VisualComputing/グラフィックスとCAD 合同 ワークショップ 2003 採録
（情報処理学会・画像電子学会共催）, pp. 195-200, 2003/06/19~20, 小倉.

点群による 3 次元形状モデルを対象とする電子透かし
Watermarking a 3D shape model defined as a point set

向山 明夫 [1], 高橋 成雄 [2], 大渕 竜太郎 [3]
Akio Mukaiyama [1], Shigeo Takahashi [2], Ryutarou Ohbuchi [3]

mukaiyama.akio@lab.ntt.co.jp, takahashis@acm.org, ohbuchi@acm.org

[1] 山梨大学大学院工学研究科 Graduate School of Engineering, Yamanashi University
(現: 日本電信電話株式会社 NTT 情報流通プラットフォーム研究所 NTT Information Sharing Platform Laboratories)
[2] 東京大学大学院総合文化研究科 Graduate School of Arts and Sciences, University of Tokyo
[3] 山梨大学医学工学綜合研究部, Interdisciplinary Grad. School of Medicine and Engineering, Univ. of Yamanashi

1. はじめに

電子透かしは, 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(音声データ, 静止画像, 動画像, 3 次元(3D)モデル, 等)に付加する([松井 98, Cox02] 等). 透かしを付加する際には, 透かしがコンテンツから容易に除去できないように, かつ, 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する必要がある. 埋め込まれた透かしは, そのコンテンツを何らかの形で管理(改ざんの検出, 正規の購入者の認証など)する目的で用いられる.

近年, 3 次元モデルの形状を表現する方法として点群モデル(図 1(a))が注目され始めている. この理由は, 3 次元レンジスキャナにより実在する物体を高密度の点群モデルとして取り込む機会が増えたことである. また, 軽量かつ詳細に形状を表現する目的で点群モデルが使われ始めたという理由もある([Carr01] など). 3 次元点群モデルの研究は今後さらに拡大し, 3 次元モデルの表現手法のひとつとして定着すると予想される.

3 次元点群モデルが一般的に普及した場合, そのモデルの著作権の保護が必要になる. 保護の技術のひとつとして電子透かしがあり, 3 次元ポリゴンメッシュ(図 1(b))を対象とした手法がいくつか提案された([Kanai98, Praun99, Yin01, Ohbuchi02, 向山 02] など). しかし, 3 次元点群モデルに対する電子透かしは, 我々の知る限り, 提案されていない. これは, 3 次元メッシュと比べ, 3 次元点群モデルには次のような特徴があるためと考えられる.

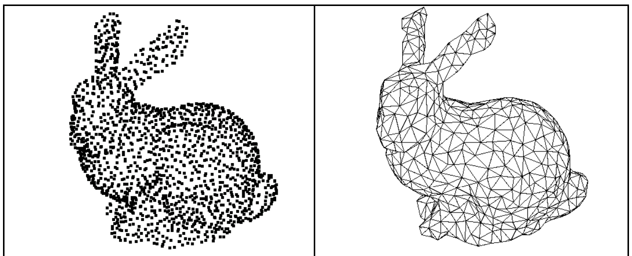
- 頂点の接続性がなく, このため,
- 補間や周波数解析などの処理手法が少ない.

本論文では, 3 次元点群モデル(図 1(a))を対象とした新しい電子透かし手法を提案する. 本透かしは, 極力攻撃に耐えることをめざす頑強な透かしで, かつ, 取り出しに透かし埋め込こみ前の元データを必要とする秘密透かしである.

本手法の基本的なアイデアは, 点群からメッシュを生成し, そのメッシュに対し, 既存のメッシュスペクトル変換による周波数分解を用いた電子透かし手法[向山 02]を適用することである. つまり, 点群モデルの形状を周波数分解し, 得られた係数を透かし情報に応じて変更して透かしを埋め込む. また, 領域分割を用いることで, 透かしを埋め込みたい部分をユーザがある程度指定できる.

ここで, 本手法が透かしのために点群から生成するメッシュは, 一般に, 2-多様体では無い. 生成するメッシュに 2-多様体の要求が無いため, 本手法の透かしメッシュ生成は大変簡便になった.

既存の 3 次元メッシュを対称とした電子透かし手法([Kanai98, Praun99, Yin01, Ohbuchi02]など)は, 本手法の一部として用いた手法[向山 02]を含め, 2 多様体メッシュを仮定する. 2-多様体でないメッシュに対し, もともと 2-多様体用に考えられた透かし手法を適用できるかが問題である. 実験の結果, 本手法のように点群から 2-多様体でないメッシュを生成して透かしを埋め込んでも, かなりの攻撃耐性(相似変換やノイズ付加等の攻撃への耐性)を持ち, 電子透かしとして使えることがわかった.



