

# ハンドル領域の回転とスケーリングを考慮したインタラクティブなメッシュ変形操作

Interactive Mesh Deformation by Translating, Rotating and Scaling Manipulation Handles

吉岡 康博<sup>†</sup>, 正会員 増田 宏<sup>†</sup>, 古川 慈之<sup>††</sup>

Yasuhiro Yoshioka<sup>†</sup>, Hiroshi Masuda<sup>†</sup> and Yoshiyuki Furukawa<sup>††</sup>

**Abstract** Recently, many researchers intensively study interactive mesh deformation techniques that preserve discrete mean curvature vectors. One of major obstacles in interactive mesh deformation is to develop the rotation method of mean curvature vectors. This paper proposes a novel framework in which the user can control the deformation of mesh models using the translation, rotation and scaling of manipulation handles. In addition, we introduce vector constraints for deforming mesh models while preserving specific regions as the same shapes. We propose a new rotation method based on the logarithms of unit quaternions, which can be manipulated in  $\mathbb{R}^3$ . This method allows to produce natural deformed models efficiently. We also show that our framework can be used for interactively controlling the distribution of mean curvature vectors.

キーワード：メッシュ，デフォーメーション，四元数，回転，離散平均曲率

## 1. ま え が き

メッシュ形状のインタラクティブな変形手法は近年活発に研究されている。これらの研究は元の曲面の特徴を保持しながらリアルタイムに形状変形をすることを目的としている。典型的な変形操作では、まずユーザが固定領域とハンドル領域を選択する。ハンドル領域とは、曲面を変形するためにユーザが操作する対象のことである。ユーザがスクリーン上でマウスやスライダを用いてハンドル領域を移動させると、それに追従してメッシュモデル全体が変形する。

このような変形を実現する手段として、平均曲率ベクトルを頂点座標を変数とする線形式で近似し、その値を保存しながらリアルタイムに形状変形する手法<sup>1)-3)</sup>が提案されている。平均曲率ベクトルとは、大きさが平均曲率であり、法線方向の向きを持つベクトルのことである。

しかしながら、平均曲率ベクトルを単に保存するだけでは、ハンドルの回転に追従した変形がうまく行えないことが知られている。ベクトルの回転は、XYZ座標系では三角関数を用いて記述される非線形操作であるため、インタラ

クティブに操作するためには特別な工夫が必要となる。

Sorkine ら<sup>1)</sup> は回転が微小であると仮定して  $\sin\theta \approx \theta$  と近似し、回転と座標を単一の線形システムで解いた。しかしこの方法では、大きな回転角のときに形状が歪むという問題がある。Lipman ら<sup>4)</sup> はすべての頂点上に局所座標系を定義し、それらの相対位置が保存するように局所座標系の各軸を算出した。しかし我々の実験では、この方法は均一なメッシュでは良好な変形結果を出力するが、不均一なメッシュでは局所的な歪みが発生する。Zayer ら<sup>3)</sup> は  $[0, 1]$  区間における調和関数を重みとして用いて、単位四元数によって表現した回転量を補間した。この方法は、メッシュが不均一な場合でも、比較的良好な回転を行える。ただし、単位四元数の4個の成分を独立に補間しているため、球面線形補間にはならず、補間後の四元数を正規化することが必要である。また、一つの調和関数では、ハンドルを単一の剛体としてしか扱えないため、ハンドル領域の形状を変形させたい場合には、ハンドル領域の頂点ごとに調和関数を求めてブレンドする必要があり、インタラクティブな形状変形は困難である。

本研究においては、ユーザがハンドルを移動、回転、スケーリングするのに追従して、メッシュモデルが変形する新しいフレームワークを提案する。また、特定の頂点間の相対位置を保持しながらメッシュ変形する手法も提案する。

以下、2章においてメッシュ変形フレームワークについて示し、3章では、ハンドルの回転に追従する変形手法、4

<sup>†</sup> 東京大学 大学院工学系研究科  
(〒 113-8656 東京都文京区本郷 7-3-1)

<sup>††</sup> 産業技術総合研究所  
(〒 305-8564 茨城県つくば市並木 1-2)

<sup>†</sup> School of Engineering, The University of Tokyo  
(7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656)

<sup>††</sup> National Institute of Advanced Industrial Science and Technology  
(1-2 Namiki, Tsukuba-shi, Ibaraki 305-8564)

