

## 大規模 3 次元メッシュを対象とした頑強な電子透かし

向山 明夫<sup>1</sup>, 大淵 竜太郎<sup>2</sup>, 高橋 成雄<sup>3</sup>

k7186@kai.yamanashi.ac.jp, ohbuchi@acm.org, takahashis@acm.org

<sup>1</sup>山梨大学大学院 工学研究科電子情報工学専攻

<sup>2</sup>山梨大学 工学部 コンピュータ・メディア工学科

<sup>3</sup>東京大学大学院 総合文化研究科広域システム科学系

### 要旨

以前我々が発表した 3 次元ポリゴンメッシュを対象とする電子透かし手法は, そのメッシュの幾何形状の「低周波」成分をメッシュスペクトルの領域で変更することで透かしの埋め込みを行った。同手法はメッシュスムージング, 部分切り取り, メッシュ単純化などに耐性を持った。しかし, 同手法はスペクトル分解にかかる計算コストが高く, さらに, 複数の操作を合わせた妨害, 例えば, メッシュ単純化・相似変換・部分切り取りを合わせた妨害などで透かしが壊れる, という弱点を持っていた。本論文ではこれらの弱点を改善する手法を提案する。まず, 計算コストを下げるため新たな固有値分解手法を導入した。また, より高精度の位置合わせを導入することで複数の操作を合わせた妨害に対する耐性を付加した。

キーワード: グラフ・ラプラシアン, メッシュスペクトル, 情報セキュリティ, 情報埋め込み。

### A Robust Watermarking Algorithm for Large-scale 3D Meshes

Akio Mukaiyama<sup>1</sup>, Ryutarou Ohbuchi<sup>2</sup>, Shigeo Takahashi<sup>3</sup>

<sup>1</sup>Graduate School of Engineering, Yamanashi University

<sup>2</sup>Computer Science Department, Yamanashi University

<sup>3</sup>Graduate School of Arts and Sciences, University of Tokyo

### Abstract

Our previous watermarking method for a three-dimensional (3D) polygonal mesh watermarked the mesh by modifying the “low-frequency” component of the mesh’s geometry in its mesh-spectral domain. Watermarks produced by the method are resistant against mesh smoothing, partial resection, and mesh simplification. However the method suffered from high computational cost of the mesh spectral analysis and the watermarks produced are destroyed by attacks that combines, for example, mesh simplification, similarity transformation and resection. This paper proposes remedies for these two shortcomings. To reduce computational costs, we applied an efficient eigen-analysis algorithm. Introduction of a better algorithm to register original and watermarked mesh improved resiliency of the algorithm against the combined attacks described above.

Keywords: Graph Laplacian, mesh spectra, information security, information hiding.

### 1. はじめに

電子透かしと呼ばれる技術は, 情報を表現する構造体(透かし)を, 埋め込み対象となるコンテンツ(静止画像, 動画画像, 音声データ, 3次元(3D)モデル, 等)に付加する(詳しくは[Katzenbeisser00, 松井 98]等を参照). 付加する際には, 透かしの存在が埋め込み対象コンテンツの本来の目的(例えば人による表示・鑑賞)を阻害しないように, かつ, 透かしがコンテンツから容易に除去できないように

する必要がある。埋め込まれた透かしは, 改ざんの検出, 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ことが多い。

3D データ, 特にその形状を対象とする透かし手法の多くは 3D ポリゴンメッシュの形状を対象としている([Kanai98, Praun99, Yin01, 大淵 01, 向山 01]など). このなかで, 我々が先に提案した 3D メッシュの形状を対象とする透かし [大淵 01, 向山 01]は, メッシュの変換領域で透かしの付加する

手法である (図 1, 説明は後述). この手法で用いる変換は, ポリゴンメッシュ形状のスペクトル分解 [Karni00] である. この透かしは, 相似変換(回転, 平行移動, 一様スケーリングの組み合わせ), 頂点座標へのランダムノイズ重畳, 形状のスムージング [Taubin95], メッシュの部分的切り取りに対する耐性を持っていた.

本論文では, 以前我々が提案した 3 次元ポリゴンメッシュの電子透かし手法 [向山 01] を改良した手法を提案する. 改良の目的は, (1) 相似変換・メッシュ単純化・部分切り取りを組み合わせた妨害に対する耐性を透かしに持たせること, および, (2) 透かし処理を効率化し大規模なメッシュに対し透かし処理ができるようにすること, である. 前者は, モデルの一部に基づく高精度の位置合わせ手法, 後者はより効率的な固有値分解計算アルゴリズムを用いて達成した.

