スペクトル分解を用いた3次元メッシュ電子透かしの耐性強化法

宮澤　貴彦[1], 向山　明夫[1], 高橋　成雄[2] , 大渕 竜太郎[1]

k7184@kki.yamanashi.ac.jp, k7186@kki.yamanashi.ac.jp, takahashis@acm.org, ohbuchi@acm.org

[1] 山梨大学工学部 コンピュータ・メディア工学科, 山梨県甲府市武田 4-3-11

[2] 群馬大学総合情報処理センター, 群馬県桐生市天神町　1-5-1

要旨

電子透かしは, 各種のデータを対象とし, それに透かしと呼ばれる構造を付加して情報を埋め込む. 我々が以前発表した 3 次元ポリゴンメッシュを対象とする手法では, メッシュのスペクトル係数を変更して透かしを付加する. この透かしはメッシュの相似変換, 頂点座標への乱数値重畳, 形状スムージングに対し耐性を持つ. しかしこの手法は, メッシュの頂点数増加に伴い計算時間が急増し, またメッシュに頂点接続性を変更する操作を加えると透かしが壊れる. 本論文では, これら 2 つの短所を改善するため, モデルの領域分割により透かし処理を高速化する手法, 形状リサンプリングにより頂点接続性を変更する操作に対する耐性を増す手法の 2 つを提案する.

Improving Efficiency and Robustness of a Mesh Watermarking Algorithm Based On Mesh Spectral Analysis

Takahiko Miyazawa[1], Akio Mukaiyama[1], Shigeo Takahashi[2], and Ryutarou Ohbuchi[1]

k7184@kki.yamanashi.ac.jp, k7186@kki.yamanashi.ac.jp, takahashis@acm.org, ohbuchi@acm.org

[1]Computer Science Department, Yamanashi University, 4-3-11 Kofu, Yamanashi, Japan.

[2]Computer Center, Gunma University, 1-5-1 Tenjin-cho, Kiryu-shi, Gunma, Japan

Abstract

Digital watermarking embeds structures called watermark into target data object with such applications are copyright protection and annotation. Previously, we published an algorithm that embeds watermarks in the shape of a 3D polygonal mesh in its mesh spectral domain. The watermark is resistant to similarity transformation, additive random noise, and mesh smoothing. However, the algorithm suffered from a long processing time for a large mesh and the watermark was lost if the mesh's vertex connectivity is altered. In this paper, we propose remedies to these shortcomings. Computational cost is reduced by partitioning the mesh into approximately equal-sized sub-meshes. Connectivity alteration is countered by mesh geometry resampling by using the connectivity of the original mesh.

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, 埋め込む情報を表現した watermark (透かし) という構造体を対象となるデータに付加する. ここで透かしには, 対象コンテンツの本来の目的 (表示と鑑賞, 等) を阻害しないこと, またそのコンテンツからの除去が困難なこと, が要求される. 透かしの用途は, 説明の追加, 著作権の保護, などである. 電子透かし一般については [松井 98, Katzenbeisser00]等を参照してほしい.

電子透かしの対象としては静止画像や音声などが中心だが, 近年これに VRML や MPEG4, さらには形状 CAD データなどの 3D モデルが加わった. 3D モデルを対象とする既存の手法の多くはポリゴンメッシュの形状を対象とする [Ohbuchi97, Kanai98, Ohbuchi98a, Ohbuchi98b, Benedens99, Praun99, Yeo99, Wagner00, 大渕 00] .

この中で我々が昨年提案した手法[大渕 00]は, ポリゴンメッシュで定義される 3 次元形状をスペクトル分解し, 得られたスペクトル係数を変更することで, メッシュに透かしを付加する. この手法で得られた電子透かしはメッシュに加えられる相似変換, 頂点座標へのランダムノイズ重畳, 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[Taubin95]に対しても耐性を持った. しかしこの手法は, 頂点数の多いメッシュを対象とした場合に計算時間がかかり実用的ではなく, また, 頂点の接続性を変更されると(例えばポリゴン簡単化

などの操作を加えられると)透かしが壊れた.

本論文ではこれら 2 つの弱点を克服する手法を提案する. まず, 計算時間を減らすためには, 領域分割を用いてメッシュスペクトル計算の対象となるメッシュの大きさを一定以下に抑える. また, 接続性の改変を修復するためには, 接続性の変わったモデルの形状を, 透かしを埋め込む前のモデルの接続性を用いてリサンプリングする.