章では、スケーリングによる変形手法について示す。5章で結論と展望を示す。

## 2. メッシュ変形フレームワーク

本章では、平均曲率ベクトルと位置制約を満たす変形を算出する線形システムについて説明し、本研究で提案するベクトル制約、回転、スケーリングを含めた定式化を行う。

### 2.1 離散平均曲率と位置制約の保存

$n$  頂点の三角形メッシュモデルにおいて、頂点  $i$  の3次元座標を  $\mathbf{p}_i = (x_i, y_i, z_i)$ 、頂点  $i$  とエッジで連結した隣接頂点の集合を  $N(i)$  と置く。また、変形前の初期メッシュの頂点座標を  $\mathbf{p}_i^0$  と書く。また、すべての頂点の集合を  $\mathbb{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ 、 $x$  座標、 $y$  座標、 $z$  座標の集合をそれぞれ  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 、 $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ 、 $\mathbf{z} = \{z_1, z_2, \dots, z_n\}$  とする。

各頂点における曲面の平均曲率  $\kappa_i$  と法線ベクトル  $\mathbf{n}_i$  の積は、頂点座標の線形和として以下のように近似できることが知られている<sup>5)</sup>。

$$\kappa_i \mathbf{n}_i = L(\mathbf{p}_i) = \sum_{j \in N(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{4\text{Area}(i)} (\mathbf{p}_i - \mathbf{p}_j) \quad (1)$$

ここで、 $\text{Area}(i)$  は頂点  $i$  を最近傍頂点とする領域の面積で、ポロノイ面積と呼ばれる。また、 $\alpha_{ij}$  と  $\beta_{ij}$  は稜線  $(i, j)$  を底辺とする二つの三角形の頂点の角度である。

変形の前で平均曲率が保存する条件は、初期状態の頂点  $i$  の平均曲率ベクトルを  $\delta_i$  としたとき、 $L(\mathbf{p}_i) = \delta_i$  と書ける。しかし、 $L(\mathbf{p}_i)$  を初期状態  $\delta_i$  に固定すると、その頂点における法線  $\mathbf{n}_i$  も固定されてしまうことになる。そのため、ハンドルの変形や回転に応じて平均曲率ベクトル  $\delta_i$  を回転させる必要がある。また、スケーリングを行うのであれば、平均曲率ベクトルのノルムの制御が必要である。

図1は、ハンドルが回転したときの平均曲率ベクトル  $\delta_i$  の向きを示している。図1(a)は変形前のメッシュ、図1(b)は  $\delta_i$  の向きを元のまま保存した例で、形状が歪んでしまうことがわかる。直観に合った変形結果を得るためには、図1(c)のように、ハンドルの回転に応じて  $\delta_i$  を適切に回転させることが必要である。

本研究では、回転とスケーリングは、座標計算の前に算出しておくこととし、ベクトルを回転する演算子  $R_i$  とスカラー量のスケールファクタ  $s_i$  を平均曲率ベクトルに作用させて、以下のように表現する。

$$L(\mathbf{p}_i) = s_i R_i(\delta_i) \quad (i = 1, 2, \dots, n) \quad (2)$$

$R_i$  と  $s_i$  は、ハンドル操作に応じてリアルタイムに計算される。これらの算出方法については次章以降で述べる。

また、位置制約として、固定頂点とハンドル頂点を以下のように指定する。

$$\begin{cases} \mathbf{p}_j = \mathbf{p}_j^0 & (j \in \Lambda_f) \\ \mathbf{p}_k = \mathbf{u}_k & (k \in \Lambda_h) \end{cases} \quad (3)$$

ここで、 $\Lambda_f$  と  $\Lambda_h$  はそれぞれ固定頂点とハンドル頂点の添字集合、 $\mathbf{p}_j^0$  は初期メッシュの頂点、 $\mathbf{u}_k$  はユーザがインタラクティブに移動させる頂点座標である。

なお、メッシュ中の頂点と同様にして、稜線上および面上の任意の点に関する位置制約についても次のように記述可能である。

$$\begin{cases} t\mathbf{p}_i + (1-t)\mathbf{p}_j = \mathbf{u}_{ij} \\ s\mathbf{p}_i + t\mathbf{p}_j + (1-s-t)\mathbf{p}_k = \mathbf{u}_{ijk} \end{cases} \quad (4)$$

ここで、 $\mathbf{u}_{ij}, \mathbf{u}_{ijk}$  は定数となるベクトル、 $t, s$  は  $0 < t < 1$ 、 $0 < s < 1$ 、 $0 < s+t < 1$  を満たす定数である。これらも頂点座標に関する線形方程式なので、式(3)と同様に扱うことができる。