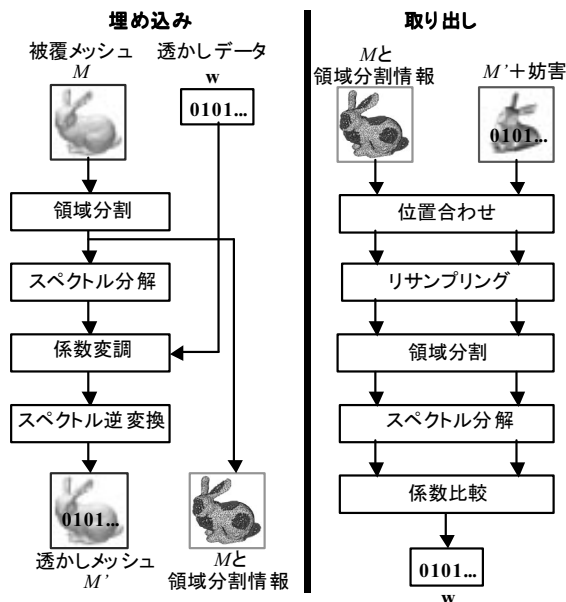


図 1. スペクトル分解を用いた電子透かしの処理.

本手法を用いることで, 次に挙げる先行研究 [向山 01] の欠点を改善できる.

- 1 相似変換・メッシュ単純化・部分切り取りを合わせた妨害に対する耐性.
- 1 固有値分解の処理時間.

改良の結果, 本透かしは, 相似変換・メッシュ単純化・部分切り取りを合わせた妨害に耐性を持ち, かつ, 頂点数数万のメッシュに対しても埋め込めるようになった. 例えば実験の結果, 本手法を用いることで, 頂点数約 48000 のメッシュに透かしを

埋め込み, 単純化によって頂点数を半分以上に減らし, 相似変換と部分切り取りを行ったメッシュからでも損失なく透かしを取り出すことができた.

本論文の構成は以下のとおりである. 第 2 章では関連研究を紹介する. 第 3 章では, 基本となる透かし処理, 位置合わせ手法の改善手法, 固有値分解処理の効率改善手法について述べる. そして, 第 4 章では実装と実験の結果を紹介し, 第 5 章でまとめと今後の課題について述べる.

## 2. 関連研究

メッシュに対する電子透かしの中で, 変換領域に埋め込む透かしを最初に提案したのは Kanai らである [Kanai98]. Kanai らはメッシュをウェーブレット変換し, その変換領域に透かしを埋め込んだ. この透かしはアフィン変換, および頂点にランダムノイズを付加するような妨害に対して耐性を持った. しかし, 欠点として埋め込み対象が 1 対 4 の再分割連続性を持つメッシュに限られた.

一種の変換領域での透かしとして考えることができる手法に Praun らの手法 [Praun99] がある. この透かしは相似変換, 切り取り, ランダムノイズの重畳などの妨害に対して耐性がある. しかし, この透かしの欠点は, 埋め込み対象となるメッシュが頂点数の多いメッシュに限られることである.

Burt-Adelson ピラミッド [Guskov99] を用いて変換領域に透かしを埋め込む手法として Yin らの手法 [Yin01] がある. この透かしは, Praun らの手法と同様に, さまざまな妨害に対し耐性を持つ.

本論文で提案する透かしは, 上述した手法と異なり, スペクトル分解 [Karni00] を用いた変換領域で埋め込む. 本透かしは, kanai らの手法と違い, 埋め込み対象を 1 対 4 の細分割連続性を持つメッシュに限定しない. また, 本透かしは Praun らの透かしと同様にさまざまな妨害に対して耐性を持つ. さらに, Praun らの透かしに比べ, 頂点あたりに埋め込める情報量が多い.

