

SSA アルゴリズムを用いた 3 次元多角形メッシュへの電子透かしの埋め込み

室谷 浩平[†] 杉原 厚吉[†]

電子透かしの埋め込みとは、メディアに何らかの構造を付与することによって、隠したい情報を埋め込む技術のことである。そして、その応用には、説明の追加、改竄の検出、正規購入者の認証などがある。電子透かしの多くの手法では、秘密情報（透かし）はスペクトル係数に隠されている。そのため、何らかのスペクトル分解が必要になる。本研究は、3次元多角形メッシュに透かしの埋め込み手法を提案する。この手法では、3次元多角形メッシュの形状を適当な頂点列で表現されるものとする。まず、頂点列のスペクトルを、時系列解析で用いられるアルゴリズム Singular Spectrum Analysis (SSA) を用いて計算し、そのスペクトルの係数の中に透かしの埋め込みを行う。このとき、頂点は、元の位置から少しだけずれたものに変換されるが、埋め込みの自由度を利用して、変換後の頂点の重心が変換前の重心にできるだけ近づけた。これによって、透かしの復元の安定性を増やすことができた。この手法によって埋め込まれた透かしは、相似変換（回転、平行移動、一様拡大縮小）や頂点座標に付加されたランダムノイズに対して頑強である。

Watermarking 3D Polygonal Meshes Using the Singular Spectrum Analysis

MUROTANI KOHEI[†] and SUGIHARA KOKICHI[†]

Watermarking is to embed a structure called a watermark into the target data such as images. The watermark can be used, for example, in order to secure the copyright and detect tampering. This paper presents a new robust watermarking method that adds a watermark into a 3D polygonal mesh in the spectral domain. In this algorithm, a shape of a 3D polygonal model is regarded as a sequence of vertices called a vertex series. The spectrum of the vertex series is computed using the singular spectrum analysis (SSA) for the trajectory matrix derived from the vertex series. Watermarks embedded by this method are resistant to similarity transformations and random noises.

1. はじめに

電子透かしの埋め込みとは、対象となるデータに電子透かしと呼ばれる秘密情報を付加する技術のことである。電子透かしの埋め込みに当たって要求される要件は主に 2 つある。1 つ目は、電子透かしの存在が、情報が隠される場所であるオリジナルデータの本来の目的を阻害しないことである。例えば、静止画や動画、音楽に電子透かしの埋め込みを行うとき、鑑賞者に気付かれるような埋め込みであってはいけないということである。2 つ目は、電子透かしはオリジナルデータから分離することは困難であるべきであるということである。電子透かしは、著作権の保護や個人認証などを目的に使われることが多いため、このような要件が求められる。

これまで電子透かしの埋め込み対象となっていたも

のは、いわゆる古典的なデータ構造を持つもの（文書データ、2次元静止画、2次元動画、音声、楽曲など）が多かったため、その様なデータ構造を持つものに対する研究は多くなされてきた¹³⁾。しかし、近年になってからは、3次元データ構造をもつ対象（3次元多角形メッシュ、3次元 CAD データ等）がより一般的に使われるようになってきたため、この様なデータ構造を持つものに対しても電子透かしの埋め込みことが求められてきた¹⁾⁴⁾¹²⁾。

そこで、本論文は、この様な要求に応えるために、3次元多角形メッシュに電子透かしの埋め込み新たな手法を提案するものである。本論文の手法は、対象となるメッシュに平行移動、回転、一様拡大縮小（相似変換と呼ぶ）の編集が行われても、安定的に透かしを取り出すことができる。さらには、透かしの破壊することを目的とした、ランダムノイズを頂点座標に付加されても、透かしの安定的に取り出すこともできる。

多くの電子透かしの技術では、秘密情報（電子透かし）はスペクトル係数に隠される。それゆえに、電子

[†] 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology
University of Tokyo

透かしを埋め込む際には何らかのスペクトル分解が必要となる．文献 [9] ではラプラシアン行列を固有値分解することによって，文献 [4] では特定のクラスのメッシュに対して Wavelet 展開を行うことによって，電子透かしを埋め込むことに成功している．だが，この 2 つの手法は，メッシュの接続構造に基づいてスペクトル分解を行っているため，メッシュの接続構造の変更には脆弱である．これに対して，本手法は，メッシュの接続構造に依存していないため，その様な変更に対して頑強である．