(a) 点群モデル bunny1. (b) メッシュモデル bunny.

図 1. 点群モデルとメッシュモデルの例.

2. 電子透かしのアルゴリズム

本手法では, 3 次元点群モデルからメッシュを生成し, このメッシュに対し既発表の電子透かし手法[向山 02]を適用し, 点群モデルに対して透かしを埋め込む(図 2).

2.1. 透かし処理の概要

透かしを埋め込む際(図 2 上), まず, 元の点群モデル(被覆モデル)M に対し領域分割を行う. 領域分割の情報は取り出しに使用するので保存しておく. 次いで, 領域ごとに点群からメッシュを生成し, 各領域でスペクトル分解を行う. 透かしデータ $\mathbf{w}$ は拡散率 c 回複製し, 各領域に繰り返し埋め込み, 攻撃耐性を得る[Hartung98]. 透かしの埋め込みは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 α だけ変更して行う. ここで α は, 被覆モデルを囲う最小の直方体である Axis-Aligned Bounding Box (AABB)を計算し, その辺の長さの最大長 ϕ に対する割合(振幅率) β を指定することで $\alpha = \phi \cdot \beta$ のように決定する. 透かしを埋め込んだ点群モデル(透かしモデル)M' は, 変調後のスペクトル係数をスペクトル逆変換して得たメッシュから接続性を取り除き, 点群とした結果である.

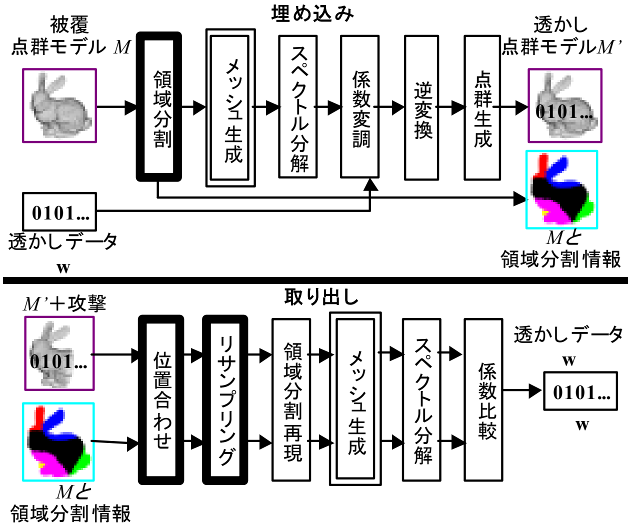


図 2. スペクトル解析を用いた点群形状モデルの電子透かし.

本手法は秘密透かしであり, 取り出しには点群モデル M' と M とが必要である(図 2 下). 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, 点の削除（例えば, 点群モデルの簡単化などの処理による）を想定し, リサンプリングにより M' 上に埋め込み時と同じ点の数に戻す. さらに, M と M'上に, それぞれ, 埋め込み時と同一の領域分割とメッシュ生成を施す. こうして得られた 2 セットの小領域群をスペクトル分解し, 小領域ごとに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（小領域ごとに）透かしデータ $\mathbf{w}$ を復元する.

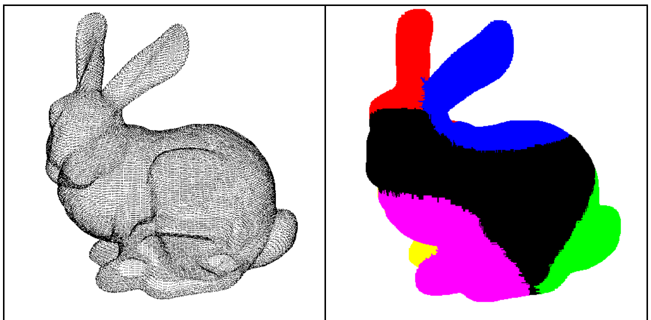
次節より, 図 2 のそれぞれの処理について説明する. 先行手法[向山 02]では, 同図の太線で囲んだ, 領域分割, 位置合わせ, リサンプリングの 3 つの処理にメッシュの接続性を利用した. 今回の手法では, これら 3 つの処理に点群に合わせた新たな手法を用いた. また, 二重線で囲んだメッシュ生成処理は先行手法には存在せず, 今回の手法で新たに追加した.

