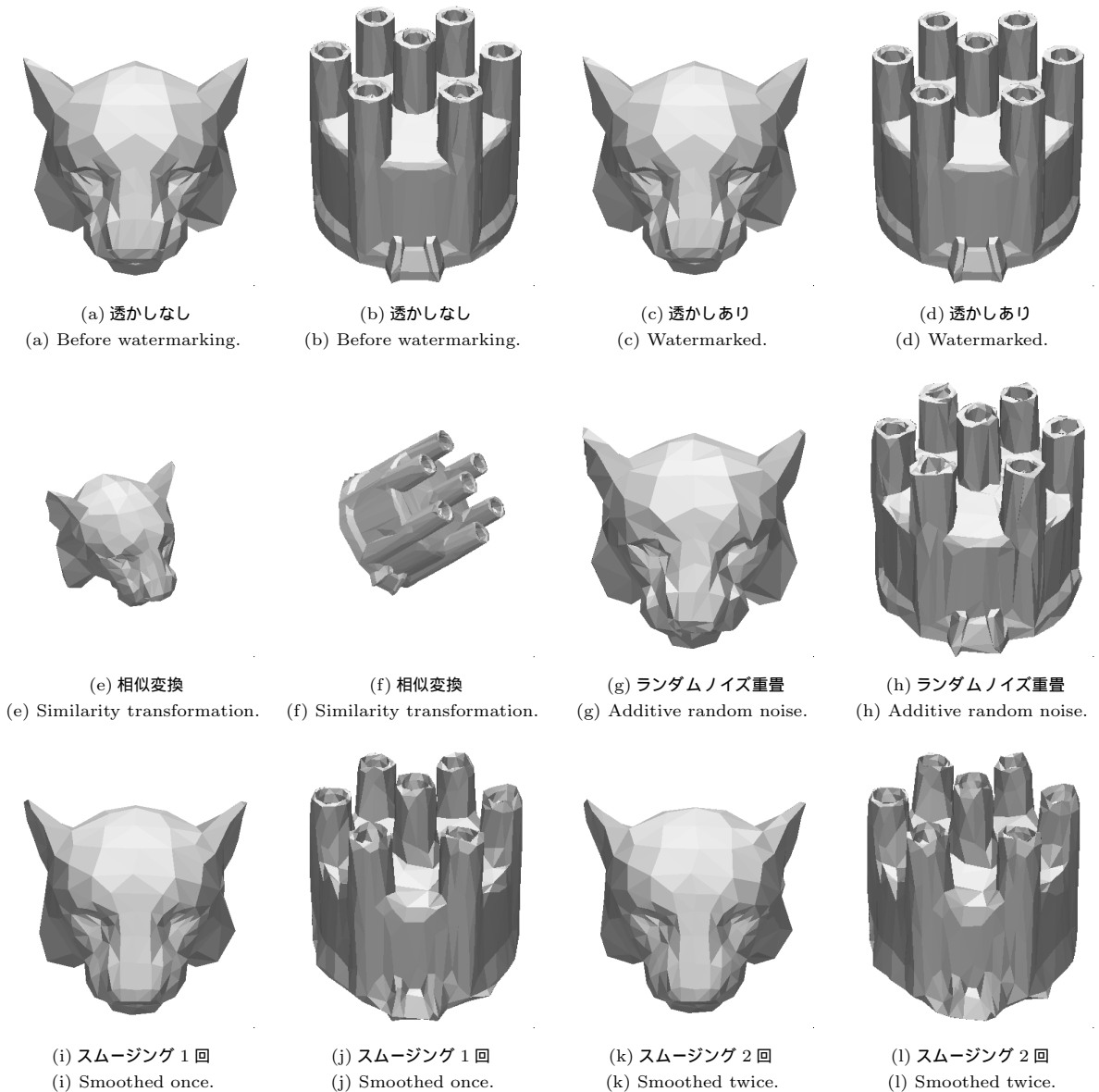


図 3 領域分割を施さない場合の透かし埋め込みの例

Fig. 3 Examples of watermarking without mesh partitioning.