以下, 第 2 節では元となる, メッシュのスペクトル分解を用いた電子透かしアルゴリズムを述べ, 第 3 節で領域分割によるスペクトル分解の効率化法を, 第 4 節でリサンプリングを用いた妨害耐性強化法を述べる. さらに, 第 5 節で実装と実験結果を, 第 6 節でまとめと今後の課題を述べる.

2. スペクトル分解を用いた電子透かし

本論文の電子透かし手法は, 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頂点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元にした, ポリゴンメッシュ形状のスペクトル分解である[Karni00].

スペクトル分解の結果は, 小さな固有値 (低周波) とその固有ベクトルがメッシュの概形を, 大きな固有値 (高周波) 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. 我々の手法[大渕 00]では, その分解によって得られる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の入ったモデル (透かしモデル) と透かし付加前のモデル (被覆モデル) が必要である. 透かしの取り出しは, 透かしモデルと被覆モデルをスペクトル分解し, 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, 頂点の接続関係で定義されたメッシュラプラシアン行列に対し固有値分解を施し, 得られた固有ベクトルに頂点座標を射影して得られた係数からなる.

メッシュラプラシアン行列にはいくつかの定義があるが, 我々は[Bollobás98]のメッシュラプラシアン行列 (別名 Kirchhoff (キルヒホフ) 行列) を用いた. この行列は, 対称行列になるため, 比較的安定で効率的な固有値分解のアルゴリズムが存在するという利点がある. キルヒホフ行列 $\mathbf{K}$ は

$$\mathbf{K} = \mathbf{D} - \mathbf{A}$$

で定義される. $\mathbf{A}$ 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.

$$A_{ij} = \begin{cases} 1 & \text{頂点}i\text{と}j\text{が隣接} \\ 0 & \text{その他} \end{cases}$$

$\mathbf{D}$ 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.

2.2. 透かしの埋め込み

本論文の透かし手法は, 前項で述べたメッシュスペクトルの係数値を変更して透かしを埋め込む.

埋め込むデータビット列の j 番目のビット a_j (値は $\{0,1\}$ を取る) はランダムノイズに対する頑強性向上のため, c 倍に拡散し, 埋め込みシンボル b_i となる.

$$b_i = a_j, \quad j \cdot c \le i < (j+1) \cdot c \tag{1}$$

b_i は{0, 1}から, {-1, 1}の値をとる透かしシンボル b'_i に変換される. これらの透かし情報を埋め込むことによって変更されたスペクトル係数 $\hat{s}_j$ は次のように計算される.

$$\hat{s}_i = s_i + b'_i \cdot p_i \cdot a \tag{2}$$

s_i : 変更する前のスペクトル係数
p_i : 透かし埋め込み鍵 k_w (整数値) から生成された $\{-1, 1\}$ の値をとる擬似乱数列
a : 変調振幅

変調振幅 α は被覆モデルを囲う最小の直方体である axis-aligned Bounding Box を計算し, その辺の長さの最大値に対する割合で指定する.

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, 透かしの取り出しの際, 透かしモデルと被覆モデルとを比較して透かしを取り出す.

取り出し処理は, 次の手順で行われる (詳細は[大渕 00]を参照). まず透かしモデルと被覆モデルから得られる固有ベクトルを使って位置合わせを行う. 位置合わせが終わると, 式(1)(2)で使用した c, s_i, p_i, a を使用して, q_j を求める.

$$q_j = \sum_{i=j\cdot c}^{(j+1)\cdot c-1} (\overline{s_i} - s_i)\cdot p_i = \sum_{i=j\cdot c}^{(j+1)\cdot c-1} b_i' \cdot \boldsymbol{a} \cdot p_i^2 \qquad (3)$$

$\overline{s_i}$: **透かしを埋め込んだモデルから計算されたスペクトル係数**

ランダムノイズなどの妨害を無視できるとすると，q_j は，式（4）のようになる．

$$q_j = c \cdot \boldsymbol{a} \cdot b_i' \qquad (4)$$

式（4）より，c と振幅 $\boldsymbol{a}$ は常に正の数であるから，q_j の正負を判定することによって透かしデータ a_j が取り出せる．

$$a_j = sign(q_j) \qquad (5)$$

3. 大規模メッシュ処理のための領域分割

モデルの頂点数が増えると，スペクトル分解のために行う固有値の計算に時間が急速に増大し，実用的でない（表 1）．そこで，頂点数が数千個以上のメッシュに透かしを埋め込む場合は，Karni [Karni00]がメッシュの圧縮について行ったように，メッシュを頂点数数百程度以下のメッシュからなる領域に区分し，それぞれの領域に対して個別にスペクトル分解を行い，透かしの埋め込み及び取り出しの処理を施すことにした．Karni らは，MeTiS[Karypis98]と呼ばれるソフトウェアを用いて分割を行ったが，我々は後述の手法を用いた．