## 3. スペクトル分解による電子透かし

### 3.1. 透かし処理の基本アルゴリズム

本手法において, 基本的な透かし処理は先行手法 [向山 01] と同様の処理を行う (図 1). 透かしの埋め込みでは, 元となるメッシュ (被覆メッシュ) をスペクトル分解 [Karni00] して得られた形状の「周波数」表現であるスペクトル係数の値を透かし情報に応

じて変更する。透かしの入ったメッシュは、固有ベクトルと変更したスペクトル係数の一次結合で求まる。透かしの取り出しの際には、透かしを付加したメッシュ(透かしメッシュ)と被覆メッシュの双方を使用する(このような透かしを「秘密透かし」と呼ぶ)。透かしメッシュと被覆メッシュを共にスペクトル分解し、得られたスペクトル係数を比較して行う。

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スペクトル係数からなる。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、その中で、本手法および先行研究ではキルヒホフ行列[Bollobás98]を用いる。メッシュスペクトルは、およそ、絶対値の小さな固有値に対応する固有ベクトルがメッシュの概形(低周波成分)を、大きな固有値に対応する固有ベクトルがメッシュの詳細(高周波成分)を、それぞれ表現する。

透かしを埋め込む際(図1左), まず, 被覆メッシュ $M$ に対し領域分割を行う。複数の小領域に分け各小領域ごとに同一の透かしを埋め込めば, 部分切り取りに対する耐性が得られる。また, 領域分割の情報は取り出しの際に使用するので保存しておく。次いで, 透かしデータ $\mathbf{w}$ を拡散率 $c$ 回複製する。同じデータを繰り返し埋め込むことで, ランダムノイズ重畳などに対する耐性を得る[Hartung98]。複製した透かしデータに応じてスペクトル係数を変調振幅 $\alpha$ だけ変更することで透かしデータを埋め込む。ここで $\alpha$ は, 被覆メッシュ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長 $\phi$ に対する割合(振幅率) $\beta$ を指定することで $\alpha = \phi \cdot \beta$ のように指定する。透かしを埋め込んだメッシュ $M'$ (透かしメッシュ)は透かしで変調された係数を逆変換して得られる。

本手法は秘密透かしであり, 取り出しには透かしメッシュ $M'$ と被覆メッシュ $M$ とが必要である(図1右)。取り出しでは, まず $M'$ と $M$ の位置・大きさ・向きを揃え(これを「位置合わせ」と呼ぶ), 次いでリサンプリングにより $M'$ 上に埋め込み時と同様の頂点接続性が作られる。さらに,  $M$ と $M'$ 上には埋め込み時と同一の領域分割も施される。こうして得られた2セットの小領域をスペクトル分解しその係数を比較する。その比較結果と埋め込みの際

に使用した拡散率 $c$ とから透かしデータ $\mathbf{w}$ を復元する。

### 3.2. 位置合わせ手法の改善

今回の改良では, 最適化アルゴリズムを使用した高精度の位置合わせを用いる。まず手動で位置合わせをする領域を選択し, 慣性主軸を使用したおおまかに位置合わせ[向山 01]を行い, さらにモデルの小領域を用いて最適化アルゴリズムを使用した詳細な位置合わせを行う。本手法を用いることで先行研究の欠点であった, メッシュ単純化・相似変換・部分切り取りを合わせた妨害に対する耐性を改善できる。

本手法では, 最適化アルゴリズムを使用した位置合わせには, Praun らが使用した位置合わせ[Praun99]と同様に, Chen らの手法[Chen92]を応用した。最適化を行う対象となる値は次のように求める(図2)。まず, 透かしメッシュ $M'$ (面数 $n$ )のある面の重心 $\mathbf{g}_i$  ( $i=1,2,\dots,n$ )の法線と, 被覆メッシュ $M$ との交点 $\mathbf{r}_i$ を求める。次いで,  $\mathbf{r}_i$ が存在する面分を含む平面 $f_i$ に直行し,  $\mathbf{g}_i$ を通過する直線を求め, その直線と $f_i$ の交点 $\mathbf{h}_i$ を求める。そして,  $\mathbf{g}_i$ と $\mathbf{h}_i$ との幾何的な距離 $d_i$ を計算する。後は同様に, 他の面の重心から $d_i$ を求め, それらを合計する。この合計の値が最適化の対象となる。 $M'$ に回転, 平行移動, 一様スケーリングを徐々に加え, 合計が最小となるように最適化を行う。また,  $M'$ の面数が多い場合, メッシュ全体に散らばるように面を選び, 選んだ面の重心を使って最適化を行えば良い。