ある．領域分割なしの場合は 686 頂点のディストリピュータキャップのメッシュ (distcap) で処理時間が 6 分を超えた．さらに, 1197 頂点のウサギのメッシュ (bunny2) では処理時間が 30 分を超えており, 実用的でない．また, 1000 頂点を超えるようなメッシュの場合, 固有値計算の数値的安定性の面からも問題がある．

しかし, 領域分割を施し, 分割後の領域の大きさを, たとえば 500 頂点以下におさえられるようにすれば, より大きなメッシュに対する透かし埋め込みが可能で

ある．表から分かるように, 2000 頂点を超えるウサギのメッシュ (bunny3) に対する埋め込みも 6 分強で処理できた．

領域分割した場合の処理時間は分割した小領域の大きさに大きく依存する．分割数を増やして小領域を小さくとると全体の処理時間は減る．また, メッシュの部分的切り取りに対する耐性は高くなる．反面, 埋め込み情報量が減り, また後述するノイズやスムージングに対する頑強性も低下する．

### 3.3 頑強性

ポリゴンメッシュに加えらるる各種の変更に対する頑強性について実験を行った。

#### 3.3.1 相似変換

本手法で埋め込まれた透かしは相似変換に対して頑強性を持つ。図3(a)は虎のモデル tiger (頂点数 254, 面数 504) を, 図3(b)はディストリビュータキャップのモデル distcap (頂点数 686, 面数 1368) を示す。これらに透かしを埋め込んだものがそれぞれ図3(c)と図3(d)である。ここで用いたのは変調振幅率  $\beta = 0.005$ , 拡散率  $c = 7$  であった。透かし入りのモデルを  $x$  軸回りに 10 度,  $z$  軸回りに 30 度, 回転し, さらに 0.6 倍の等方スケールリングを施した結果が図3(e)および図3(f)であるが, この変換後もビット誤りなしで透かしを取り出すことができた。

ただし, 形状によっては相似変換後の取り出しがうまくいかないこともある。たとえば球や円筒のように点対称あるいは軸対称に近い形状では, 形状の固有値を使用したモデルの重ね合わせがうまくいかない。特に, 相似変換と形状のスムージングを組み合わせると, 推定した相似変換の誤差が大きくなり, 透かしが取り出せない場合がある。

#### 3.3.2 頂点へのランダムノイズ重畳

埋め込みの拡散率  $c$  が高いと頂点座標に重畳されるランダムノイズに対する耐性が高まる。

Bbox 最大長の 0.5% の振幅率 ( $\beta = 0.005$ ) で透かしを埋め込んだ虎のモデルの頂点座標値に Bbox 最大長の 0.7% の振幅率 ( $\beta = 0.007$ ) でノイズを重畳する (図6(g)) と形の劣化が認められる (目の付近などに注目)。この場合, 拡散率  $c = 1$  ではデータが破壊されたが, 拡散率を  $c = 7$  (可能な最大値) まで高めると透かしが残った。

図4は図1(a)の bunny3 のメッシュに図1(b)の領域分割を施し, これに透かしを埋め込んだ場合において, 重畳するノイズの振幅を固定し, 拡散率とビットエラー率の関係をプロットしたものである。

ここで, 以後本論文で用いるビットエラー率とは, 取り出しの結果, 誤っていたビットの数を埋め込んだ総ビット数で割った値である。たとえば, まったく誤りがなければ 0 となる。ビットエラー率を測定する実験は 3 種類の埋め込みビットパターンについてそれぞれ 10 回ずつ試行し, その平均をとった。実験に用いた 3 種類のビットパターンはすべて 0, すべて 1, およびランダムパターンの 3 種類であった。ここで, 頂点座標値に重畳する乱数値は試行ごとに変えたが, 透かしに埋め込んだランダムなビットパターンは試行を