2.2. 透かし埋め込み処理

2.2.1. 領域分割

領域分割ステップでは, モデルをいくつかの小領域に分割する. 各領域は 3 次元 Euclid 空間で近傍の点の集合からなり, 1 つの透かしメッセージを埋め込む対象となる. その目的は, モデルを分割し, スペクトル分解の手間を押さえることが主である. ユーザが領域分割を指定すれば, 透かしを埋め込む位置をユーザが指定できる. また, 複数の領域に反復して同一の透かしを埋め込めば, 切り取りに耐える透かしが実現できる.

本手法では, 領域の間に隙間を許す疎分割(図 3)を行う. 点群モデルに疎分割を行うには, まず, 与えられたモデルの点群から領域の中心となる点(特徴点)をユーザが選ぶ. そして, その特徴点から 3 次元の Euclid 距離が近い順に周りの点を領域に加えてゆく. この領域の拡張操作を, 領域に含まれる点の数があるしきい値を超えるまで行う. このしきい値はユーザが指定する.



(a) 点群モデル bunny2.　(b) 疎分割の例(領域数 5).

図 3. 点群モデル bunny4 (点の数 34839)を疎分割した例(黒色以外の部分が選択領域).

2.2.2. メッシュ生成

点群モデルでは接続性がないため, 本手法では点群モデルからメッシュを生成し, このメッシュに先行手法[向山 02]を適用することで, 点群モデルを擬似的な周波数表現に変換する. モデルの各点において, 点を中心とし一定の半径内にある点を隣接する点とみなし, メッシュを生成する. 後は先行手法[向山 02]のスペクトル分解と同様な処理を行いえば良い. 具体的には, 図 4 のように, ある点 p から一定半径 r 以内にある点を接続していると仮定する. あとは, メッシュのスペクトル分解と同様な処理を行うことで周波数表現に変換する.

一定の半径 r の計算方法は, 各点から N_n 個の点を含むような半径を求め, それらを平均することで計算する. 今回 N_n =12 とした. 他にも N_n =24, 36 など

で予備実験したが，透かしの攻撃耐性はあまり変化しなかった．

透かしの場合，このような単純なメッシュ生成でも十分である．なぜなら，透かしに要求されることは攻撃に対する耐性だからである．そのため，人の見た目には良くないメッシュでも，攻撃に耐性があれば透かしとしては十分である．

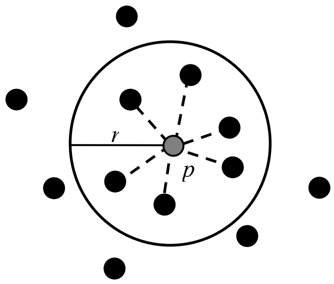


図 4. 接続していると仮定する近傍点(2 次元で図示).

2.2.3. スペクトル分解

本手法では，点群モデルを周波数表現に変換するため，メッシュのスペクトル分解[Karni00]を擬似的に当てはめる．上述したようにメッシュを生成した後，生成したメッシュ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点の座標を射影して，スペクトル係数を得る．また，領域分割を行った場合，各小領域のメッシュに対し個別にスペクトル分解を行う．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，本手法では[Bollobás98]のメッシュラプラシアン行列 $\mathbf{K}$ (別名 Kirchhoff (キルヒホフ)行列)を用いた．

n 個の点からなる小領域から定まる $n \times n$ のキルヒホフ行列 $\mathbf{K}$ 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 次元固有ベクトル $\mathbf{w}_i$ $(1 \le i \le n)$ が得られる．この固有ベクトルを正規化すると基底ベクトル $\mathbf{e}_i$ が得られる．

$$\mathbf{e}_i = \mathbf{w}_i / \|\mathbf{w}_i\| \qquad (1 \le i \le n). \tag{1}$$

n 個の点の座標 $\mathbf{v}_i = (x_i, y_i, z_i)$ $(1 \le i \le n)$ を，その x, y, z 各成分ごと，i 番目の基底ベクトル $\mathbf{e}_i$ に射影すると，スペクトル係数 $\mathbf{r}_i = (r_{s,i}, r_{t,i}, r_{u,i})$ $(1 \le i \le n)$ が得られる．ここで s, t および u は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．

本透かしの処理時間のうち，スペクトル分解のために行う固有値計算の時間がほとんどを占める．そこで本手法では，スペクトル分解処理を効率化するために Arnoldi 法 [Golub96] を導入した．Arnoldi 法により点の数が数千の小領域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．