領域ごとの透かし処理は，それぞれの領域内の頂点群に関して定義されるメッシュラプラシアン行列を用いて行う．当然，透かしの埋め込みと取り出しにおいては，同一の領域分割が必要である．

同一の透かしを各領域に繰り返し埋め込むことにより，メッシュの切り取りに対する耐性も高めることができる．これは，領域ごとに同じ内容の透かし情報を埋め込むので，透かしを埋め込んだ領域がどれか一つでも切り取られずに残っている場合，透かしが取り出せるためである．

表 1　固有値計算の時間．

モデル	頂点数	面数	時間
tiger	254	504	20s
distcap	686	1368	6m19s
bunny1	1197	2390	33m16s
bunny2	2218	4432	2h54m40s

3.1. 分割のアルゴリズム

まず，メッシュの頂点の中から，領域の中心となる頂点を何点か選ぶ（これを特徴点と呼ぶ）．その特徴点を出発点として，隣接する頂点へと領域を拡張していく．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．

3.2. 特徴点の求め方

3．1 で述べた特徴点は，各領域に埋め込める情報量が最大となるよう，全ての領域の頂点数がほぼ同じになるように選びたい．そのためには，メッシュ上で特徴点がなるべく等間隔になるように特徴点を選ぶ必要がある．

手動で特徴点を指定すると，領域分割に形状特徴をある程度反映できるが，指定に手間がかかる．特に，多数の領域に含まれる頂点数がほぼ等しくなるように特徴点を配置するのは困難である．そこで，我々は自動的に特徴点を選ぶ手法を考案した．

我々は，平面に正三角形を充填するとその三角形の頂点を中心に面積が互いに等しいボロノイ領域が作られることに注目した．これを 3 次元メッシュに近似的に当てはめ，与えられたモデルの表面に擬似的に正三角形を充填することで領域分割を行うアルゴリズムを考案した．以下にそのアルゴリズムを示す．ここでいう「距離」とは，グラフ上の 2 点間の位相距離が最短の道を求め，これをポリゴンメッシュにおける距離の近似として用いた．

(1) ユーザが特徴点の個数 F（分割する領域数）と，1 つの頂点を特徴点として指定する．
(2) 操作(1)で指定した特徴点から「距離」R にある頂点の集合 V_1 を求める．
(3) V_1 からユーザが特徴点として 1 つ選ぶ．
(4) 操作（3）で求めた頂点から「距離」R にある頂点の集合 V_2 を求める．
(5) V_1 と V_2 が重なるところを次の特徴点にする．
(6) 新たに決めた特徴点からも同様に V_i $(i = 3, \ldots, F)$ を求め，V_{i-1} と重なる頂点を次の特徴点とする．（この時，今まで決めてきた特徴点と近過ぎない点を選ぶ．）
(7) 操作（6）を繰り返し，すべての特徴点を決めていく．

(2)で指定する距離 R は，各領域に含まれる頂点数からを元に，実験的に求めた数値を使っている．

図 1 は, (1)で選ばれた頂点と, (3)で求めた集合V_1の一例である. ユーザは, 集合V_1から1点を選ぶと, 残りの特徴点が自動的に決まる.

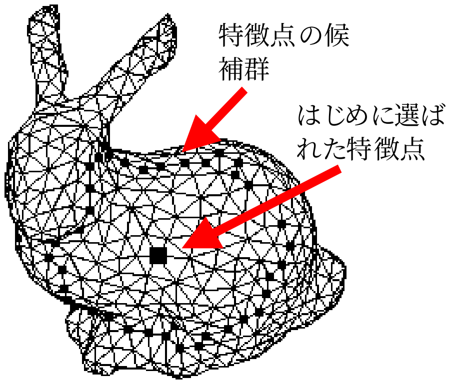

図 1　特徴点の探索

4. リサンプリングを用いた妨害耐性強化法

透かしの入ったポリゴンメッシュに加えられる操作の中には, 頂点の接続性を変更するようなもの (メッシュ簡単化等) がある. 我々が提案した手法[大渕 00]では, このような操作を受けた透かしモデルからは透かしを取り出だせなかった. そこで, 本論文では, メッシュ簡単化など, 頂点接続性を変更する妨害に耐える透かしを実現するため, リサンプリングを用いた. ここでいうリサンプリングとは, 頂点接続性を変更するような妨害を加えられた透かしモデルから, 被覆モデルの頂点接続性と, 透かしモデル (メッシュ簡単化などの操作を受けている) の幾何形状とを持つモデルを作り出す操作のことである. そのリサンプリングによって作り出したモデルから透かしを取り出す.