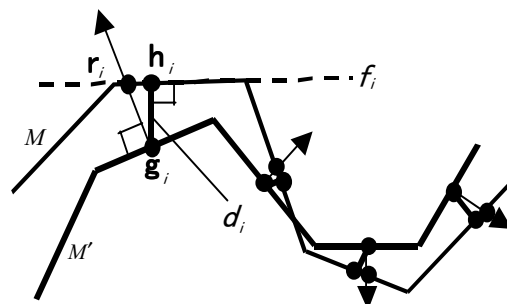


図2. メッシュ間の誤差の計算(2次元で図示)。

### 3.3. 固有値分解処理の効率改善

今回, スペクトル分解の処理時間を改善するために Arnoldi 法[Golub96]を導入する。Arnoldi 法により頂点数が数千のメッシュに対しても実用的な時間でスペクトル分解を行うことが可能になる。

Arnoldi 法とは, 次元が大きい正方行列に対し,

固有値分解を行う手法である。この手法を行うことで、値が最も小さい、および大きい固有値とそれに対応する固有ベクトルから求めて行くことができる。先行手法[向山 01]では頂点数数百程度のメッシュが透かしの対象であった。この理由は、スペクトル分解のために行う固有値分解の際、全ての固有値・固有ベクトルを求めるため計算時間が非常にかかるためであった。本透かし手法では埋め込みに必要な固有値・固有ベクトルだけで十分で、全ての固有値・固有ベクトルを求める必要はないため Arnoldi 法によりスペクトル分解の処理時間を改善できる。

本手法では、Arnoldi 法を導入するため、Arnoldi 法を含んだライブラリ[Stewart]を用いた。このライブラリの Arnoldi 法を用いるためには、求める固有値・固有ベクトルの数(大きい固有値から求めるものと小さい固有値から求めるものとの総数)を指定する必要がある。例えば、 $n$  次元の正方行列に対し、 $2m(2m < n)$  の固有値・固有ベクトルを求めるよう指定した場合、固有値が小さいほうから  $m$  個、大きいほうから  $m$  個の固有値・固有ベクトルを求める。そのため、 $2m \ll n$  になると、 $n$  個全ての固有値・固有ベクトルを求める場合に比べて大幅に処理時間の改善ができる。

表 1. Arnoldi 法の計算時間。

メッシュの頂点数	求めた固有値の数		
	500	1000	1500
2218	1m05s	4m55s	13m56s
7565	3m56s	13m20s	30m46s
18957	13m38s	35m25s	1h13m18s

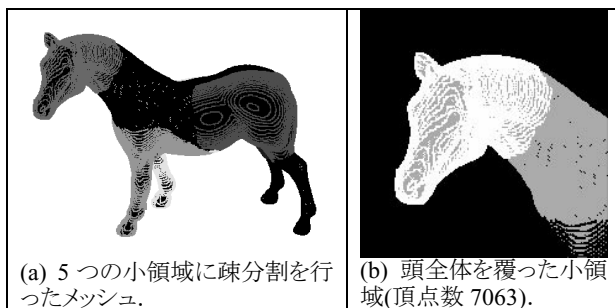


図 3. 特徴的な部分を含んだ疎分割を行ったメッシュ horse2(頂点数 48485, 面数 96966)。

表 1 に、メッシュの頂点数と求める固有値・固有ベクトルの数との計算時間の関係を示す。使用したのは CPU が Athlon1900+, メモリが 1.5GB の PC である。先行研究の手法では、処理時間が大きすぎるために頂点数が数千のメッシュの固有値分解は困難だった。しかし、表 1 からわかるように、

Arnoldi 法を用いれば頂点数が数千以上のメッシュの固有値分解が実用的時間で可能である。

上述のように Arnoldi 法を用いることで頂点数が多いメッシュに対して固有値分解を行うことが可能となった。そのため、総頂点数が約 5 万のメッシュの特徴的な部分にも透かしの埋め込める。図 3 はメッシュ horse2(頂点数 48485, 面数 96966)を 5 つの小領域に疎分割[向山 01]した例である。各小領域の頂点数は約 7000 である。図 3(b)に示すように、頂点数約 7000 の小領域で頭全体を覆うことができる。このように、Arnoldi 法を用いることで大規模なメッシュの特徴的な部分に固有値分解を行うことができ、透かしが埋め込めるようになった。