2.2.4. 係数変調

本手法では，スペクトル係数の振幅を透かし情報に応じて変更し，透かしを埋め込む．スペクトル係数の値を変更して透かしデータ $\mathbf{w}$ を埋め込む際，$\mathbf{w}$ の j 番目のビット w_j (値は{0,1}を取る)は拡散率 c 回複製され，変調信号 $\mathbf{b}$ に変換される．

$$b_i = w_j, \quad j \cdot c \le i < (j+1) \cdot c. \tag{2}$$

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]．次いで，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．変更されたスペクトル係数 $\mathbf{r}'_i = \left(r'_{x,i}, r'_{y,i}, r'_{z,i}\right)$ は次のように計算する．

$$\mathbf{r}'_i = \mathbf{r}_i + b'_i \cdot p_i \cdot \alpha, \tag{3}$$

$\mathbf{r}_i$:変更する前のスペクトル係数,
p_i :あるシード値から生成された{-1,1}の値をとる擬似乱数列,
α :変調振幅.

ここで，変調振幅 α は，被覆メッシュを囲う最小の直方体である AABB を計算し，その辺の長さの最大長 ϕ に対する割合(振幅率) β を指定することで $\alpha = \phi \cdot \beta$ のように求める．

2.2.5. 逆変換

点の座標値に対しスペクトル分解の逆変換をするには，式(4)のように，固有ベクトル $\mathbf{e}_i$ を，透かしを埋め込んだスペクトル係数 $r'_{s,i}$, $r'_{t,i}$ および $r'_{u,i}$ 倍して線形和を計算する．こうすると，透かしを埋め込んだ形状の座の標値 $\mathbf{v}'_i = (x'_i, y'_i, z'_i)$ を得られる．

$$\begin{aligned}
\left(x'_1, x'_2, ..., x'_n\right)^T &= r'_{s,1}\mathbf{e}_1 + r'_{s,2}\mathbf{e}_2 + \cdots + r'_{s,n}\mathbf{e}_n, \\
\left(y'_1, y'_2, ..., y'_n\right)^T &= r'_{t,1}\mathbf{e}_1 + r'_{t,2}\mathbf{e}_2 + \cdots + r'_{t,n}\mathbf{e}_n, \\
\left(z'_1, z'_2, ..., z'_n\right)^T &= r'_{u,1}\mathbf{e}_1 + r'_{u,2}\mathbf{e}_2 + \cdots + r'_{u,n}\mathbf{e}_n.
\end{aligned} \tag{4}$$

2.3. 透かし取り出し

2.3.1. 位置合わせ

精度の高い位置合わせは次のリサンプリングの前提となる．位置合わせの精度が低い場合，リサンプリングがうまくいかない．

本位置合わせは，攻撃によって切り取られたモデルはその影響がない領域を選択して，切り取られていないモデルはモデル全体で行う．まず，両モデルの向きを合わせる．ここでは，モデルの選択した領域または全体の点群を質点とみなして，慣性主軸を求める Gottschalk らの手法[Gottschalk96]を用いる．また，モデルの重心位置を合わせるには，モデルの点群の重心を求め，それらを重ねる．モデルの幾何的な大きさを合わせるには，求めた重心から Euclid 距離が最大になる点を求め，その点と重心との幾何的な距離が等しくなるように一様スケーリングを行う．しかし，この位置合わせは，ランダムに点が削除される等で点の数やその位置が変化したモデルとの位置合わせの場合，精度が低い．

VisualComputing/グラフィックスとCAD 合同 ワークショップ 2003 採録
（情報処理学会・画像電子学会共催）, pp. 195-200, 2003/06/19~20, 小倉.

2.3.2. リサンプリング

位置合わせを行った後，リサンプリングを行うことで，点の削除などに対する耐性を付加する．ここで，リサンプリングとは，攻撃を受けた透かし点群モデル M' から，M' の幾何形状を持ち，被覆点群モデル M と等しい点の数を持つ新たな点群モデル M'' を生成する処理である．この M'' は幾何形状が M' とほぼ同じため，透かしが取り出せる．

本リサンプリングでは，被覆点群モデル M の各点の座標を，その各点から最も近い透かし点群モデル M' の点の座標に置き換える．ただし，Euclid 距離が離れすぎる点と置き換えないように，しきい値を設定する．M のある点から，そのしきい値以内の Euclid 距離に M' の点が存在しなければ，その M 上の点の座標は変更しない．本手法のリサンプリングでは以下のような手順で処理を行う．