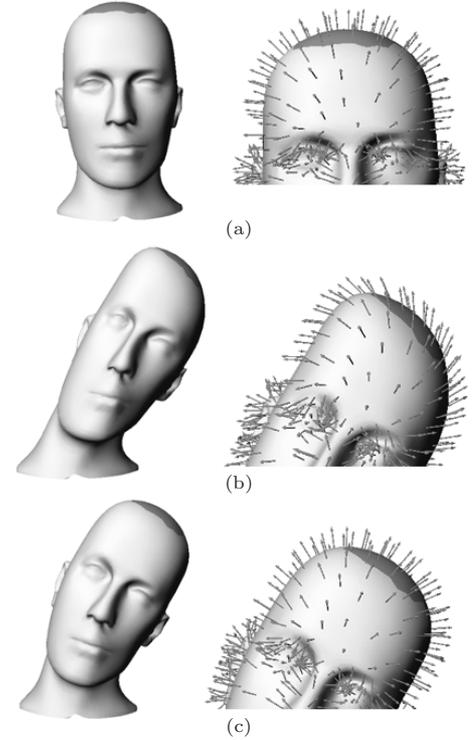


図1 メッシュ変形における法線ベクトルの回転の有無  
Rotation of normal vectors during deformation.

### 2.2 ベクトル制約

従来研究<sup>1)-3)</sup>では、位置制約と平均曲率ベクトルのみを用いて座標計算を行っていたが、本研究では、これらに加えてベクトル制約を導入する。ベクトル制約とは、メッシュ上の任意の2点の相対位置を固定する制約である。頂点間の相対位置を固定する場合には以下のように記述される。

$$\mathbf{p}_i - \mathbf{p}_j = s_{ij} R_{ij}(\mathbf{v}_{ij}) \quad ((i, j) \in E_f) \quad (5)$$

ここで、 $\mathbf{v}_{ij} \equiv \mathbf{p}_i^0 - \mathbf{p}_j^0$  と定義し、 $E_f$  はベクトル制約が与えられた頂点ペアの集合とする。また、 $s_{ij}$  はスケール、 $R_{ij}$  は回転の演算子で、式(2)と同様の理由で、 $\mathbf{v}_{ij}$  を回転・スケーリングするために導入される。ここでは2頂点

の相対位置に関するベクトル制約を示したが、式 (4) に示した式を用いて、メッシュ上の任意の2点間にベクトル制約を設定することも可能である。

ベクトル制約を用いることで、メッシュの特定領域の形状や特定の点間距離を保持するようにメッシュ変形を行うことができる。図2は頂点間の相対位置を保持した変形の例で、変形後も矢印で示した口と鼻の距離が保たれている。

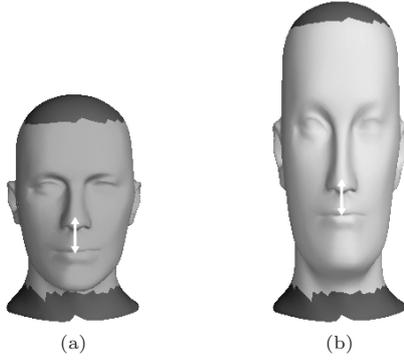


図2 ベクトル制約を含むモデルの変形  
Deformation of a mesh model with vector constraints.

### 2.3 数値計算

式 (2) (3) (4) (5) に示す制約式の総数は頂点数よりも多いため、厳密解は存在しない。そのため、頂点座標の計算では最小二乗法がよく用いられている<sup>1)</sup>。ここで、式 (3) (4) の制約式の個数を  $m$ 、制約式に含まれる頂点の集合を  $\mathbb{P}_j \subset \mathbb{P}$  ( $j = 1, 2, \dots, m$ ) と置き、各制約式を一次関数  $f_j$  を用いて以下のように記述するものとする。

$$f_j(\mathbb{P}_j) = \mathbf{u}_j \quad (j = 1, 2, \dots, m) \quad (6)$$

このとき、頂点座標は、以下のような最小二乗法で解くことができる。 $n$  はメッシュモデルの頂点数である。

$$\min_{\mathbf{p}} \left( \sum_{i=1}^n \|L(\mathbf{p}_i) - s_i R_i(\delta_i)\|^2 + \sum_{j=1}^m \|f_j(\mathbb{P}_j) - \mathbf{u}_j\|^2 \right) + \sum_{(i,j) \in E_f} \|\mathbf{p}_i - \mathbf{p}_j - s_{ij} R_{ij}(\mathbf{v}_{ij})\|^2 \quad (7)$$