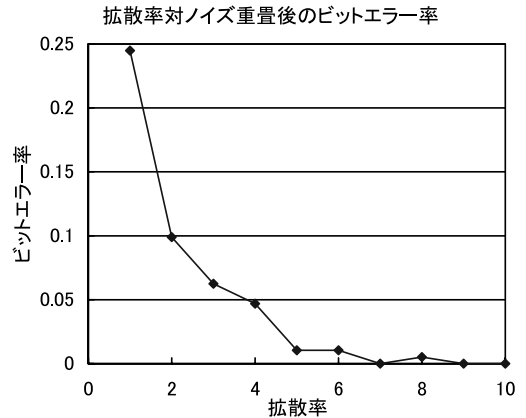


図4 頂点座標に一定振幅のノイズを重畳した場合の, 拡散率に対するビットエラー率の変化

Fig. 4 The bit-error rate is plotted against the chip rate after the vertex coordinate is added with the uniform random noise.

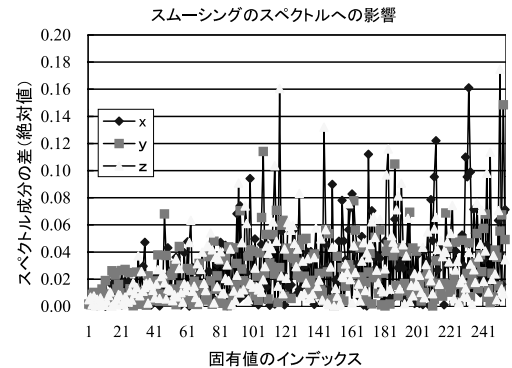


図5 スムージングがスペクトルに与える影響。tiger モデル, 領域分割なしで実験

Fig. 5 Changes in the spectral coefficient amplitude due to mesh smoothing. This experiment used the tiger model (254 vertices) without mesh partitioning.

繰り返す間は一定に保った。

図4で分かるように, 拡散率の増加につれてビットエラー率が下がり, 拡散率  $c = 9$  以上ではビットエラー率は 0 であった。拡散率をさらに高めるにはメッシュ自体の頂点数が大きい必要がある。

#### 3.3.3 メッシュスムージング

メッシュの fairing や透かしの破壊の目的で頂点座標に対しスムージングを加える操作が加えられることがある。ローパスフィルタであるスムージングに対抗するには, スペクトル成分の中で低周波成分だけを用いて透かしを埋め込めばよいことが想像される。図5は, tiger モデルについて Taubin らの手法<sup>16)</sup>によるメッシュのスムージングの前後でのメッシュスペクトル係数の変化の絶対値を固有値のインデックスに対し



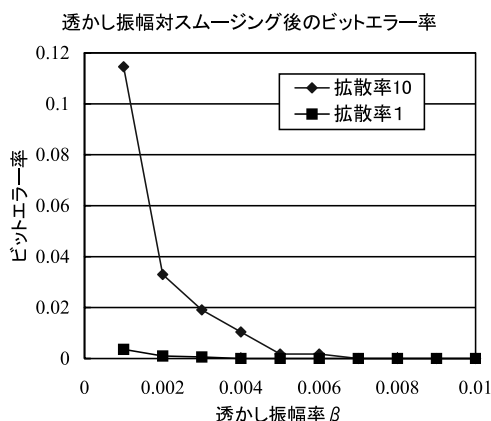


図 6 Taubin のスムージング 1 回を施した後の埋め込み振幅率  $\beta$  とビットエラー率

Fig. 6 The bit-error rate after an application of Taubin's mesh smoothing, plotted against watermark embedding amplitude ratio  $\beta$ .

てプロットしたものである。横軸は全 254 個の固有値を小さい順に並べたインデックスで、左(小さいインデックス)が低周波、右(大きいインデックス)が高周波に相当する。このグラフから、予想どおり、スムージングが最も影響を与えるのはスペクトルの高周波成分であることが分かった。

tiger と distcap の 2 つのモデルに対し、スペクトルの低周波成分を使って 32 ビットのデータを拡散率  $c = 1$  で埋め込み、このモデルに対し Taubin のスムージングを施して透かしが残るかどうかが実験した。透かしの埋め込んだデータ(図 3(c), 図 3(d))を 1 度スムージングしても(図 3(i), 図 3(j))透かしが取り出せるが、2 度スムージングすると(図 3(k), 図 3(l))透かしは破壊された。スムージング前の図 3(c) および図 3(d) とスムージング後の図 3(i) および図 3(j) では形状の品質が視認可能な程度低下していることに注意してほしい。

