

入力された形状を考慮したスケッチによるメッシュの表面特徴の変形

清田 智文[†] 今井 桂子^{††}

概要

本研究では、3次元表面メッシュデータに対する変形アルゴリズムを提案する。3次元コンピュータグラフィックスのモデリングやアニメーション作成において、メッシュの形状を対話的に変形させる技術が用いられている。近年、入力方法として、より直感的に変形を指示することが可能であるスケッチを用いる変形手法が注目されている。その一つに Y.Kho らの手法がある。その手法はメッシュのポーズの変形に主眼を置いていたため、メッシュの表面特徴を変形することができない。そこで、本研究ではその手法をもとに、スケッチによって入力された2本の曲線を用いてメッシュの表面特徴を対話的に変形する手法を提案し、実装する。

Mesh Deformations for Surface Features Using Sketches in Consideration of Initial Shapes

TOMOFUMI KIYOTA[†] and KEIKO IMAI^{††}

Abstract

In this paper, we propose a new deformation algorithm for 3D surface meshes by sketching curves in the image plane. Interactive mesh deformations are used for modeling and animation in computer graphics. Recently, sketch-based interfaces are a very popular method for editing and deformation of 3D surface meshes. Kho and Garland presented an interactive system for deforming unstructured polygon meshes, and it is one kind of sketch-based interfaces. Their system was very easy to use for non-specialists. However, it is hard to deform surface features by their algorithm, because their approach was targeted towards reposing object bodies and limbs. Therefore, we have improved the algorithm so that we can deform surface features. We have included some experimental results.

1. 序 論

メッシュとは与えられた領域を三角形や四角形などの図形で分割したものである。メッシュは3次元コンピュータグラフィックスや有限要素解析など様々な分野で応用されている。3次元コンピュータグラフィックスのモデリングやアニメーション作成において、メッシュを対話的に変形させる技術が用いられる。本研究では、三角形表面メッシュの変形について扱う。

様々なメッシュに対する変形手法 ([3], [5] 等) が提案されている。さらに近年、ユーザにとって自分の要求する変形を直感的に示すことのできるスケッチが注目されている。スケッチとは、ポインティングデバイスを用いてウィンドウ上に曲線を描く入力方法のことである。スケッチを用いる変形手法の代表的なものに、

入力された2本の曲線から変形を行なう Kho らの手法がある [2]。他にもスケッチは、ユーザにとって扱いやすい入力として、メッシュに凹凸を作り出す Nealen らの手法でも用いられている [4]。また、[2] では変形による頂点の移動を限定することによって、メッシュを短時間で変形する事を可能にした。

しかし、Kho らの手法は対象のポーズの変形に主眼をおいているため、表面特徴の変形を行なうことができない。表面特徴とは、目や口などの人間が輪郭を認識できるメッシュの特徴的な凹凸である。そこで、本研究では Kho らの手法を改良し、スケッチを基にした表面特徴の変形を行なうアルゴリズムを提案する。しかし、この方法をそのまま実行すると、結果が不自然に見えることがある。これは、変形によって表面特徴が不自然に歪むためである。そこで、メッシュの変形手法の一つである Sheffer らのアイデアを用いる [6]。それは、隣り合う頂点同士的位置関係をなるべく保持するように変形を行なう手法である。本研究では、その手法を応用して入力されたメッシュの形状を保持することによって、変形結果を修正する。

[†] 中央大学大学院 理工学研究科 情報工学専攻
Information and System Engineering Course, Graduate
School of Science and Engineering, CHUO University

^{††} 中央大学 理工学部 情報工学科
Department of Information and System Engineering,
Faculty of Science and Engineering, CHUO University

本論文の構成は以下の通りである。まず 2 節において、提案するアルゴリズムを順を追って述べる。3 節では、それを実装し提案手法の有効性を示す。最後に 4 節で結論と今後の課題について記す。

2. スケッチを基にしたメッシュの表面特徴の変形

本節では、提案するアルゴリズムについて述べる。与えられた 3 次元の曲面メッシュに対して、入力としてユーザが 2 本の曲線をスケッチする。1 本目の曲線はユーザの変形したい表面特徴を示す曲線である。この曲線を指示曲線と呼ぶ。2 本目の曲線は指示曲線で示した表面特徴の変形後の形状を表す。この曲線を目的曲線と呼ぶ。図 1 に指示曲線 (実線) と目的曲線 (点線) の入力例を示す。メッシュで表現された耳を拡大しようとする場合、まず耳の淵に沿って指示曲線を入力する。そして、要求する変形後の耳の淵の形状を目的曲線としてスケッチする。