最小二乗法の解は、連立一次方程式の解法問題に帰着させることができる。ここでは、式 (2) (3) (4) (5) はいずれも線形式であり、また  $x_i, y_i, z_i$  に依存関係はないので、 $\mathbf{x}, \mathbf{y}, \mathbf{z}$  を同じ行列  $A$  を用いて下記のように別々の線形システムとして記述することができる。ここで、 $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$  は定数ベクトルである。

$$A\mathbf{x} = \mathbf{c}_x, \quad A\mathbf{y} = \mathbf{c}_y, \quad A\mathbf{z} = \mathbf{c}_z \quad (8)$$

このとき、最小二乗法の解を算出する線形システムは以下のように書くことができる。

$$A^t A \mathbf{x} = A^t \mathbf{c}_x, \quad A^t A \mathbf{y} = A^t \mathbf{c}_y, \quad A^t A \mathbf{z} = A^t \mathbf{c}_z \quad (9)$$

ユーザがハンドル頂点の座標を変化させると、式 (9) の定数ベクトル  $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$  の値が変わるので、それに応じてメッシュ頂点を再計算する。

線形システム (9) を高速に解くために、正方行列  $A^t A$  を因子分解しておく<sup>6)7)</sup>。行列  $A^t A$  は正定値対称行列なので、コレスキー分解を用いて、下三角行列  $L$  の積  $L^t L$  に分解できる。元の行列が疎行列であるとき、列を適切に入れ換えれば、行列  $L$  も疎行列になることが知られている<sup>8)</sup>。一旦この形に分解しておけば、 $\mathbf{x}$  は、 $L^t \mathbf{w} = A^t \mathbf{c}_x$  と  $L\mathbf{x} = \mathbf{w}$  を解くことで十分インタラクティブな速度で座標計算が行える。 $\mathbf{y}, \mathbf{z}$  についても同様である。

コレスキー分解については、高速なソルバが開発されている。本研究では、ソルバとして TAUCS<sup>9)</sup> を用いて実装を行った。CPU: 2.0 GHz Pentium M, メモリ: 1.0 GB, Windows OS のノート PC を用いて計測を行った結果、1.9 万頂点のメッシュで 1.7 秒、3.5 万頂点のメッシュで 4.0 秒の CPU 時間で計算が行えた。これはメッシュ変形の前処理の計算時間として十分なパフォーマンスといえる。

### 3. ハンドルの回転による形状変形

ハンドル頂点が回転するとき、平均曲率ベクトルもそれに追従させて回転させることが必要である。回転させる  $\delta_i$  の座標値は式 (9) の右辺に含まれるため、行列の因子分解には影響を与えない。そのため、回転を作用させたメッシュ変形もインタラクティブに行うことが可能である。

回転の補間では、線形補間よりも球面線形補間の方が滑らかな補間ができることが知られている。Shoemake<sup>9)</sup> は、アニメーションでの動作の補間に関して、単位四元数を用いて二つの回転を球面線形補間する手法を示した。Johnson<sup>10)</sup> は、キャラクターモーションにおける回転角を単位四元数の対数を用いて記述し、3個以上の回転を球面線形補間する方法を示した。ただし、いずれの方法もインタラクティブな変形操作を目的としたものではない。

Zayer ら<sup>3)</sup> は単位四元数を調和関数によって線形補間したが、球面線形補間ではなく、また、複数の回転を補間する際は、個々の回転に応じて調和関数を算出して合成する必要がある。そのため、図4(b)のようにハンドル領域である頭部の形状をインタラクティブに変形することは困難である。この例では、ハンドル領域での各頂点の回転量が異なるために多数の調和関数を合成することが必要となる。

本研究では、Johnson が提案した単位四元数の対数を用いた回転補間方法をインタラクティブなメッシュ変形に応用することで、平均曲率ベクトルの回転量を計算する方法を示す。また、単位四元数の対数が張る空間におけるエネルギー最小曲面を求めることで、複数の回転を補間する際も調和関数やその合成を用いることなく直接的に補間を行う。四元数は実数  $(w, x, y, z)$  を用いて、以下のように書ける。

$$Q = (w, x, y, z) = w + xi + yj + zk \quad (10)$$

ここで,  $i, j, k$  は四元数単位である．絶対値が 1 である四元数は単位四元数と呼ばれ, 回転軸  $\hat{n}$  と回転角  $\theta$  を用いて以下のように書ける． $\hat{n}$  は, 純四元数である．

$$\hat{Q} = \exp\left(\frac{\theta}{2}\hat{n}\right) = \cos\frac{\theta}{2} + \hat{n}\sin\frac{\theta}{2} \quad (11)$$