1) 被覆モデル M と全く同じ点の座標値を持ったメッシュ M_c をリサンプリング用に作成する．
2) M_c の点 $\mathbf{p}_i$ (i=1,…,m, m はモデルの点の数)から，幾何的な距離が最も短い，透かしモデル M' 上の点を $\mathbf{p}'_i$ とする．
3) $\mathbf{p}_i$ と $\mathbf{p}'_i$ の距離が設定したしきい値以内だったら，$\mathbf{p}_i$ の座標を $\mathbf{p}'_i$ の座標に置き換える．
4) 操作 2)，3)を全ての $\mathbf{p}_i$ に行い，新たなモデル M'' を生成する．

リサンプリングの際に指定するしきい値は，M_c の各点において，それぞれ異なる値を設定する．被覆モデル M において，各点から最も近い点までの距離の 1/2 を各点のしきい値とする．この 1/2 という値は，予備実験の結果最も良かった値である．

2.3.3. 領域分割再現

リサンプリングで作り出したモデル M'' と被覆モデル M とに対し，同じ領域分割を行う．領域分割では，埋め込みの際に保存しておいた領域分割の情報を用いて，M と M'' とを同じ小領域に分割する．

2.3.4. メッシュ生成

埋め込みと同様に，被覆モデル M に対してメッシュ生成を行う．M と M'' との点の数は等しいので，M に対しメッシュ生成を行い，M'' にその結果を適用すればよい．

2.3.5. スペクトル分解

各小領域に対しスペクトル分解を行い，M のスペクトル係数 $\mathbf{r}_i$ と M'' のスペクトル係数 $\hat{\mathbf{r}}_i = \left(\hat{r}_{x,i}, \hat{r}_{y,i}, \hat{r}_{z,i}\right)$ とを得る．取り出しのメッシュ生成同様，M に対し固有値分解を行い，M'' にその結果を適用すればよい．

2.3.6. 係数比較

求めた係数 $\mathbf{r}_i$ と $\hat{\mathbf{r}}_i$ との差分をとり，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 x, y, z の 3 つの成分についてそれぞれ行い，その総和を取る．

$$\begin{aligned} q_j &= \frac{1}{3} \sum_{l \in \{x,y,z\}} \sum_{i=j \cdot c}^{(j+1) \cdot c-1} (\hat{r}_{l,i} - r_{l,i}) \cdot p_i \\ &= \frac{1}{3} \sum_{l \in \{x,y,z\}} \sum_{i=j \cdot c}^{(j+1) \cdot c-1} b'_i \cdot \alpha \cdot p_i^2 \end{aligned} \tag{5}$$

擬似乱数系列 p_i が同期しており，かつ，透かしモデルに対して加えられた各種の攻撃を無視できるとすると，

$$q_j = c \cdot \alpha \cdot b'_i \tag{6}$$

となる．ここで q_j は $\{-\alpha c, \alpha c\}$ の 2 値いずれかをとり，c と振幅率 α は常に正の数である．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 $\mathbf{w} = \left(w_1, w_2, ..., w_m\right)$ は求まる．

$$w_j = sign(q_j) \tag{7}$$

3. 実験結果

2 章で述べたアルゴリズムを実装し，実験を行った．本実験では，透かしを埋め込んだ際のモデルの形状変化と，透かしの攻撃耐性とについて調べた．

ただし，点群モデルに対する処理手法(例えば簡単化[Pauly02]など)は，近年提案されたばかりであり，手元にはなかった．そこで，メッシュモデルから頂点だけを取り出すことで点群モデルを作成し実験を行った．特に，点群モデルに簡単化を行う場合，まず，点群モデルに元のメッシュモデルの頂点接続性を当てはめ，メッシュモデルのリメッシュ[MeshToss01]を行った．その後，リメッシュを行ったメッシュモデルから頂点だけを取り出し，簡単化された点群モデルとした(このような点群の簡単化を，本論文では「擬似的な簡単化」と呼ぶことにする)．

3.1. 透かし埋め込み例

透かしを埋め込んだ場合のモデルの形状の変化を調べた結果を図 6 に示す．図 6 は，図 5(a)のモデルに疎分割を行い透かし 32bit を埋め込んだ例である．この疎分割の際には，図 5(b)のように分割を行い，各領域に同じ透かしを埋め込んだ．埋め込みの振幅率 β は，モデルに対する AABB の最大長の 0.2%(β=0.002)，1.5%(β=0.015)を使用し，拡散率 c=5，10 を用いた．また，スペクトル分解の際にユーザが指定する値 $N_n = 12$ とした．図 6 からわかるように β や c が大きくなると形状品質が劣化する．

この実験では，振幅率を指定する際，モデル全体

の AABB を使用している. しかし, 領域分割を使用して透かしを埋め込む際には, 各領域の AABB を求め, それらの AABB に対して振幅率を指定したほうが良いかもしれない.