まず，2 章で本手法の核となるアルゴリズムの Singular-Spectrum Analysis について説明し，3 章でそれをういた電子透かしの埋め込みと取り出しについて述べる．そして，4 章で実験を通して本手法の性能を検証し，5 章でまとめと今後の課題を述べたい．

2. 基本 SSA

Singular-Spectrum Analysis (SSA)⁽²⁾³⁾ は，新しい時系列解析の手法である．最近では，SSA は，気象学や気候学の分野において特徴検出の一般的な手法の一つとなっている．

電子透かしの多くの技術において，秘密情報はスペクトル係数に隠される．それ故に，これらの電子透かしの技術では，何らかのスペクトル分解が必要になる．本論文では，一種の時系列のスペクトル分解である SSA を電子透かしに用いている．

この章では，基本 SSA のアルゴリズムについて述べる．そして，次の章では，多変量時系列に一般化した SSA を私達の目的に当てはめる．

基本 SSA は分解ステップと再構成ステップの 2 つのステップから成り立っている．

2.1 分解ステップ

分解ステップは，更に次のような 2 ステップから成り立っている．

2.1.1 1st step: 埋め込み

N を $N > 2$ の整数とし，実数値を要素とする系列 $F = (f_0, f_1, \dots, f_{N-1})$ について考える．まず最初に，オリジナルな系列から，次のような方法でベクトルを構成する． L を $1 < L < N$ の整数として， $K = N - L + 1$ とする．そして，ベクトル X_i を

$$X_i = (f_{i-1}, \dots, f_{i+L-2})^T, \quad 1 \leq i \leq K \quad (1)$$

と定義する． X_i を， L ラグベクトルと呼ぶことにする．そして，系列 F の L 軌道行列を

$$X = [X_0 : \dots : X_K], \quad (2)$$

と定義する．書き換えると， L 軌道行列は L 行 K 列の行列

$$X = (x_{ij}) = \begin{pmatrix} f_0 & f_1 & f_2 & \cdots & f_{K-1} \\ f_1 & f_2 & f_3 & \cdots & f_K \\ f_2 & f_3 & f_4 & \cdots & f_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{L-1} & f_L & f_{L+1} & \cdots & f_{N-1} \end{pmatrix} \quad (3)$$

となる．このようにして作られた軌道行列 X は，Hankel 行列となっている．そして， N と L が固定されれば，軌道行列 X と系列 F の間には一対一の対応が存在する．

2.1.2 2nd step: 特異値分解

次に，前ステップで求めた軌道行列を特異値分解する． S を $S = X X^T$ とする． $\lambda_1, \dots, \lambda_L$ を S の固有値を大きい順に並べたものとする．そして， U_1, \dots, U_L を S の固有値とし， d を $d = \text{Max}\{i, \text{such that } \lambda_i > 0\}$ とする． V_i を $V_i = X^T U_i / \sqrt{\lambda_i}$ と定義し， $X_i^T = \sqrt{\lambda_i} U_i V_i^T$ ($i = 1, \dots, d$) とおく．こうして，軌道行列 X の特異値分解は

$$X = X_1 + X_2 + \dots + X_d \quad (4)$$

と書くことができる． X_i はランク 1 の行列である．

2.2 再構成ステップ

2.2.1 3rd step: 平均化

最後のステップで，(4) で分解された各ランク 1 の行列を長さ N の新たな系列に変換する． Y を $y_{ij}, 1 \leq i \leq L, 1 \leq j \leq K$ ，の要素を持つ $L \times K$ 行列として，

$$g_k = \begin{cases} \frac{1}{k+1} \sum_{m=1}^{k+1} y_{m, k-m+2} & \text{for } 0 \leq k < L-1, \\ \frac{1}{L} \sum_{m=1}^L y_{m, k-m+2} & \text{for } L-1 \leq k < K, \\ \frac{1}{N-k} \sum_{m=k-K+2}^{N-K+1} y_{m, k-m+2} & \text{for } K \leq k < N \end{cases} \quad (5)$$

のように変換にする．式 (5) は $i + j = k + 2$ が成り立つ要素の平均を求めている．例えば， $k = 0$ では $g_0 = y_{11}$ ， $k = 1$ では $g_1 = (y_{12} + y_{21})/2$ といった具合にである．