図 6 は図 1(a) の bunny3 のメッシュに図 1(b) の領域分割を施し、これに透かしの埋め込んだ場合において、Taubin のスムージング 1 回を加えた後のビットエラー率を、拡散率 1 および 10 の 2 つの場合について、透かしの振幅を変えて測定した結果をプロットしたものである。透かしの振幅を増やしていくとビットエラー率が低減し、0 になった。また、スペクトルの低周波成分しか使わない拡散率  $c = 1$  の場合の方が、高周波成分まで使う拡散率  $c = 10$  の場合よりもスムージングに対するエラー率は低かった。これは、スムージングのローパスフィルタとしての振舞いから予想されたとおりの結果である。

このほか、透かしの振幅と拡散率によっては 2 回以上のスムージングに耐える場合もあった。ただし、図 4 のグラフからも分かるように拡散率を下げすぎるとノイズ重畳に弱くなる。

### 3.3.4 メッシュの部分切り取りに対する耐性

透かしの埋め込みの際に、メッシュに領域分割を施しておけば、メッシュ形状を部分的に削除されても、残った領域から透かしの抽出ができる。

図 7(a) は、ウサギのモデル bunny3 (頂点数 2218, 面数 4432) に対し図 1(b) に示した領域分割を施し、分割後の小領域ごとに拡散率  $c = 10$ ,  $\beta = 0.002$  の振幅で透かしの埋め込んだメッシュを示している。このメッシュからは、分割した領域それぞれで完全な透かしを抽出することができる。ここで、図 7(b) のように、ウサギの耳の部分を取り切った場合、透かしが正しく抽出できるかどうかを実験した。この場合、耳の領域に隣接していない尻尾の領域を使って透かしを取り出すことができた。また、図 7(b) のメッシュに相似変換を施した場合も、先の尻尾の領域を使って相似変換を補正し、透かしを取り出すことができた。

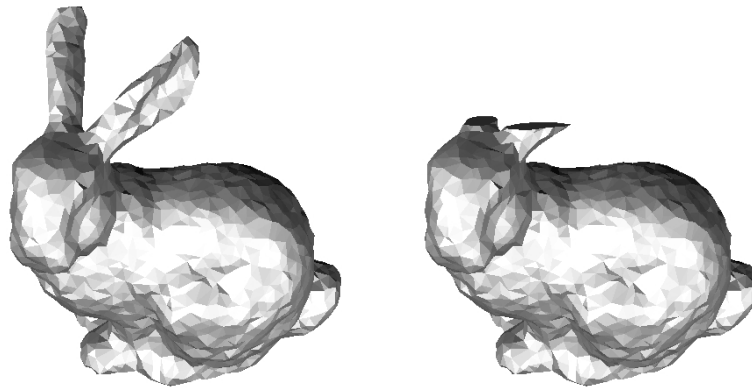
### 3.3.5 その他の変更に対する耐性

本手法はメッシュの単純化、再パラメータ化、エッジフリッピングなど、頂点接続関係を変更する操作に対しては頑強性が低い。頂点接続関係が変わると、メッシュスペクトル計算のもととなるメッシュラプラシアン行列が大きく変わってしまうためである。これら接続性の変更に対する頑強性向上は今後の課題である。

## 4. 関連研究

3 次元形状を対象とした電子透かしの研究に端緒をつけたのが Ohbuchi らである<sup>11)~13)</sup>。Ohbuchi らは頂点座標値を変更する透かしと頂点接続性を変更する透かしのそれぞれ複数発表した。前者の透かしは、想定されたクラスの幾何変換には耐えるが、それ以外の頂点座標値の変更、たとえば乱数値の重畳や形状のスムージングなどで破壊される。また、頂点接続性を変更するような変更でも破壊される。後者の、頂点接続性を変更する透かしは、任意の座標変換に耐えるが、頂点あたりの埋め込み情報量(以後、これを「透かし情報密度」と呼ぶ)が相対的に低い。