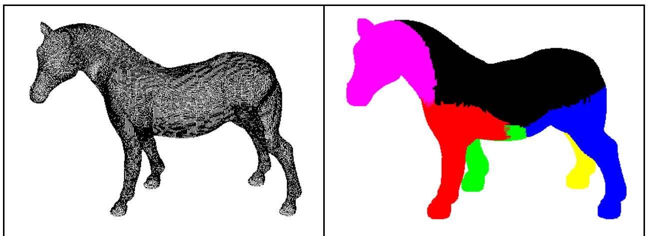

(a) 点群モデル horse1.　(b) 疎分割の例.

図 5. 点群モデル horse1(点の数 48485)とその疎分割(黒色部以外が小領域, 各領域の点の数約 7000)の例.

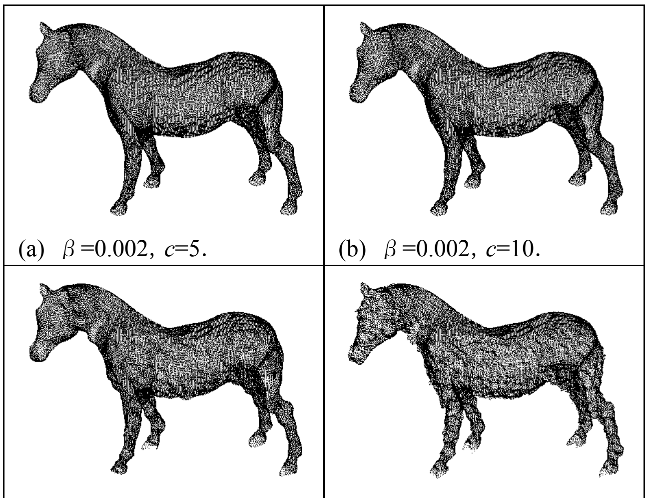

(a) β=0.002, c=5.　(b) β=0.002, c=10.　(c) β=0.015, c=5.　(d) β=0.015, c=10.

図 6. 点群モデル horse1 の透かし埋め込みの例.

3.2. 攻撃に対する耐性

点群モデルに加えられると想定される各種の攻撃に対する耐性について実験を行った. 図 9 に点群モデル bunny2(点の数 34839) を用いた実験結果の例を示す. 図 9(c)は, 図 9(a) に図 9(b)のように疎分割を行い, 透かし 32bit を埋め込んだ例である. 図 9(a)(c) のように, 透かし埋め込み前後では見た目にそれほど違いがない.

本透かしは, 図 9(d)~(f)のような, 攻撃を受けたモデルからでも, 損失なく取り出せる. 図 9(d)~(f)は図 9(c)のモデルにランダムノイズ付加や切り取り, 相似変換, 擬似的な簡単化を与えたモデルである. ただし, 攻撃によって形状が大きく変化した場合, 透かしは壊れた. 例えば, 振幅を AABB 最大長 に対し 0.8%にしてランダムノイズを図 9(c)のモデルに行った場合透かしは壊れた. また, 図 9(c)に対し擬似的な簡単化を行い, 点の数を 5000 にした場合も透かしは壊れた.

また, 振幅率 β や拡散率 c が変化すると, ランダムノイズに対する耐性がどのように変化するか調べた. その結果を図 8 に示す. 図 8 は, horse2 (図 7)に対して, 無分割で透かし 32bit を埋め込み, ランダムノイズを付加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. そのため, 取り出した値に誤りがなければ 0 となる. 図 8 で示すビットエラー率は, 同一の β と c とノイズの振幅率とに対し 20 回ずつ実験を行い, その平均を示している. 図 8 からわかるように, ノイズの振幅率が大きくなればビットエラー率は増えるが, 振幅率 β や拡散率 c が大きいほどランダムノイズに対する耐性が増した. ただし, モデルの形状によって, ビットエラー率は異なる.

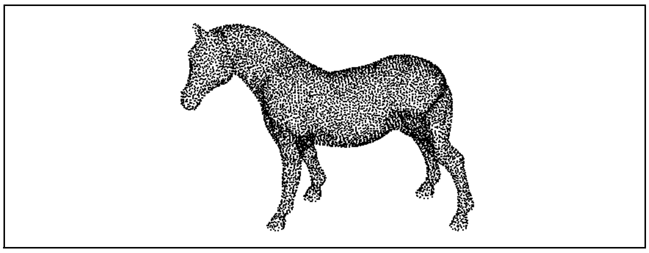


図 7. 点群モデル horse2(点の数 8485).