分解された行列 X_k を式 (5) に当てはめると，系列 $\tilde{F}^{(k)} = (\tilde{f}_0^{(k)}, \dots, \tilde{f}_{N-1}^{(k)})$ が得られ，初期の系列 $F = (f_0, \dots, f_{N-1})$ はそれら d 個の系列の和として

$$F = \sum_{k=1}^d \tilde{F}^{(k)} \quad (6)$$

のように表現される．

3. スペクトル領域での電子透かしアルゴリズム

本論文の透かし埋め込みアルゴリズムは、3次元多面体の表面多角形メッシュ(これを以下では、3次元多角形メッシュと呼ぶ)に秘密情報を埋め込むものである。これまでの3次元多角形メッシュへの電子透かし埋め込みは、メッシュの接続構造を用いた手法であったため、メッシュの接続構造の変更に対して頑強でなかった。一方、私達の手法は、メッシュの接続構造を用いず、代わりに、3次元多角形メッシュを頂点列として扱う。それ故、私達の手法は、メッシュの接続構造の変更に対して頑強である。

3.1 頂点列のスペクトル分解

前章で説明した基本 SSA では、各要素が 1 実数値である系列 $F = (f_0, f_1, \dots, f_{N-1})$ について考えられてきたが、各要素が 3 実数値である系列 $F = (F_0, \dots, F_{N-1})$, $F_i = (f_{i,x}, f_{i,y}, f_{i,z})^T$, を用いるために、基本 SSA を多変量版に拡張する。軌道行列 (2) を、 $3L \times K$ 行列

$$X = \begin{pmatrix} F_0 & F_1 & F_2 & \cdots & F_{K-1} \\ F_1 & F_2 & F_3 & \cdots & F_K \\ F_2 & F_3 & F_4 & \cdots & F_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F_{L-1} & F_L & F_{L+1} & \cdots & F_{N-1} \end{pmatrix} \quad (7)$$

に拡張し、そしてその軌道行列 (7) に対して特異値分解を行う。近似的にはあるが、大きな特異値は低空間周波数に対応し、小さな特異値は高空間周波数に対応している。つまり、大きな特異値に対応するランク 1 の行列はメッシュの大まかな部分を表現するものであり、一方、小さな特異値に対応するランク 1 の行列はメッシュの詳細な部分を表現するものである。

3.2 電子透かしの埋め込み

0 または 1 の値を要素とする m 次元ビットベクトル a を、埋め込む秘密情報とする。各 a_i は

$$b_i = a_j, \quad jc < i \leq (j+1)c \quad (8)$$

によって c 回複製され、長さ $m \times c$ の $b = (b_1, b_2, \dots, b_{mc})$, $b_i \in \{0, 1\}$ に変換される。このように同じビットを c 回繰り返して埋め込むことによって、透かしの耐久性は向上する。次いで、 $b' = (b'_1, b'_2, \dots, b'_{mc})$, $b'_i \in \{-1, 1\}$ は

$$b'_k = \begin{cases} -1 & \text{for } b_i = 0, \\ 1 & \text{for } b_i = 1 \end{cases} \quad (9)$$

で変換されて導かれる。

$i = 1, 2, \dots, mc$ に対して、 $p_i \in \{-1, 1\}$ を後に述べるように適切な方法で選ぶ。変調振幅と呼ばれる正の定数 α を用いて、ここで、 i 番目の特異値は

$$r_i = \sqrt{\lambda_i} + b'_i p_i \alpha \quad (10)$$

のように変更される。 α は、透かしの深さを表すパラメータである。式 (10) の r_i , $i = 1, 2, \dots, d$ を用いれば、

$$\begin{aligned} X' &= \sum_{i=1}^d r_i U_i V_i^T \\ &= \sum_{i=1}^d \sqrt{\lambda_i} U_i V_i^T + \sum_{i=16}^d b'_i p_i \alpha U_i V_i^T \end{aligned} \quad (11)$$

のようにして、行列を求めることができる。式 (11) の右辺の第 2 項の $i = 16$ となっている理由は次節で説明する。この行列から頂点座標 $F' = (F'_0, \dots, F'_{N-1})$, $F'_i = (f'_{i,x}, f'_{i,y}, f'_{i,z})^T$ は、(5) から求められる。その結果、オリジナルメッシュの頂点は、元の位置から少し移動した、透かしの入った頂点に変更される。