2 本の曲線がスケッチされた後のアルゴリズムの概略について述べる。まず、指示曲線、目的曲線の位置関係を明らかにするために、これらの曲線上の点同士を対応付ける。そして、指示曲線と目的曲線の位置関係から変形によって移動する頂点を決める。さらに、移動する頂点を、指示曲線上の点と対応付ける。その後、対応付けた指示曲線上の点をそれに対応付けられた目的曲線上の点付近に移動する。最後に、入力されたメッシュの形状を考慮してメッシュの修正を行なう。これによって、表面特徴が変形されたメッシュが出力される。以下では、提案するアルゴリズムについて順を追って詳細を述べる。

2.1 形状の保存

Sheffer らの変形アルゴリズムを応用して、入力されたメッシュの形状を保存する方法を考える。形状の保存はグローバルな座標ではなく、各頂点に対してそれに隣接する頂点との位置関係を保持することによって行う。本節では、その方法について述べる。頂点 v に対して、座標を $\mathbf{v}, v_1, v_2, \dots, v_m$ をそれに隣接する頂点を反時計回りに表わしたものとす。頂点 v における法線ベクトル $\mathbf{n} = (n_x, n_y, n_z)$ と $d = -\sum_{i=1}^m \mathbf{n} \cdot \mathbf{v}_i$ を用いて射影平面を

$$P: n_x x + n_y y + n_z z + d = 0 \quad (1)$$

と定義する。頂点 v の法線ベクトル \mathbf{n} は、 v と、 v_1, v_2, \dots, v_m を用いて構成されるすべての面の法線ベクトルの平均とする。 v と v_i を P へ射影したものを、それぞれ v^P, v_i^P と定義する。射影された頂点 v^P, v_i^P を結ぶ辺を $\langle v^P, v_i^P \rangle$ とし、 $\langle v^P, v_i^P \rangle$ と

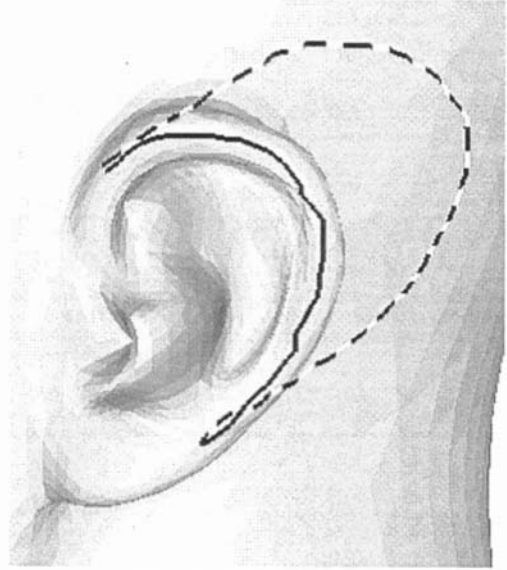


図 1 指示曲線, 目的曲線.

$\langle v^P, v_{i+1}^P \rangle$ のなすの角度を α_i , \mathbf{n} と $\langle v^P, v_i^P \rangle$ のなすの角度を β_i , $\langle v^P, v_i^P \rangle$ のユークリッド距離を l_i とする (図 2)。これらを与えられたメッシュのすべての頂点に対して求める。

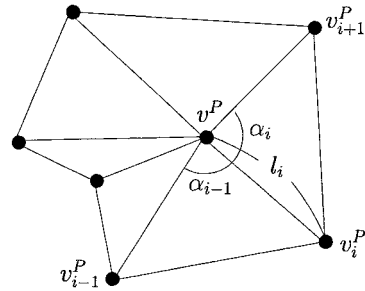


図 2 射影平面 P

次に、 v_i とこれらの値を用いて v を表わす方法について述べる。はじめに、平均座標の復元性質 [1] を用いて、射影平面 P 上において、 v_i^P から v^P を導く (図 2)。具体的には、

$$\sum_{i=1}^m \lambda_i v_i^P = v^P$$

$$\sum_{i=1}^m \lambda_i = 1$$

となる重み $\lambda_1, \dots, \lambda_m \geq 0$ を求める。もし、 α_i が $0 < \alpha_i < \pi$ となるなら、 v^P に対する v_1^P, \dots, v_m^P の重

みは,

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{l_i}$$

を用いると,

$$\lambda_i = \frac{w_i}{\sum_{j=1}^m w_j}$$

となる.

次に, 与えられた v^P から v の位置を導く為に, \mathbf{n} に沿って補正值

$$h_i = \|\mathbf{v}^P - \mathbf{v}_i^P\| \cot(\beta_i) + (\mathbf{v}_i - \mathbf{v}_i^P) \cdot \mathbf{n} \quad (2)$$

を計算する. そして, \mathbf{n} に沿った射影平面からの距離 h_i の平均で v^P を補正することによって,

$$\mathbf{v} = \mathbf{v}^P + \left(\frac{1}{m} \sum_{i=1}^m h_i \right) \mathbf{n} \quad (3)$$

を得る. 変形後にこの式を用いて形状の修正を行う為に, 入力されたメッシュのすべての頂点に対して, λ_i と β_i の値を保持する. これによって, 隣接する頂点の座標から元の頂点の位置を求めることが可能となる.