4.1. アルゴリズム

本手法では, リサンプリングを次のように行う. 説明の都合上, 被覆モデルM (頂点数n) と透かしモデルM'の位置合わせは終わっているものとする.

(1) Mと全く同じ頂点の座標値と接続性を持ったモデルM_cを作成する.
(2) M_cの頂点$\mathbf{v}_i$ $(i=1,2,\ldots,n)$に接する面の法線ベクトルの平均を, $\mathbf{v}_i$の法線ベクトル$\mathbf{n}_i$として計算する.
(3) $\mathbf{v}_i$を通り, $\mathbf{n}_i$を方向ベクトルとする直線l_iを考え, M'の表面とl_iとの交点を求める (図 2).
(4) 先の求めた交点のうち, 幾何的な距離が$\mathbf{v}_i$に最も近い点 (リサンプル点) の座標で$\mathbf{v}_i$の座標を置き換える.

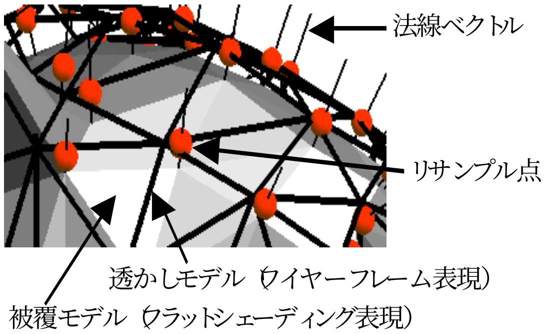

図 2　リサンプリング (モデル表面拡大図). 説明の都合上, 被覆モデルは縮小してある.

上述した方法でリサンプリングを行うとき, 問題が 2 点ある. 第 1 点は, 被覆モデルの頂点$\mathbf{v}_i$に対するリサンプル点$\mathbf{v}'_i$が求まらない場合 (例えば, 図 3 頂点$\mathbf{a}$の法線方向には透かしモデルがなく, $\mathbf{a}$からリサンプル点が求まらない) である. この場合, $\mathbf{v}'_i$の座標値を$\mathbf{v}_i$の座標値と同じにする. 第 2 点は, $\mathbf{v}'_i$が適切に求まらない場合 (例えば, 図 3 頂点$\mathbf{b}$に対する理想的なリサンプル点は, 形状が類似した頂点$\mathbf{b}'$であるが, 実際は$\mathbf{b}$の法線方向に$\mathbf{b}'$はなく, $\mathbf{b}$とは形状の対応しない点$\mathbf{c}$がサンプル点となってしまう) である. この場合, $\mathbf{v}_i$と$\mathbf{v}'_i$との幾何的距離が, あるしきい値を超えたとき, $\mathbf{v}'_i$の座標値を被覆モデルの頂点$\mathbf{v}_i$の座標値に変更する.

このリサンプリング処理は, 被覆モデルと透かしモデルの位置が一致していないと使えない. 頂点接続性が異なるモデル同士の位置合わせの詳細については報告の予定である.

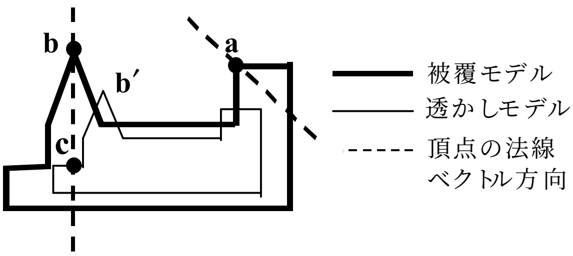

図 3　リサンプリングの失敗の例 (2 次元で図示).