最後に、 p_i がランダムに 1 か -1 の値を取るとしたら、オリジナルメッシュの重心 $\phi(F)$ と透かしメッシュの重心 $\phi(F')$ がかなりずれてしまう。ただし、 $\phi(F)$ は頂点列 F の重心であるとする。透かしを取り出す際に、この重心の普遍性は重要であるので、 $\phi(F) = \phi(F')$ を満たすのが望ましい。しかし、厳密に $\phi(F) = \phi(F')$ を満たすのは困難であるので、式 (11) の右辺の第 2 項の値の絶対値ができるだけ小さくなるように p_i を選択する。つまり、最小化問題

$$\begin{aligned} \text{Min} \left\| \sum_{i=16}^d b'_i p_i \phi \circ \psi(U_i V_i^T) \right\|_2 \\ \text{s.t. } p_i \in \{-1, 1\}. \end{aligned} \quad (12)$$

を近似的に解いて、最適に近い解 p_i を求める。ただし、 ψ は一般の行列を受け取り、(5) の変換を受けて頂点列を返す写像である。

3.3 電子透かしの取り出し

透かしの取り出しの際には、透かしの埋め込まれたメッシュと透かしを埋め込む前のオリジナルメッシュの双方を用いる。透かしの埋め込まれたメッシュには、平行移動や、回転、一様拡大縮小の変換(これを相似変換と呼ぶ)が加えられているものとする。取り出しの処理の際には、相似変換を加えられた透かしメッシュをオリジナルメッシュと幾何学的に重ね合わせる必要がある。まず、オリジナルメッシュと透かしメッシュの重心が一致するように、透かしメッシュの頂点を平衡移動する。次に、両方のメッシュを SSA を用いて

分解し、その内の特異値の大きい方から 15(ここでは 15 としたが、15 である必要はない) 個の X_i を用いてメッシュの大まかな形を求める。式 (11) から、特異値の大きい方から 15 個の X_i は透かしの影響は受けていない。そして、両方のメッシュの概形に対して頂点座標の 3×3 の共分散行列を求め、それを固有値分解して、それぞれのメッシュの主軸を求める。この 2 つの主軸から回転と一様拡大縮小を検出することができる。

オリジナルメッシュの頂点列を F とし、SSA を用いて求められる特異値を $\sqrt{\lambda_i}$ とする。そして、幾何的に重ね合わせた透かしメッシュを F' とし、SSA を用いて求められる特異値を $\sqrt{\lambda'_i}$ とする。ここで、 $(\sqrt{\lambda'_i} - \sqrt{\lambda_i})$ に p_i をかけて c 回足し合わせると、 q_j は

$$q_j = \sum_{i \in I_j} (\sqrt{\lambda'_i} - \sqrt{\lambda_i}) p_i \approx \sum_{i \in I_j} b'_i p_i^2 \alpha, \quad (13)$$

$$I_j = \{j, j+m, j+2m, \dots, j+(c-1)m\}$$

となる。頂点座標に外乱が無ければ、

$$q_j = b'_j \alpha c \quad (14)$$

となり、 q_j は $-ac$ か ac のいずれかの値をとる。 α と c は正の定数であるので、 q_j の正負を判定すれば埋め込んだ秘密データ a_j が

$$a_j = \{sign(q_j) + 1\} / 2 \quad (15)$$

のようにして求まる。このようにして、埋め込まれた秘密データを取り出すことができる。

3.4 頂点列の分割

大きなメッシュに電子透かしを埋め込む時には、図 2 の (a) にあるように取り扱いやすいサイズのメッシュに分解して、それぞれのサブメッシュに対して本手法を用いるとよい。このように、メッシュのサイズを小さくすることによって、計算量は減り、精度は上がる。図 2 の (b) は、各領域が Voronoi 領域に従うように頂点を 5 つに分割したものである。

4. 実験結果

4.1 パラメータ

4.1.1 拡散率 c

静止画や 2 次元の動画への電子透かしに用いられる拡散率 c は、透かしを入れる際に扱われるデータのサイズが数万単位であるため、比較的大きな値のものを用いることができる。だが、3 次元多角形メッシュで扱われるデータサイズは、それほど大きいとは限らない。例えば、本論文では、頂点数 1494、面数 2915 の "bunny" モデルを用いた。そこで、埋め込むデータ