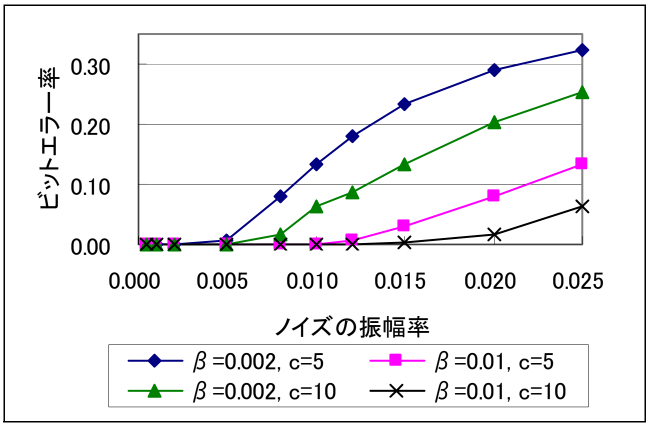


図 8. ランダムノイズを加えた際のノイズの振幅率とビットエラー率.

4. まとめと今後の課題

本論文では, 3 次元点群モデルを対象とした電子透かしについて述べた. 本手法の特徴としては, 点群からメッシュを生成し, 3 次元メッシュを対象としたメッシュのスペクトル分解による透かし手法[向山 02]を利用したこと, その際生成するメッシュは 2-多様体メッシュではないこと, 等があげられる. 実験の結果, 相似変換, ランダムノイズ付加, 切り取り, 擬似的な簡単化などの攻撃対して耐性を示し, 点群モデルの電子透かしとして有効な手法であることがわかった.

今後の改善点としては, スペクトル分解の際のメッシュ生成法の改善があげられる. 本手法で用いた簡便なメッシュ生成法でも, 各種の攻撃に対し一定の耐性を示すなど, 透かしとしての有効であった. しかし, 耐性や計算処理時間などの点でより良いメッシュ生成法があるか検討する必要がある.

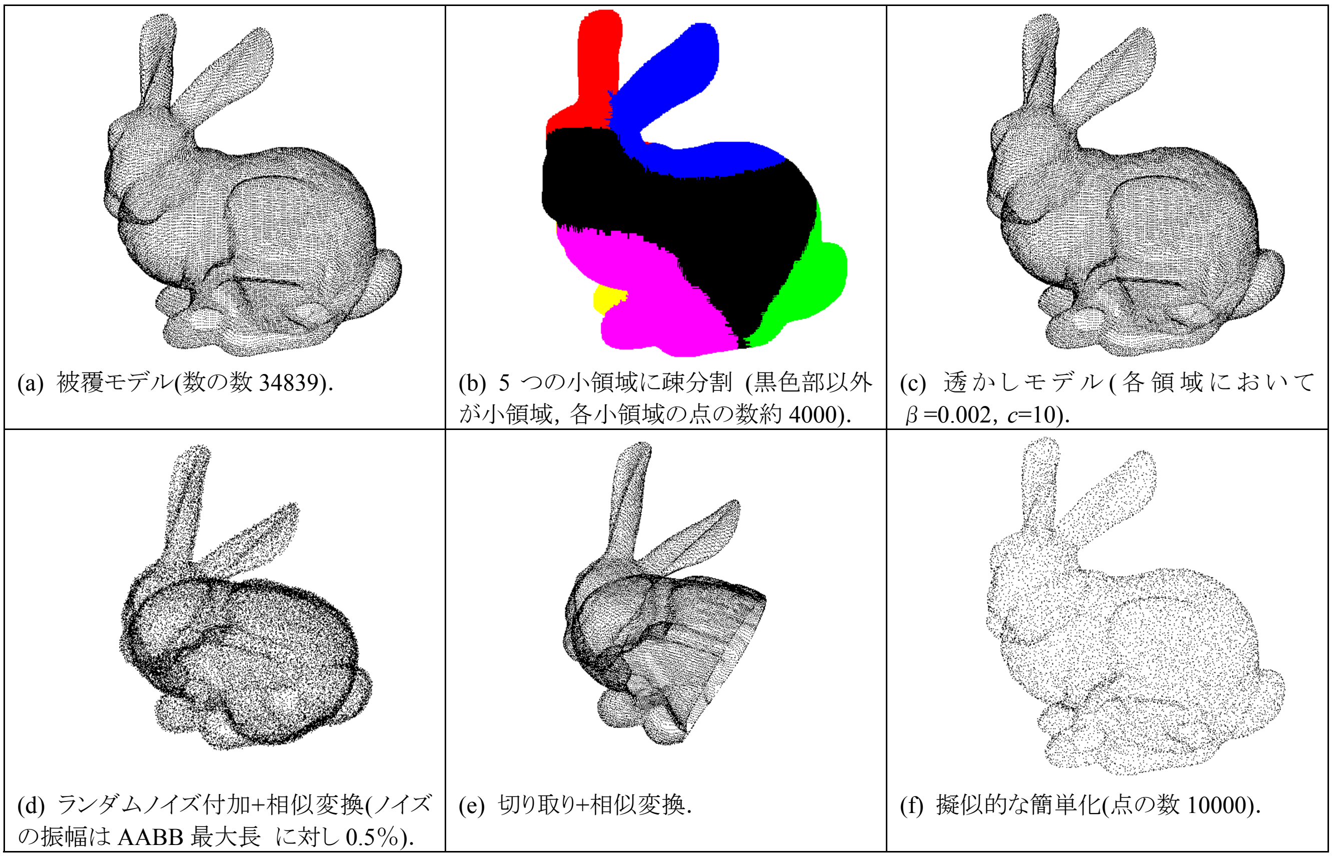
VisualComputing/グラフィックスと CAD 合同 ワークショップ 2003 採録
（情報処理学会・画像電子学会共催）, pp. 195-200, 2003/06/19~20, 小倉.