## 4. 実験結果

我々は 2 節で述べたアルゴリズムを C++ と GUI ツールキット fltk (<http://www.fltk.org>) を用いて実装し、実験を行った。

### 4.1. 改良した位置合わせによる耐性強化

今回最適化アルゴリズムを用いることで位置合わせを改良した。そこで最適化アルゴリズムの有無で妨害に対する耐性の変化を調べた。

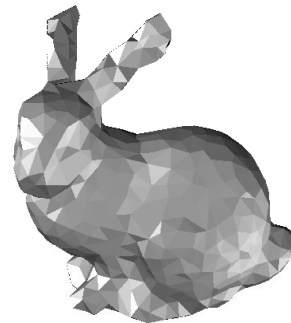


図 4. メッシュ bunny1(頂点数 646, 面数 1288)。

図 5 は、bunny1 (図 4)に対して、変調振幅率  $\beta$  と拡散率  $c$  とをある値にし、無分割で 32 ビットの透かしの埋め込み、メッシュ単純化によって頂点数を減らした場合のビットエラー率を示している。ここで、ビットエラー率とは、取り出しに失敗したビットの数を、埋め込んだ透かしのビット数で割った値である。取り出した値に誤りがなければ 0 となる。ここでは 3 種類の透かし(全て 0, 全て 1, ランダム)を使用した。図 5 には、その 3 種類の透かしの平均ビットエラー率を示した。図 5(a)は、位置合わせの際には慣性主軸の位置合わせだけを、図 5(b)は慣性主軸と最適化アルゴリズムを用いた位置合

わせを行った結果である。

図 5(a)(b)からわかるように、慣性主軸と最適化アルゴリズムを用いた位置合わせを行うことでビットエラー率がさらに減少する。

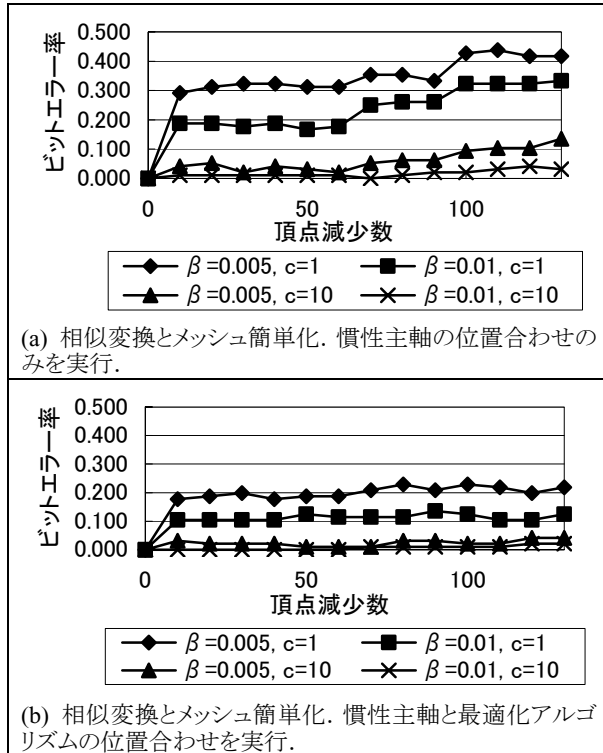


図 5. メッシュ単純化を加えた際の頂点減少数とビットエラー率。

#### 4.2. 大規模メッシュに対する透かしの埋め込み

頂点数が約 5 万の比較的大きなメッシュに対し実験を行った。図 6 にメッシュ horse(頂点数 48485, 面数 96966)を用いた実験結果の例を示す。図 6(a)のメッシュを図 6(b)のように疎分割を行い、各小領域に透かしを 32 ビット埋め込んだメッシュが図 6(c)である。このとき振幅率  $\beta = 0.001$ , 拡散率  $c = 10$  である。図 6(a)(c)のように、透かし埋め込み前後では見た目にはそれほど違いがない。

図 6(d)~(i)は図 6(c)のメッシュにメッシュ単純化やリメッシュ(MeshToss[MeshToss01]を使用), それらの操作と他の操作を合わせた妨害を与えたメッシュである。本手法を用いれば, 図 6(d)~(i)のようなメッシュからでも, 損失なく透かしを取り出すことができた。ただし, 透かしを取り出す際には, 各領域から取り出した透かしを使用し修正を行い, 最終的な透かしを決定した。修正方法は, 各領域から取り出した透かしの同じビットを調べ, 0 と 1 のう