このとき, 単位四元数の対数  $\mathbf{q} \in \mathbb{R}^3$  は, 以下のように定義される．

$$\mathbf{q} = \ln \hat{Q} = \frac{\theta}{2}\hat{n}. \quad (12)$$

本研究では, 頂点  $i$  に対して, 単位四元数の対数  $\mathbf{q}_i$  を割り当てる．固定頂点とハンドル頂点においては, 以下のように回転量が指定される．

$$\begin{cases} \mathbf{q}_j = 0 & (j \in \Lambda_f) \\ \mathbf{q}_k = \mathbf{r}_k & (k \in \Lambda_h) \end{cases} \quad (13)$$

ここで,  $\mathbf{r}_k$  は, ユーザが指定するハンドルの回転量である．

ベクトル制約がエッジに与えられた場合には, 両端点の回転量が等しくなるように以下の式を追加する．

$$\mathbf{q}_i - \mathbf{q}_j = 0 \quad ((i, j) \in E_f) \quad (14)$$

この式により, 式 (5) を回転する演算子は,  $R_{ij} = R_i (= R_j)$  として算出できる．

次に, 単位四元数の対数が張る 3 次元空間  $\mathbb{R}^3$  について考える．本研究では, 回転量が 3 次元ベクトルとして記述されているため, 滑らかに補間された回転量は,  $\mathbb{R}^3$  において固定頂点とハンドル頂点を通る滑らかな曲面を構成すると考えられる．そこで, 本研究では, 式 (13) (14) を境界条件とするエネルギー最小曲面について考える．

本研究では, 平均曲率の二乗和をエネルギー関数としたときのエネルギー最小曲面を算出する．そのために, 次式のような制約を導入する．

$$L(\mathbf{q}_i) = \sum_{j \in N(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{4\text{Area}(i)} (\mathbf{q}_i - \mathbf{q}_j) = 0 \quad (15)$$

式 (13) (14) (15) は, 座標の計算と同様にして, 連立一次方程式  $A\mathbf{q} = \mathbf{r}$  を構成する．行列  $A$  は座標計算の式 (9) で用いた行列と同じであるので, 最小二乗法の計算に現われる行列  $A^t A$  の因子分解は一度だけ行っておけばよい．そのため, 前処理に要する計算時間は, 回転を行わない場合とほぼ同一となる．

各頂点における回転量  $q_i$  が算出されると, 回転軸が  $\mathbf{q}_i/|\mathbf{q}_i|$ , 回転角が  $2|\mathbf{q}_i|$  と算出され  $R_i$  が一意に決まるので, この演算子を用いて式 (2) の  $\delta_i$  を回転させる．

図 1(c) は, 本手法によって, 平均曲率ベクトルを回転させて座標計算を行った例である．このモデルは 2732 頂点で, 行列の因子分解に 0.15 秒要した．一方, 因子分解した行列を用いて, 回転と座標の算出を行う計算は一回あたり平均 0.05 秒で行えた．この計算速度は, インタラクティブなメッシュ変形には十分であるといえる．

図 3 は, 凹凸パターンのある曲面の例である．法線の方

向が初期状態に固定されると, 図 3(b) のように, 曲面上の凹凸は元の向きを保とうとする．本手法で示したように平均曲率ベクトルを回転することで, 図 3(c) のように曲面上のパターンをハンドルの回転に追従させることができる．

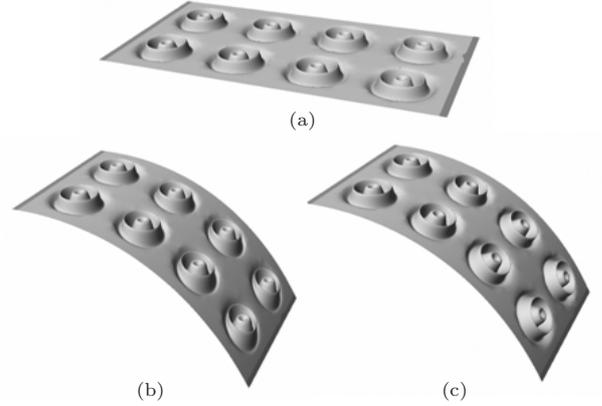


図 3 曲面上のパターンの回転  
Rotation of patterns on a surface.