4.2. リサンプリング処理時間の短縮

上述したリサンプリング手法の計算時間は, 被覆モデルの頂点数nと透かしモデルの面数mとに依存し, ?$(n\times m)$で処理時間が増加する. そ

こで，我々は，空間の等方分割を用いた光線・物体交差判定高速化技術[GraphicsGems94]を応用し，ある法線と交差判定を必要とする面数を刈り込むことでリサンプリング処理を高速化した．具体的には図 4（2 次元で図示）のように行う．まず被覆モデル M と透かしモデル M' とが存在する空間を 3 次元正方格子状に分割する．次いで，M の頂点の法線が触れる格子（複数）に所属する面（複数）と法線との間でのみ詳細な交差判定を行う．

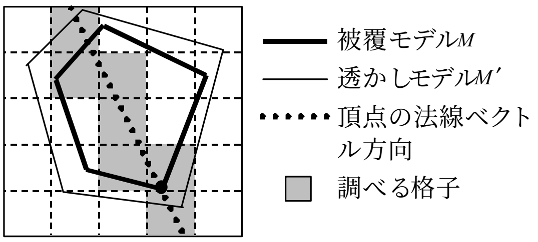

図 4 計算する面数を減らして，計算時間を短縮．

5. 実験結果

我々は上記で述べたアルゴリズムを C++を用いて実装し，実験を行った．

5.1. 領域分割

我々は，頂点数の違う3 つのモデルを使って，1 つの領域に含まれる頂点数が，約 400 程度になるように領域の個数を指定して分割を行った．

ユーザが手動で出発点となる特徴点 2 つを選んだ後，自動で残りの特徴点を求め領域分割を行った結果，表 2 に示すような頂点数を持つ領域に分割することができた．しかし，初期特徴点の選び方によっては，複数の領域の間でそれぞれの領域に含まれる頂点数のばらつきが大きくなり，領域ごとの埋め込み可能情報量にばらつきが生じてしまうこともあった．

表 2 領域分割した結果．

モデル	頂点数	分割数	頂点数最大値	頂点数最小値
bunny2	2218	5	637	345
bunny3	4114	10	934	254
bunny4	13990	34	851	262

表 2 に示したような領域分割を行うための特徴点の探索にかかった時間と，その特徴点を用いて分割を行った後に，各領域に対してスペクトル分解のための固有値計算を行った時間を表 3 に示す．領域分割にかかった時間は，その殆どが特徴点を探索する時間であった．

表 1 と表 3 を比較すると，透かし埋め込み処理時間の殆どを占めるスペクトル分解の処理時間が領域分割により減ったことが分かる．そのため，無分割では実用的な時間でできなかった頂点数が10000 を越えるモデルでも，スペクトル分解を施すことが可能になった．

しかし，特徴点の探索に多くの計算時間がかかった．この時間の殆どは，既存の特徴点の近くに新たな特徴点を配置してしまわないようチェックするため，新たな特徴点の候補から既に特徴点と決まっている全ての点までの距離を計算時間である．

表 3 固有値計算時間と領域分割に要した時間．

モデル	頂点数	領域数	処理時間 固有値計算	処理時間 領域分割
bunny2	2218	5	8m8s	21s
bunny3	4114	10	21m16s	2m46s
bunny4	13990	34	53m12s	31m26s

しかし，頂点数の多い領域を作るために，はじめに決める「距離」を大きく取ると，特徴点が決まらないことがあった．これは，被覆モデルが球と同じ位相を持っているため，特徴点の候補となる集合が重ならないことがあるためである．

5.2. リサンプリング

本手法で埋め込まれた透かしは，リサンプリングを用いれば，メッシュ簡単化によりある程度幾何形状が変化した透かしモデルからでも透かしが取り出せる．我々は，distcap モデル（頂点数 686，面数 1368）を被覆モデルとして，次のように実験を行った．

(1) 被覆モデル M に 32 ビットの透かしを埋め込み，透かしモデル M' を作る．
(2) M' にメッシュ簡単化を行い，モデル M'' を作る．
(3) M'' にリサンプリングを行う．
(4) リサンプリングによってできたモデルと M にスペクトル分解を行い，透かしを取りだす．

以上のように，実験を行った結果を図 5 に示す．実験には，3 種類の透かし（全て 0，全て 1，ランダム）を使用した．図中，頂点減少数は，メッシュ

簡単化によってモデルから取り除いた頂点数である．また，エラービット率の算出方法は，取り出しに失敗したビット数（エラービット数）を調べ，条件（拡散率 c，振幅率 α，頂点減少数）が全て等しい 3 種類の透かしのエラービット数から平均をとり，透かしビット数 32 で割る．