ち数が多いものをそのビットの最終的な透かしとする。ただし, 図 6(d)(i)のような部分切り取りされたメッシュの場合は, 切り取られずに残った領域から透かしを取り出し, 修正を行った。また, 取り出しの際には, 図 6(d)(g)にはリサンプリングのみを, 図 6(e)(h)には位置合わせとリサンプリングを使用した。図 6(f)(i)の場合は, ある程度手動で領域を選択し, 選択した領域を使用して位置合わせを行った後, リサンプリングを行った。

図 6(f)(i)のように, 部分切り取りが行われたメッシュに対しては, 位置合わせを行う部分がある程度手動で指定する必要があった。これは切り取りによってメッシュのどの部分が残るかわからないためである。本手法では, 疎分割を使用して領域を指定し, その領域に対して位置合わせを行った。

メッシュ単純化やリメッシュ, 部分切り取りによって透かしメッシュの頂点数を大幅に減らすと, 透かしの取り出しに失敗した。この原因は, メッシュの形状が大きく変形してしまい位置合わせがうまく行かなかったためである。

#### 5. まとめと今後の課題

本論文では, 以前我々が提案した手法[向山 01]を改善する提案を行った。今回の改善により, 相似変換・メッシュ単純化・部分切り取りを合わせた妨害に対する耐性が得られ, また大規模メッシュに透かしを埋め込めることができるようになった。具体的には, 位置合わせ手法を改善してほぼ任意の部分メッシュを用いて位置合わせができるようになったこと, また, Arnoldi 法で固有値分解の計算を行ったことである。その結果, 例えば頂点数が約 48000 の大きなメッシュに透かしを埋め込むことができ, さらに, その透かしメッシュを単純化して頂点数を半分以上に減らし, 相似変換と部分切り取りを行ったメッシュからでも損失なく透かしを取り出すことができた。

今後, リサンプリングのための位置合わせ手法の更なる改善が必要である。現在, メッシュ単純化, 部分切り取り, 幾何変換の 3 種を合わせた妨害が加えられた場合, 手動で位置合わせを行う領域を選択する必要がある。このような合わせた妨害が加えられた場合でも, 自動で位置合わせが行えるようにしたい。

謝辞: 本研究は文部科学省科学研究費補助金(課題番号 12680432)からの助成による。

## 参考文献

- [Bollobás98] B. Bollobás, *Modern Graph Theory*, Springer, 1998.
- [Chen92] Y. Chen, G. Medioni: "Object modelling by registration of multiple range images," *Image and Vision Computing* 10, 3(Apr. 1992), pp. 145-155.
- [Golub96] Golub, G. H., Van Loan, C. F., *Matrix Computations*, Third Edition, Johns Hopkins University Press, 1996.
- [Guskov99] I. Guskov, W. Swelden, P. Schröder, Multiresolution signal processing for meshes, Proc. *SIGGRAPH '99*, pp. 325-334, 1999.
- [Hartung98] F. Hartung, P. Eisert, and B. Girod, Digital Watermarking of MPEG-4 Facial Animation Parameters, *Computer and Graphics*, 22(4), pp. 425-435, Elsevier, 1998.
- [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] Z. Karni, C. Gotsman: "Spectral Compression of Mesh Geometry," Proc. *SIGGRAPH 2000*, pp. 279-286, 2000.
- [Katzenbeisser00] S. Katzenbeisser, F. A. P. Petitcolas: "Digital Watermarking", Artech House, London, 2000.
- [MeshToSS01] 金井 崇, MeshToss, Version 1.0.1, <http://graphics.sfc.keio.ac.jp/MeshToSS/indexE.html>.
- [Praun99] Praun, E., Hoppe, H. and Finkelstein, A.: "Robust Mesh Watermarking," Proc. *SIGGRAPH '99*, pp. 49-56(1999).
- [Stewart] David E. Stewart, Zbigniew Leyk, *Mesach Library* Version 1.2b, <http://www.netlib.org/c/mesach>.
- [Taubin95] G. Taubin.: "A Signal Processing Approach to Fair Surface Design," Proc. *SIGGRAPH '95*, pp. 351-358, 1995.
- [Yin 01] K. Yin, Z. Pan, J. Shi, Robust mesh watermarking based on multiresolution processing, *Computer and Graphics*, 25, pp. 409-420, 2001.
- [大淵01] 大淵 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の埋め込み, 情報処理学会論文誌, 2001年5月, 42(5), pp. 1103-1114.
- [松井98] 松井 甲子雄, 電子透かしの基礎, 森北出版 (1998)
- [向山01] 向山 明夫, 宮澤 貴彦, 高橋 成雄, 大淵竜太郎, スペクトル変換領域で埋め込む3次元メッシュの電子透かし, 画像電子学会 情報処理学会 Visual ComputingグラフィクスとCAD合同シンポジウム2001, 2001年6月.