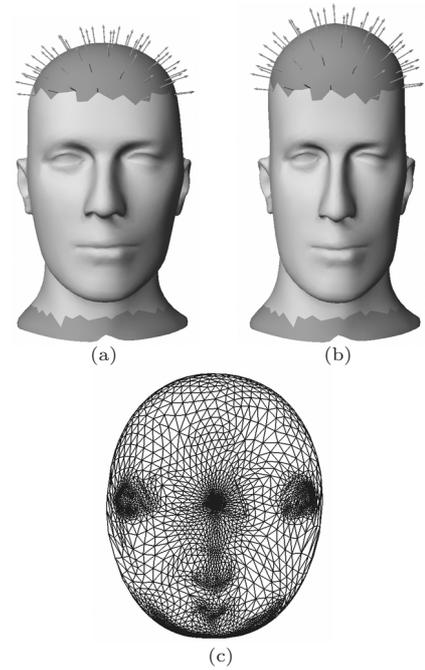


図 4 回転量を補間するエネルギー最小曲面  
Energy-minimization surface that interpolates rotations.

図 4 は回転量の補間で用いたエネルギー最小曲面を可視化したものである．図 4(a) は元の曲面で, 首の部分を固定し, 頭部にハンドルを設定している．図 4(b) は, ユーザが頭部が尖るようにハンドル頂点を移動した場合の変形を示している．図 4(c) は, 単位四元数の対数によって張られる空間  $\mathbb{R}^3$  におけるエネルギー最小曲面で, 固定点は中心付近の原点, ハンドル頂点の回転量は輪郭部分に相当する．この図に示すように, 式 (15) を用いて算出されたエネ

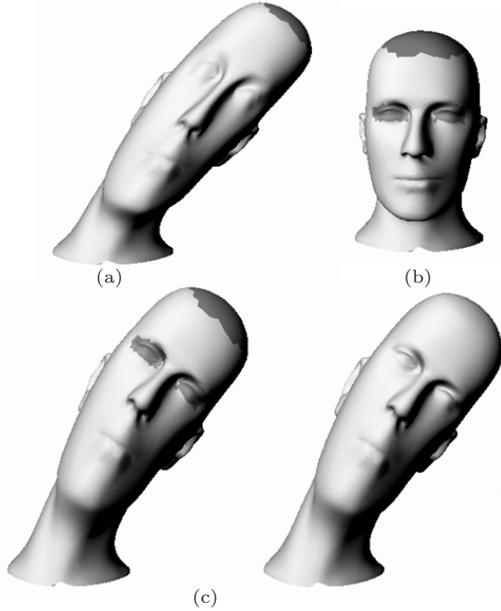


図 5 ベクトル制約を含むモデルの変形  
Deformation of a mesh model with vector constraints.

ルギー最小曲面では，元のメッシュの疎密を反映した回転量が計算される．この理由は，式 (15) が，曲面を 2 次元の uv 空間に等角写像し，uv 空間でラプラシアンを算出したものに相当しているためである．直観的には，式 (15) の分母となる面積が三角形の大小を反映させ，cot の項が正三角形からのずれを反映させている（数学的に厳密な議論は，文献<sup>5)</sup>を参照のこと）．

図 5 は，ベクトル制約を与えた領域がハンドルに追従して回転した例である．ベクトル制約を与えていない図 5(a) では，目の部分が引き延ばされて印象の異なった顔になる．一方，図 5(b) に示すようにベクトル制約を与えて変形させると，図 5(c) に示すように目の領域の形状を保持したまま変形できる．このように，モデルの重要な部分をベクトル制約で固定することで，元のモデルにより忠実な変形結果を得ることが可能となる．

#### 4. スケーリングによる形状変形

各頂点における平均曲率ベクトルの大きさを制御することによっても，形状変形が行える．ここでは，ハンドル部のスケーリングに追従させて全体の形状を変形する方法と，メッシュ全体に対してスケーリングを行うことで，強調や平滑化を行う手法について示す．

##### 4.1 ハンドルのスケーリングによる形状変形

ハンドル領域を拡大させると，曲率半径が大きくなり，平均曲率は小さくなる．このとき，他の頂点の平均曲率ベクトルのスケールの調整を行わないと，図 6(b) に示すように，元の形状の特徴を損なった不自然な形状が生成される．