2.2 変形平面

提案手法では, 頂点の移動を, ある平面と平行な移動のみに限定する. これによって, 短時間で変形を行なうことが可能になる. ここでは, その平面を視線に垂直な平面とする. この平面は, 入力されたメッシュの頂点の中で, 視点に一番近い頂点を通るものとする. この平面を変形平面と呼ぶ. また, 指示曲線と目的曲線はこの平面上にあるものとする (図3). 変形平面上に直交座標系を考え, 以後の頂点の座標に対する計算はすべて変形平面に射影した座標で行なうものとする.

2.3 指示曲線と目的曲線

はじめに, ユーザによって指示曲線と目的曲線がスケッチされる. これは, ユーザのポインティングデバイスの動きから得られる 2 次元平面上的折れ線である. この指示曲線と目的曲線は, 人間の手によって入力されるため, かなり荒く扱いづらい. そこで, 指示曲線, 目的曲線をより扱いやすくする必要がある. 本項ではスケッチによる入力から近似の折れ線を得る手法と 2 本の曲線の対応付けについて述べる.

スケッチによる指示曲線, 目的曲線の入力においては, ユーザのポインティングデバイスの動きによって, 座標が順次入力される. この座標は, ウィンドウ上のピクセルを 1 単位とする. そこで, 入力される座標が 5 ピクセル以上離れるごとに節点として座標を記録して点数を減らし, より扱いやすい近似の折れ線を作る.

次に, 指示曲線と目的曲線上の点同士をパラメータを用いて対応付ける. これにより, 2 本の曲線の位置関係を明らかにする. パラメータ s は指示曲線と目的

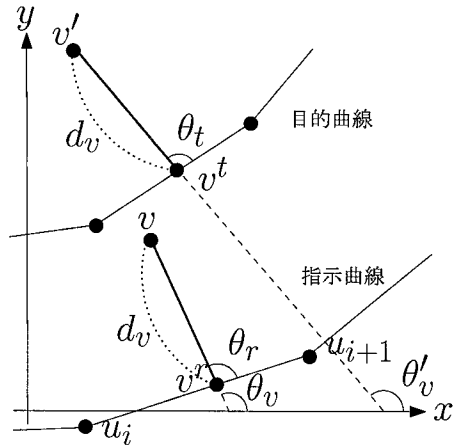


図3 変形後の頂点の位置の計算

曲線上の点をそれぞれ値域 $[0,1]$ に写す. 曲線上の点 u に対してパラメータ値 $s(u)$ を求める. まず, 曲線の始点から u までの線分の長さを合計する. そして, その値をその曲線全体の長さで割ることによってパラメータ値を求める. したがって, 始点 u_s ではパラメータ $s(u_s) = 0$ となり, 終点 u_t では $s(u_t) = 1$ となる.

2.4 対象領域と頂点の移動

対象領域とは, 含まれる頂点を移動させることによって変形を行う領域である. この領域はスケッチされた指示曲線と目的曲線から求める. まず, 対応付けられた指示曲線上の点と目的曲線上の点の距離を求める. そして, それぞれの曲線から, 求めた距離の最大値の 2 倍以内の曲線を含む連結な面を対象領域とする.

提案手法では, 対象領域に含まれる各頂点 v の座標を指示曲線上の最近点 v^r の座標を用いて表すことによって, 指示曲線と目的曲線の対応付けから頂点の移動後の座標を求める (図3). 頂点 v を変形平面上の直交座標系に射影した座標を v^r を用いて書き換える. 頂点 v^r の x 座標を v_x^r , y 座標を v_y^r とする. また, 頂点 v から最近点 v^r までの距離を d_v , 変形平面の x 軸と最近点 v^r と頂点 v を結ぶ線分とのなす角を θ_v とすると, 頂点 v の変形平面上的座標 (v_x, v_y) は,

$$(v_x, v_y) = (v_x^r + d_v \cos \theta_v, v_y^r + d_v \sin \theta_v)$$

と書き換えることができる (図3).