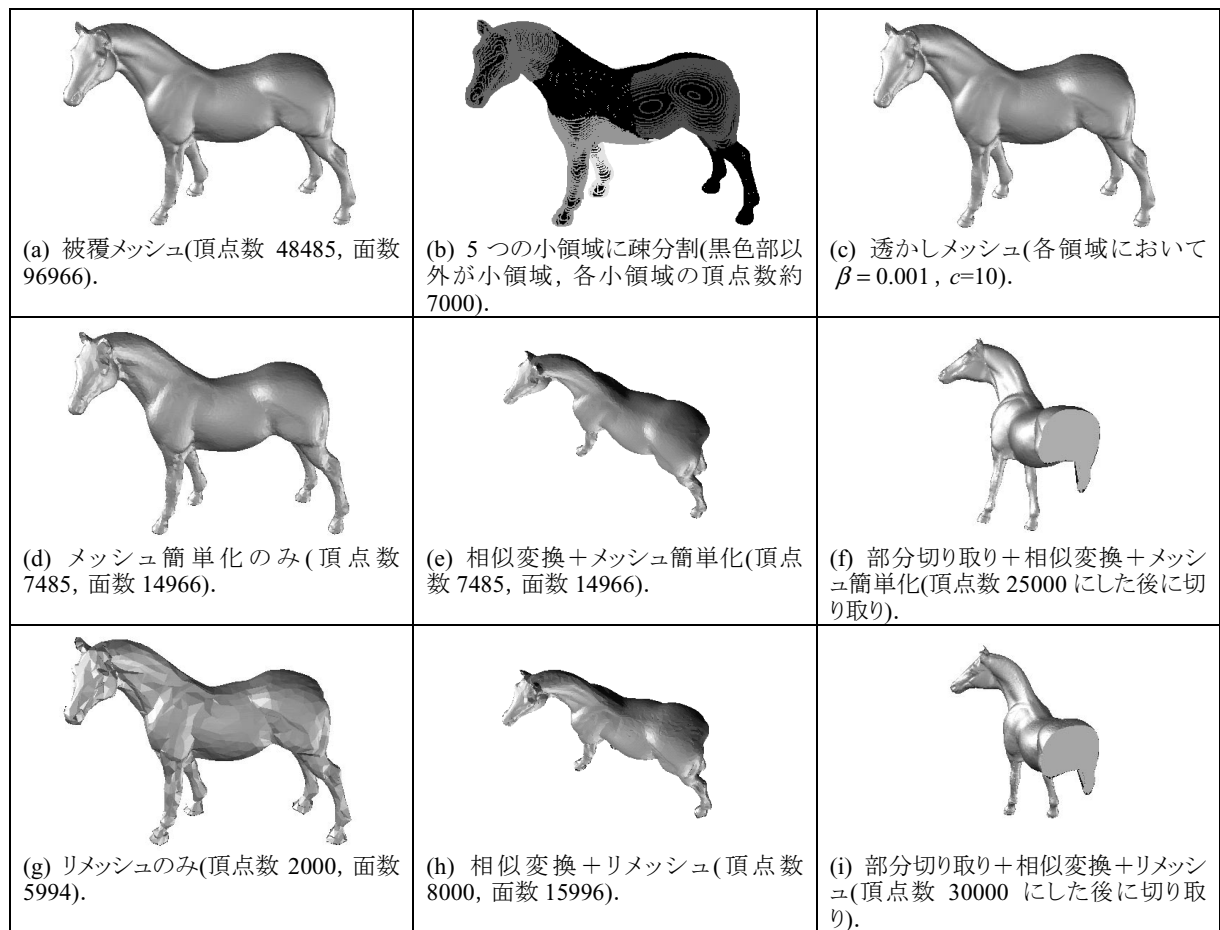


図 6. メッシュ horse を使用した実験結果の例.