Ohbuchi らの仕事の後、より妨害耐性の高い頑強な透かしの手法が発表された。Benedens の手法<sup>3)</sup>は頑強な秘密透かしの手法で、その透かしは相似変換に耐え、また頂点座標値へのノイズの重畳や頂点接続性の変更にも耐える。しかし、透かし情報密度が低い。Praun らの透かし<sup>15)</sup>も頑強な秘密透かしで、相似変



(a) 領域分割により透かしを入れたメッシュ  
(a) The mesh was partitioned and then watermarked.  
(b) 耳を切り取ったメッシュ  
(b) Ears of the bunny mesh are resected, but the watermark remained.

図7 領域分割を用いて透かしを埋め込んだメッシュと、その一部を切り取ったメッシュ

Fig. 7 Resecting a part of the partitioned and watermarked mesh left the watermark intact.

換、切り取り、ノイズの重畳、ポリゴン単純化、形状スムージングなどに対して耐性を持つ。この手法の欠点は、Benedensの手法同様に透かし情報密度が低いことである。たとえばポリゴン単純化に対する耐性の実験も、数万頂点からなる大きなメッシュを対象として行っている。Wagnerの透かし<sup>17)</sup>は頑強な公開透かしで、その透かしはアフィン変換や頂点座標値への乱数値の重畳などに耐える。しかし、頂点接続性を変更する操作を加えると透かしは破壊される。長所として、Wagnerの手法はBenedensやPraunの手法よりも頂点数あたりに埋め込める情報量が多く、また公開透かしである。

2次元画像や音声に対する電子透かしで用いられる手法に、対象データにFourier(フーリエ)変換やWavelet(ウェーブレット)変換などの変換を施し、その変換領域の係数を操作して透かしを埋め込むものがある<sup>8)</sup>。このアプローチの利点は、変換領域の1つの変更がもとの領域では広範囲に拡散されるため、透かし可知性が低くなる利点がある。また、周波数領域において特定の周波数帯の係数を対象として選ぶと、たとえばローパスフィルタ処理に耐性を持つ透かしを実現できる。

2次元画像などと異なり、3次元ポリゴンメッシュは不規則サンプルデータである。そのため、フーリエ変換のような扱いやすい変換が存在しない。このためか、上述の3次元ポリゴンメッシュを対象とした手法はすべて変換を使わずに透かしを埋め込む手法である(ただし、Praunの手法は多少変換領域を意識している)。ポリゴンメッシュを対象とした、数学的後ろ盾を持

つ変換として、ポリゴンメッシュの上のウェーブレット変換がある<sup>10)</sup>。これをいち早くポリゴンメッシュの電子透かしに応用したのがKanaiらである<sup>6)</sup>。Kanaiらは、1対4の細分割連続性を仮定した3角メッシュをウェーブレット変換し、その変換領域のウェーブレット係数を操作して透かしを埋め込んだ。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頑強性を有した。ただ、欠点として、埋め込み対象が1対4の細分割連続性を持つ3角メッシュに限定された。

本論文で述べた電子透かしは、Kanaiらと同様、何らかの変換領域で透かしを埋め込む。さらに、Kanaiらの手法より広い範囲のポリゴンメッシュを対象とする。つまり、本手法は、透かし埋め込みの対象を1対4の細分割連続性を持つ3角メッシュに限定しない。本論文の透かしは、Kanaiらの手法と似て、頂点座標値への乱数値重畳に耐性を示し、また形状のスムージングにも耐性を示す。ただ、現在のところ、本論文の透かしが耐えることのできる幾何変換は相似変換に限定されている。Praunの手法やBenedensの手法と比較すると、頂点あたりの情報量は我々の手法の方が高い。たとえば、頂点数300程度の小さなメッシュにも数十ビット以上の実用的な量の透かしを埋め込むことができる。