を 50 ビットとし、拡散率 c を 15 とした。扱うメッシュのデータサイズは一定であるので、より大きな拡散率をとると、埋め込むことができるデータサイズが低下してしまうが、電子透かしとしての頑強性は上がる。

4.1.2 変調振幅 α

透かしを埋め込む深さを表す変調振幅 α を、対象となっているメッシュを覆うことができるボックスの最大辺を l 、振幅率を β とした場合、 $\alpha = \beta \times l$ と定義する。図 1 には、 $\beta = 0.01, 0.1, 1$ の結果が載っている。変調振幅 α の値が大きければ大きい程、電子透かしの頑強性が上がるが、見た目は保存されなくなる。

4.1.3 頂点列

私達の電子透かしの埋め込み方では、頂点列を色々選ぶことができ、その選び方によって透かしメッシュの外見や頑強性が変わるかもしれない。今回は、図 2 の (b) にあるように、TSP (Traveling Salesman Problems) を解いて頂点列を求めた。

4.2 頑強性

この節では、相似変換とランダムノイズの重畳に対する私達の手法の頑強性を、実験を通して考察する。

4.2.1 相似変換

3.3 節にあるような方法で、オリジナルメッシュと透かしメッシュの重心ができる限り一致するように透かしを埋め込んでいるため、私達の透かし埋め込みは相似変換に対して頑強である。

p_i をランダムにとると、オリジナルメッシュと透かしメッシュの重心が、ユークリッドノルムで約 $0.01 \times \alpha$ だけずれる。これに対して、(12) による方法で適切な p_i を選んだら、2 つのメッシュ間の重心のずれは、 $10^{-5} \times \alpha$ 程度となった。

ここで、透かしメッシュにランダムに相似変換を加えた後、透かしを取り出す実験を 100 回行った結果、ランダムに p_i を選んだ場合では、50 ビット中 20.12 ビットの平均正答が得られ、(12) による方法で適切な p_i を選んだ場合では、50 ビット中 50.00 ビットの平均正答が得られた。この結果から、(12) のオリジナルメッシュと透かしメッシュの重心を合わせる方法によって、相似変換に対して頑強になったのだと考えられる。

4.2.2 ランダムノイズの重畳

図 3 は、振幅 $\alpha \times \gamma$ ($\beta = 0.1$) でランダムノイズを頂点座標に加えたメッシュである。図 3 の (a) は $\gamma = 0.01$, (b) は $\gamma = 0.1$, (c) は $\gamma = 1$ のランダムノイズが加えられたメッシュである。このように頂点座標にランダムノイズを加え、透かしを取り出す実験を 100 回行った。図 4 は、埋め込まれた 50 ビットの内

正しく復元できた数を表すものである。各縦のバーは一回の実験の結果を表している。赤色のバーは $\gamma = 1$, 黄色のバーは $\gamma = 0.1$, 緑色のバーは $\gamma = 0.01$ の結果を表し, これらを順に重ねている。図 4 の (a) は私達の方法で透かしを取り出した結果で, $\gamma = 1, 0.1, 0.01$ で正しく値を返す数はそれぞれ, 21.14, 45.47, 50.00 であった。(b) は文献 [9] の方法で透かしを取り出した結果で, $\gamma = 1, 0.1, 0.01$ で正しく値を返す数は, それぞれ, 32.31, 49.13, 50.00 であった。この実験から, およそ $\gamma < 0.1$ のランダムノイズに対しては透かしは頑強であることが分かる。さらには, 私達の方法は, 文献 [9] の方法と比べて頑強性は若干劣るものの, それほど差は見られないことを確認することができた。

5. 今後の課題

私達は, 3次元多角形メッシュへの電子透かし埋め込みの新しい手法を提案した。本手法は, オリジナルメッシュから作られる軌道行列を特異値分解し, 透かしの埋め込みものである。

本手法で埋め込まれた透かしは, 相似変換と頂点座標への適度なランダムノイズの付与に対して頑強であり, そして, 埋め込める情報量が比較的高密度である。 m のデータを埋め込むには, 少なくとも $\lfloor 3(m+1)/4 \rfloor$ の頂点数のメッシュであればよい。もし, より大きなメッシュに透かしの埋め込み時には, 頂点列を幾つかの扱いやすいサイズの頂点列に分解することができる。そうすれば, 計算量は減り, 計算精度も上がる。