本研究では，ハンドル領域のスケーリングに追従した変形を平均曲率のスケールを補間することで実現する．スケー

ルの補間に関しても，頂点や回転の計算と同様に，以下の式を制約式として用いる．

$$L(s_i) = \sum_{j \in N(i)} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{4\text{Area}(i)} (s_i - s_j) = 0 \quad (16)$$

ここで， $s_i \in \mathbb{R}$  は頂点  $i$  での平均曲率のスケールである．

また，固定頂点，ハンドル頂点，ベクトル制約は，回転の場合と同様に，以下のように書くことができる．

$$\begin{cases} s_j = 1 & (j \in \Lambda_f) \\ s_k = t_k & (k \in \Lambda_h) \\ s_i - s_j = 0 & ((i, j) \in E_f) \end{cases} \quad (17)$$

ここで， $t_k$  は，ハンドル頂点に対してユーザが指定したスケールである．

式 (16) (17) から構成される連立一次方程式も，座標と回転の計算で用いる行列  $A$  によって記述できるので，高速な計算が可能である．

図 6 に拡大量を補間した変形の例を示す．図 6(b) は，平均曲率の大きさを固定した場合で，ハンドル領域の拡大・縮小に追従していない．図 6(c) では，ハンドル領域の拡大に追従させて，平均曲率ベクトルのスケールを調整した例である．この例のように，スケールの補間を行った方がより自然な変形結果が得られる．

##### 4.2 全体のスケーリングによる形状変形

本研究のフレームワークでは，メッシュ全体に関して，同一のスケール値を与え，その値をユーザが操作することによっても形状変形が行える．

図 7(a) は凹凸パターンのあるメッシュモデルである．このモデルに対して，図 7(b)-(e) に示すようにスケールを操作することで，曲面上のパターンを制御することができる．スケール値の変更は，スライダ等を用いてリアルタイムに行える．また，本手法を用いれば，グラフィックイコライザのように，特定の範囲の平均曲率の値をスケーリング操作で編集することによって形状を操作することが可能である．図 7(f) は条件式を用いて制御を行った例で，平均曲率が 0.1 以下のときに平均曲率を 0 倍，それ以外で平均曲率を 2 倍にしている．

図 8 は顔のモデリングに応用した例である．図 8(a) は上下各 10% ずつの領域を固定し，変形領域はフィルタでスケールを単純に 1.5 倍した．同様に図 8(b) は上 10%，下 25% の領域を固定し，変形領域はフィルタでスケールを単純に 2 倍した．このように容易にキャラクタのパリエーションを作成することも可能である．

## 5. む す び

本研究では，ハンドル操作に追従させたインタラクティブなメッシュ変形操作を実現するために，ベクトル制約を導入し，またハンドル操作に応じて回転・スケーリングを行う手法を示した．回転は，単位四元数の対数を用いた手

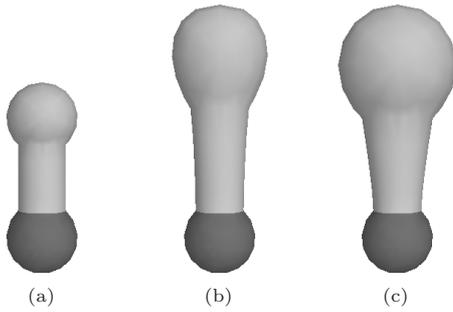


図 6 スケールを補間した変形の例  
An example of scale interpolation.

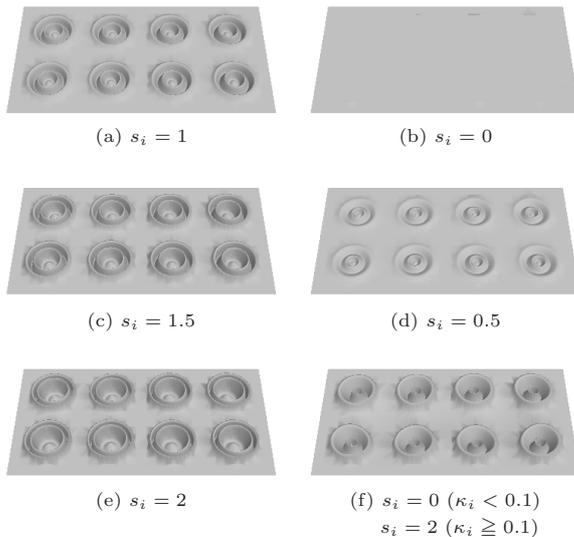


図 7 パターンの平滑化と強調  
Smoothing and enhancement of patterns on a surface.