図 5 からわかるように c，α が増加するに従い，メッシュ簡単化に対する耐性が増した．

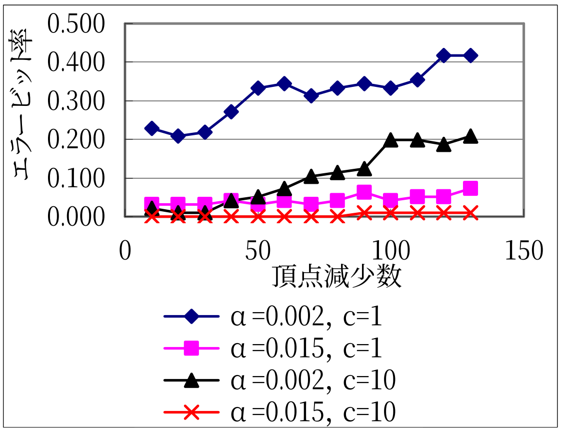

図 5　メッシュ簡単化に対する耐性の実験結果．

6. まとめと今後の課題

本論文では，以前我々が発表した，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手法[大渕 00]に，メッシュを幾つかの領域に分割してその領域単位で透かしの処理を行うことで処理を高速化する手法と，被覆モデルを頂点接続性の参照モデルとして，接続性が変更されたメッシュの形状をリサンプリングする手法を追加した．

既発表の手法では，メッシュの頂点数が増えるに連れ，透かし処理の時間が急激に増大した．また，メッシュに頂点接続性を変更するような操作を加えられると透かしが壊れた．我々が行った実験から，今回提案した手法により，これらの欠点をある程度解決した．

本論文で述べた手法にも問題点がある．まず，本手法による領域分割を行うと，各領域に含まれる頂点数が大きく異なり，領域によって埋め込める透かしの情報量が大きく異なる場合がある．これには，全ての特徴点を決めた後に，各特徴点を必要に応じて近傍の頂点に変更する等の微調整をする手法を導入したい．また，メッシュ簡単化と相似変換を組み合わせると，透かしが取り出せないことがある．これには，モデルの位置あわせのアルゴリズムを改良することにより対処したい．

7. 参考文献

[Bollobás98] B. Bollobás, Modern Graph Theory, Springer, 1998.

[Benedens99] O. Benedens, Geometry-Based Watermarking of 3D Models, *IEEE CG&A*, pp. 46-55, January/February 1999.

[GraphicsGems94] Paul S. Heckbert. Ed, Graphics Gems 4. AP Professional, 1994

[Kanai98] S. Kanai, H. Date, and T. Kishinami, Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition, Proc. of the *Sixth IFIP WG 5.2 GEO-6*, pp. 296-307, Tokyo, Japan, December 1998.

[Karni00] Zachi Karni, Craig Gotsman, Spectral Compression of Mesh Geometry, Proceedings of the *SIGGRAPH* 2000, pp. 279-286, July 2000, New Orleans, U.S.A.

[Katzenbeisserr00] S. Katzenbeisser, F. A. P. Petitcolas, *Digital Watermarking*, Artech House, London, 2000.

[Karypis98] G. Karypis and V. Kumar, 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: http://wwwusers.cs.umn.edu/ ~karypis/metis/metis.html

[Ohbuchi97] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models, *Proc. ACM Multimedia '97*, Seattle, Washington, USA, November 1997, pp. 261-272.

[Ohbuchi98a] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, pp. 551-560, *IEEE JSAC*, May 1998.

[Ohbuchi98b] R. Ohbuchi, H. Masuda, and M. Aono, Geometrical and Non-geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models, *Computer Communications*, Vol. 21, pp. 1344-1354, Elsevier (1998).

[Praun99] Emil Praun, Hugues Hoppe, Adam Finkelstein, Robust Mesh Watermarking, Proc. *SIGGRAPH '99*, pp. 49-56, 1999.

[Taubin95] G. Taubin, “A Signal Processing Approach to Fair Surface Design”, Proc. ACM SIGGRAPH ’95, pp. 351-358, 1995.

[Yeo99] -L. Yeo and M. M. Yeung, Watermarking 3D Objects for Verification, *IEEE CG&A*, pp. 36-45, January/February 1999.

[Wagner00] M. G. Wagner, Robust Watermarking of Polygonal Meshes, Proc. *Geometric Modeling & Processing 2000*, pp. 201-208, Hong Kong, April 10-12, 2000.

[大渕00]大渕 竜太郎，高橋 成雄，宮澤 貴彦，向山明夫，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，情報処理グラフィクスCAD研究会，2000年9月，Vol. 2000, No. 78, pp. 7-12

[松井98]松井 甲子雄，電子透かしの基礎，森北出版，1998年8月