今後の課題として, 多角形メッシュの幾何的特長と電子透かしの性能の間にどのような関係があるかを調べていきたい。まず, 頂点列をどのように取ればよいかを調べたい。本論文では, 頂点列を TSP を用いて求めたが, この他にも様々な方法で頂点列を作ることができる。頂点列の選び方によって, 電子透かしの性能は変わるのかを実験を通して調べたい。また, メッシュを分割して電子透かしの埋め込んだが, この分割の方法によって電子透かしの性能は変わるかも調べたい。

謝 辞

本研究は, 文部科学省 21 世紀 COE プログラム「情報科学技術戦略コア」, 及び, 科学研究費補助金基盤研究 (S)15100001 の援助を受けている。

参 考 文 献

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

2) Galka, A., *Topics in Nonlinear Time Series Analysis*, World Scientific, pp. 49-71, 2001.

3) Golyandina, N., Nekrutkin, V., and Zhigljavsky, A., *Analysis of Time Series Structure-SSA and Related Techniques*, Chapman & Hall/CRC, 2001.

4) Kanai, S., Date, H., and Kishinami, T., Digital Watermarking for 3D Polygons Using Multiresolution Wavelet Decomposition, *Proceedings of the Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6)*, pp. 296-307, Tokyo, Japan, December 1998.

5) Ohbuchi, R., Masuda, H., and Aono, M., Watermarking Three-Dimensional Polygonal Models, *Proceedings of the ACM International Conference on Multimedia '97*, pp. 261-272, Seattle, USA., November, 1997.

6) Ohbuchi, R., Masuda, H., and Aono, M., Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, *IEEE Journal on Selected Areas in Communication*, Vol. 16, No. 4, pp. 551-560, May, 1998.

7) Ohbuchi, R., Masuda, H., and Aono, M., Targeting Geometrical and Non-Geometrical Components for Data Embedding in Three-Dimensional Polygonal Models, *Computer Communications*, Vol. 21, pp. 1344-1354, October, 1998.

8) Ohbuchi, R., Masuda, H., and Aono, M., A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces, *Proceedings of the Computer Graphics International'99*, pp. 180-177, Canmore, Canada, June 7-11, 1999.

9) Ohbuchi, O., Takahashi, S., Miyazawa, T., and Mukaiyama, A., Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain, *Proceedings of the Graphics Interface 2001*, pp. 9-17, Ontario, Canada, June 2001.

10) Praun, E., Hoppe, H., Finkelstein, A., Robust Mesh Watermarking, *ACM SIGGRAPH 1999*, pp. 69-76, 1999.

11) Wagner, M. G., Robust Watermarking of Polygonal Meshes, *Proceedings of Geometric Modeling & Processing 2000*, pp. 201-208, Hong Kong, April 10-12, 2000.

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

13) 松井 甲子雄, 電子透かしの基礎, 森北出版, 1998.

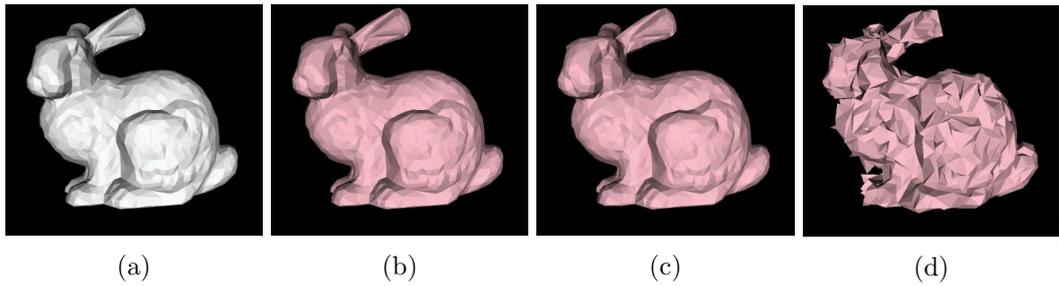


図 1. 変調振幅 α , 拡散率 $c = 15$ の電子透かしを埋め込まれた, "bunny" モデル (1494 vertices, 2915 faces). 埋め込まれた秘密情報は 50 ビット. (a) はオリジナルメッシュ. (b) は $\beta = 0.01$ で埋め込まれた透かしメッシュ. (c) は $\beta = 0.1$ で埋め込まれた透かしメッシュ. (d) は $\beta = 1$ で埋め込まれた透かしメッシュ.