また，点群モデルの位置合わせ手法の改良が重要である．現在の透かし手法では，点のランダム削除と相似変換を組み合わせた攻撃に対しては耐性が低い．その理由は，点群モデルに対する位置合わせの精度が低く，ランダムに点を削除されると位置合わせが失敗するからである．そのため，点群モデルに対し，何らかの方法でより精度の高い位置合わせを行う必要がある．

謝辞：本研究の一部は文部科学省科学研究費補助金（12680432），大川情報通信基金，および人工知能研究振興財団の補助による．

参考文献

[Bollobás98] B. Bollobás, *Modern Graph Theory*, Springer, 1998.
[Carr01] J.C. Carr, R.K. Beaton, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum and T.R. Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. Proc. *SIGGRAPH 2001*, pp. 67-76, 2001.
[Cox02] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufman Publishers, 2002.
[Golub96] Golub, G. H., Van Loan, C. F., *Matrix Computations*, Third Edition, Johns Hopkins University Press, 1996.
[Gottschalk96] S. Gottschalk, M.C. Lin, D. Manocha, OBBTree: A Hierarchical Structure for Rapid Interference Detection, Proc. *SIGGRAPH '96*, pp. 171-180, 1996.
[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] Zachi Karni, Craig Gotsman, Spectral Compression of Mesh Geometry, Proceedings of the *SIGGRAPH 2000*, July 2000, New Orleans, U. S. A.
[MeshToSS01] 金井　崇, *MeshToss*, Version 1.0.1, http://graphics.sfc.keio.ac.jp/MeshToSS/indexE.html.
[Ohbuchi02] Ryutarou Ohbuchi, Akio Mukaiyama, Shigeo Takahashi, A Frequency-Domain Approach to Watermarking 3D Shapes, *Computer Graphics Forum* 21(3), pp. 373-382, (2002).
[Pauly02] M. Pauly, M. Gross, L. P. Kobbelt, Efficient Simplification of Point-Sampled Surces, *IEEE Visualization 2002*, 2002.
[Praun99] E. Praun, H. Hoppe and A. Finkelstein, Robust Mesh Watermarking, Proc. *SIGGRAPH '99*, pp. 49-56(1999).
[Yin 01] K. Yin, Z. Pan, J. Shi, Robust mesh watermarking bassed on multiresolution processing, *Computer and Graphics*, 25, pp. 409-420, 2001.
[松井98] 松井　甲子雄，電子透かしの基礎，森北出版 (1998)
[向山02] 向山　明夫，大渕　竜太郎，高橋　成雄，周波数領域で埋め込む3次元メッシュの電子透かしの改良，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2002，pp. 171-176，2002年6月．



(a) 被覆モデル(数の数 34839).

(b) 5 つの小領域に疎分割 (黒色部以外が小領域，各小領域の点の数約 4000).

(c) 透かしモデル(各領域において β=0.002, c=10).

(d) ランダムノイズ付加+相似変換(ノイズの振幅は AABB 最大長 に対し 0.5%).

(e) 切り取り+相似変換.

(f) 擬似的な簡単化(点の数 10000).

図 9. 点群モデル bunny2 を使用した実験結果の例.