本論文で述べたような頑強な透かしにとってその耐性の評価は重要である。しかし、現実にある透かしの耐性を他の手法のそれと比較するのには困難がともなう。たとえば、今のところ、ベンチマークとなる標準的なデータや標準的な妨害手法は存在しない。我々も、

本論文の手法と他の手法との頑強性の直接比較は行うことができなかった。

3Dモデルにおける透かしの可知性の評価も、2次元画像などの透かしのそれとは異なる困難がともなう。たとえば、3Dモデルでは、透かしの可知性を判断するのはディスプレイに表示されたモデルを眺めるヒトではなく、モデルを編集しようとする3D CADシステムかもしれない。また、ディスプレイでモデルを表示する場合にも、レンダリング手法(ワイヤフレームか、スムーズシェーディングかなど)、光源の種類と配置、カメラパラメータなど、数々の変数が存在する。

3Dモデルの透かしの研究を進めるには、今後、頑強性や可知性を評価する手法や標準の研究も重要であろう。

## 5. まとめと今後の課題

本論文では、ポリゴンメッシュで定義された3次元形状をスペクトル分解し、そのスペクトル(周波数成分)を変更して透かしを埋め込む手法を提案した。メッシュの頂点の接続関係のみから得られるメッシュラシアン行列を求め、それを固有値分解して得られた固有ベクトルに頂点座標値を射影することでメッシュのスペクトルが得られる。このスペクトル係数を白色ノイズ化した透かしビット列で変更して透かしを埋め込んだ。

本論文で提案した手法を分類すると、妨害に耐えて残ることを目標とする頑強透かしで、かつ、取り出しにもとのモデルと透かし入りのモデルの双方が必要な秘密透かしである。本手法により埋め込まれた透かしは、メッシュに加えられる相似変換(回転、並行移動、一様スケール)に対して頑強で、頂点座標へのランダムノイズ重畳や、形状のローパスフィルタリングに相当するメッシュのスムージングに対してもある程度の耐性を持つ。さらに、メッシュを小領域に分割し、繰り返して同じ透かしの埋め込みによりメッシュの部分的切り取りに対する耐性も持つ。

本論文で提案した手法にはまだ不十分な点も多い。まず、メッシュの接続性を変更する妨害、たとえばメッシュ単純化や再メッシュ化に対する耐性がない。こういったメッシュ接続性の変更に対する透かしの頑強性向上が大きな課題である。これについては、もとの(透かし埋め込み前の)メッシュを用いて透かし入りのメッシュの形状をリサンプリングすることで解決する予定である。また、メッシュの形状特徴を考慮し、小領域の大きさのばらつきを抑えつつ、自動的にかつ効率よくメッシュの領域分割を行う手法を考案したい。

謝辞 本研究は文部省科学研究費補助金(課題番号12680342 および 12780185)、および中山隼雄科学技術文化財団、株式会社ミリオン、および柏森情報科学振興財団からの助成による。

## 参考文献

- 1) Biggs, N.: *Algebraic Graph Theory*, 2nd Ed., Cambridge University Press (1993).
- 2) Bollobás, B.: *Modern Graph Theory*, Springer (1998).
- 3) Benedens, O.: Geometry-based Watermarking of 3D Models, *IEEE CG&A*, pp.46-55 (Jan./Feb. 1999).
- 4) Gottschalk, S., Lin, M.C. and Manocha, D.: OBBTree: A Hierarchical Structure for Rapid Interference Detection, *Proc. SIGGRAPH '96*, pp.171-180 (1996).
- 5) 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).
- 6) Kanai, S., Date, H. and Kishinami, T.: Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition, *Proc. 6th IFIP WG 5.2 GEO-6*, Tokyo, Japan, pp.296-307 (Dec. 1998).
- 7) Karni, Z. and Gotsman, C.: Spectral Compression of Mesh Geometry, *Proc. SIGGRAPH 2000*, New Orleans, U.S.A. (July 2000).
- 8) Katzenbeisser, S. and Petitcolas, F.A.P.: *Digital Watermarking*, Artech House, London (2000).
- 9) Karypis, G. and Kumar, V.: MeTis: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-reducing Orderings of Sparse Matrices, Version 4.0, Univ. of Minnesota, Dept. of Computer Science (1998). Available at the following address: <http://wwwusers.cs.umn.edu/~karypis/memis/memis.html>
- 10) Lounsbery, M., DeRose, T.D. and Warren, J.: Multiresolution Analysis for Surfaces of Arbitrary Topological Type, *ACM Trans. Graphics*, Vol.16, No.1 (1997).
- 11) Ohbuchi, R., Masuda, H. and Aono, M.: Watermarking Three-Dimensional Polygonal Models, *Proc. ACM Multimedia '97*, Seattle, Washington, USA, pp.261-272 (Nov. 1997).
- 12) Ohbuchi, R., Masuda, H. and Aono, M.: Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, *IEEE JSAC*, pp.551-560 (May 1998).