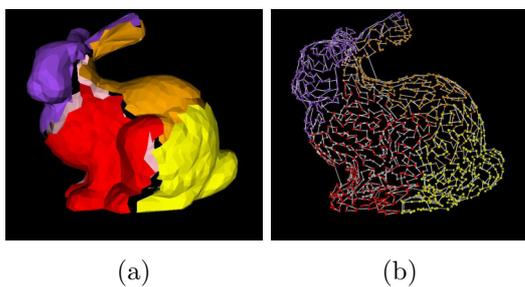


図 2. 頂点の分割と TSP によって作られた頂点列. (a) は頂点の数が同じようになるように 5 つのグループに分割したメッシュ. (b) は各領域毎に TSP を解いて求めた頂点列.

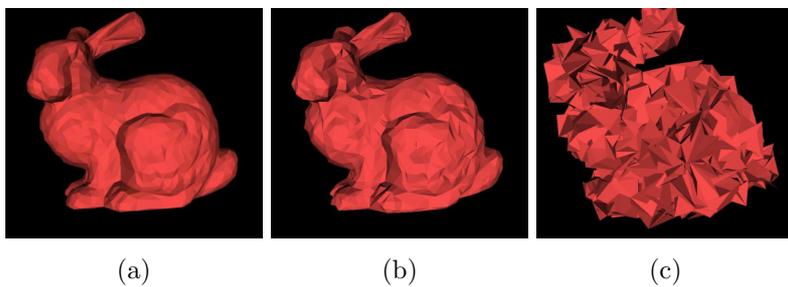


図 3. 振幅 $\alpha \times \gamma$ ($\beta = 0.1$) のランダムノイズを頂点座標に加えたメッシュ. (a) は $\gamma = 0.01$ のランダムノイズが加えられたメッシュ. (b) は $\gamma = 0.1$ のランダムノイズが加えられたメッシュ. (c) は $\gamma = 1$ のランダムノイズが加えられたメッシュ.

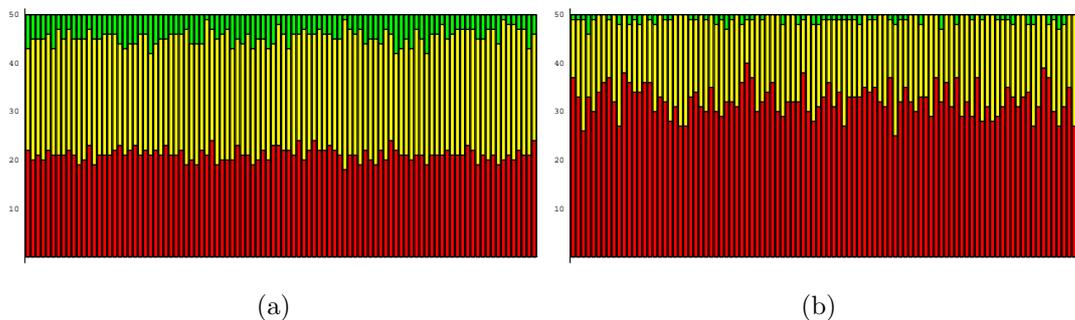


図 4. ランダムノイズを頂点座標に加える実験を 100 回行い, 埋め込まれた 50 ビットの内正しく返した数. 各縦のバーは一回の実験の結果を表している. 赤色のバーは $\gamma = 1$, 黄色のバーは $\gamma = 0.1$, 緑色のバーは $\gamma = 0.01$ の結果を表し, これらを順に重ねている. (a) は私達の方法で透かしを取り出した結果である. $\gamma = 1, 0.1, 0.01$ で正しく値を返す数はそれぞれ 21.14, 45.47, 50.00 である. (b) は [9] の方法で透かしを取り出した結果である. $\gamma = 1, 0.1, 0.01$ で正しく値を返す数はそれぞれ 32.31, 49.13, 50.00 である.