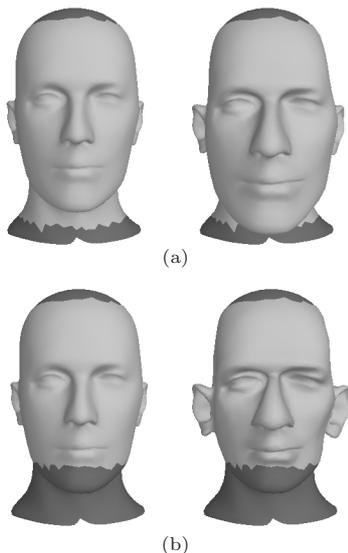


図 8 スケーリングの操作による形状変形  
Shape deformation controlled by scaling factors.

法を提案し、良好な変形結果が得られることを示した。また、スケーリング操作によって、多様な形状変形操作が実現することも示した。本手法は、大規模な連立一次方程式を解くことが必要であるが、十分高速に計算が行えること

を示した。

本手法は、実装が容易であるため、コンテンツ作成支援システムに容易に組み込むことができる。本手法を用いたモデリングにおいて、現在の実装では、ハンドル領域と固定領域を画面上で指定した後、マウスを用いてハンドルの移動と回転を指定している。また、ROI(region of interest)に対する拡大/縮小、強調/スムージングは、それぞれ独立したスライダによって操作している。これらは一度、行列の分解を行っておけば、いずれもインタラクティブに行うことができる。今後は、非常に大規模なメッシュモデルへの対応や、ソルバの高速化、インタラクティブな操作に適したユーザインタフェースに関して検討していきたいと考えている。

### 〔文 献〕

- 1) O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rössl, and H. Seidel : "Laplacian surface editing," Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 175-184 (2004).
- 2) M. Botsch, and L. Kobbelt : "An intuitive framework for real-time freeform modeling," ACM Trans. Graphics, 23(3), pp. 630-634 (2004).
- 3) R. Zayer, C. Rössl, Z. Karni and H. Seidel : "Harmonic Guidance for Surface Deformation", Computer Graphics Forum, Proceedings of Eurographics 2005, 24(3), pp.601-609 (2005).
- 4) Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or : "Linear rotationinvariant coordinates for meshes," ACM Trans. Graphics, 24(3), pp.479-487 (2005).
- 5) M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr : "Discrete differential-geometry operators for triangulated 2-manifolds," Visualization and Mathematics III, pp.35-57 (2003).
- 6) O. Sorkine. "Laplacian mesh processing," Proceedings of Eurographics 2005, pp.53-70 (2005).
- 7) M. Botsch, D. Bommers, and L. Kobbelt : "Efficient linear system solvers for mesh processing," Proceedings of IMA Conference on the Mathematics of Surfaces 2005, pp. 62-83 (2005).
- 8) S. Toledo, D. Chen, and V. Rotkin : "Taucs: A library of sparse linear solvers," <http://www.tau.ac.il/stoledo/taucs/> (2003).
- 9) K. Shoemake : "Animating rotation with quaternion curves," Proceedings of SIGGRAPH 1985, pp. 245-254 (1985).
- 10) M. P. Johnson : "Exploiting quaternions to support expressive interactive character motion," PhD thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences (2003).

よしおか やすひろ  
**吉岡 康博** 2004年、東京大学工学部システム創成学科知能社会システムコース卒業。2006年、同大学院工学系研究科環境海洋工学専攻修士課程修了。メッシュモデリング、コンピュータグラフィックスに関する研究に従事。

ますだ ひろし  
**増田 宏** 1985年、東京大学工学部精密機械工学科卒業。1987年、同大学院工学系研究科修士課程修了。同年日本アイ・ピー・エム(株)東京基礎研究所。1998年、東京大学人工物工学研究センター助教授、2003年より東京大学大学院工学系研究科助教授。形状モデリング、コンピュータグラフィックス、CADに関する研究に従事。工学博士。正会員。

ふるかわ よしゆき  
**古川 慈之** 1998年、東京大学工学部船舶海洋工学科卒業。2003年、同大学院工学系研究科環境海洋工学専攻博士課程修了。同年(独)産業技術総合研究所ものづくり先端技術研究センターに入所、現在に至る。三次元形状処理、設計・製造支援アプリケーション用ソフトウェアプラットフォームに関する研究に従事。博士(工学)。