- 13) Ohbuchi, R., Masuda, H. and Aono, M.: Geometrical and Non-geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models, *Computer Communications*, Vol.21, pp.1344-1354, Elsevier (1998).
- 14) Ohbuchi, R., Masuda, H. and Aono, M.: A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces, *Proc. Computer Graphics International '99*, Canmore, Canada, pp.180-177 (June 1999).
- 15) Praun, E., Hoppe, H. and Finkelstein, A.: Robust Mesh Watermarking, *Proc. SIGGRAPH '99*, pp.49-56 (1999).
- 16) Taubin, G.: A Signal Processing Approach to Fair Surface Design, *Proc. SIGGRAPH '95*, pp.351-358 (1995).
- 17) Wagner, M.G.: Robust Watermarking of Polygonal Meshes, *Proc. Geometric Modeling & Processing 2000*, Hong Kong, pp.201-208 (Apr. 2000).
- 18) Yeo, B-L. and Yeung, M.M.: Watermarking 3D Objects for Verification, *IEEE CG&A*, pp.36-45 (Jan./Feb. 1999).
- 19) 松井甲子雄：電子透かしの基礎，森北出版 (1998).
- 20) 大淵竜太郎：3D デジタルコンテンツのためのモデリング技術—インターネット流通を目指した圧縮と透かしの技術，*情報処理*，Vol.41, No.10, pp.1113-1118 (2000).

(平成 12 年 10 月 4 日受付)

(平成 13 年 3 月 9 日採録)



大淵竜太郎 (正会員)

1981 年上智大学理工学部電気電子工学科卒業。1983 年電気通信大学大学院計算機科学科修士課程修了。同年日本アイ・ピー・エム (株) 入社，東京基礎研究所 (現在) に勤務。

以後留学し，1994 年 University of North Carolina at Chapel Hill Computer Science Department より Ph.D. 取得。同年より日本アイ・ピー・エム (株) 東京基礎研究所勤務。1999 年より山梨大学工学部コンピュータ・メディア工学科助教授。マルチメディアデータ型としての 3 次元モデルの処理，たとえば圧縮，電子透かし，検索等が興味を中心である。その他，拡張現実表示とその応用にも興味を持つ。ACM, IEEE, 日本ソフトウェア科学会各会員。



高橋 成雄 (正会員)

1968 年生。1997 東京大学大学院理学系研究科情報科学専攻博士課程修了。博士 (理学)。同年群馬大学工学部情報工学科助手。同大学総合情報処理センター助教授を経て，2001 年より東京大学大学院総合文化研究科広域システム科学系助教授。コンピュータ・グラフィックス，幾何形状モデリング，地理情報システム等に興味を持つ。ACM, IEEE, 電子情報通信学会各会員。



宮澤 貴彦

1977 年生。2001 年山梨大学工学部電子情報工学科卒業。工学士。同年より富士通長野システムエンジニアリング (株) 勤務。3 次元モデルの処理，特に形状 CAD データの著作権保護，圧縮等の技術に興味を持つ。



向山 明夫

1978 年生。2001 年山梨大学工学部電子情報工学科卒業。工学士。同年より同大学工学研究科博士前期課程入学。3 次元モデルの処理，特に電子透かしに関する研究に従事。