対象領域内の頂点 v に対する変形後の頂点 v' の座標は求めるために, まず v' に対する目的曲線上の最近点 v^t を求める. v^t は目的曲線でパラメータ $s(v^t) = s(v^r)$ となる点とする.

v^r, v^t は指示曲線, 目的曲線上にあり, それらを含む線分の角度は異なっている. これらの線分が重なるように変形したとき, $\langle v, v^r \rangle$ と $\langle v', v^t \rangle$ も重

なるように v' の位置を決める必要がある。また、指示曲線、目的曲線は折れ線で近似しているため、図3の θ_r と θ_t を折れ線におきかえる前のそれぞれの曲線の接線と、 $\langle v, v^r \rangle, \langle v', v^t \rangle$ のなす角度とする。そして、 $\theta_r = \theta_t$ となる位置に v' を移動することにする。そのため、 $\theta_r = \theta_t$ となったときの θ'_v を求め、 $v' = (v'_x, v'_y)$ を、

$$(v'_x, v'_y) = (v_x + d_v \cos \theta'_v, v_y + d_v \sin \theta'_v)$$

とする(図3)。

2.5 メッシュの整合性保証

表面特徴の変形は頂点 v を 2.4 で求めた v' に移動することによって行なわれる。しかし、頂点を移動すると、面の形が崩れたり、面が裏返り、他の面と重なる事が考えられる。そこで、提案手法では、隣接する頂点同士の移動後の座標における位置関係の変化を小さくするために、指示曲線からの距離が遠いほど、移動距離を小さくする。まず、対象領域内の各頂点 v に対して v^r との距離 d_v の最大値を d_{\max} とする。移動ベクトル Δv を、

$$\Delta v = v' - v$$

とすると、変形平面上の頂点の移動後の座標 v'' は

$$v'' = v + (1 - d_v/d_{\max})^2 \Delta v$$

と表せる。図4に、図1に対する変形結果を示す。耳の淵が目的曲線の形状に変形され、それに伴って周辺の頂点も移動している。

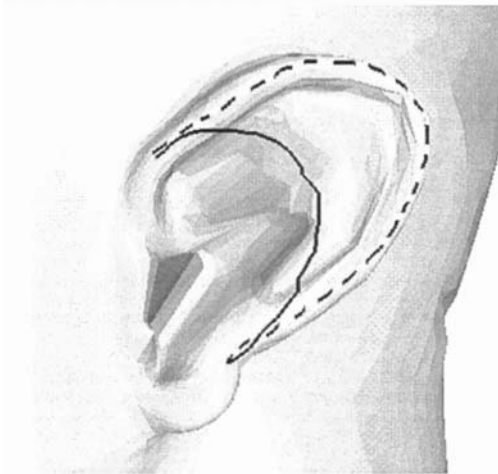


図4 変形結果。

2.6 メッシュの形状の修正

指示曲線と目的曲線という2本の曲線のみを基にメッシュの表面特徴を変形すると、変形結果が図4の耳の内部のように一部の形状が歪むことがある。そこ

で、Shefferらの変形アルゴリズム[6]を応用して、入力されたメッシュの形状に近づける変形を行なうことによって修正する。まず、コントロール頂点を定める。これは、固定されて動かない頂点である。したがって、修正の必要のない対象領域に含まれない頂点はコントロール頂点とする。また、変形によって移動した頂点の中心は目的曲線の周辺に集まっているので、その周辺の頂点もコントロール頂点とする。本研究では、目的曲線の各節点から最も近い頂点をコントロール頂点と定めた。これによって、変形した形状を維持しつつ入力された形状にメッシュを近づけていくことになる。図5は、図4からコントロール頂点を定め、それらを結ぶ辺を実線にしたものである。

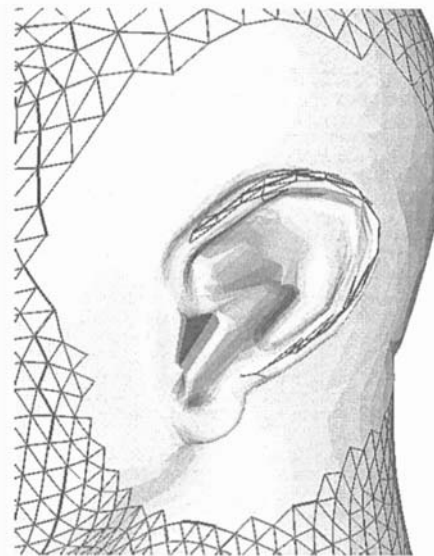


図5 コントロール頂点。

コントロール頂点の集合 V_c が定まると、残りの頂点の位置を以下を繰り返すことによって計算する。

STEP 1 各頂点 v に対して、もし v が V_c に含まれていなければ、頂点 v の位置を、式(3)を用いて更新する。

STEP 2 V_c に含まれている各コントロール頂点に対して、高さ補正 h_i (式(2))を計算し、隣接する頂点の位置を、式(3)を用いて更新する。

STEP 3 STEP 1, 2 を収束するまで繰り返す。

頂点の法線ベクトルは、頂点が更新される度に更新される。したがって、頂点を射影する平面 P (式(1))は、頂点の位置が更新されると変化する。それに伴って、 v', v'_i の座標も変化して行く。したがって、2.1節で保持しておいた位置関係を示す λ_i と β_i の値以外

は反復するごとに更新される。また、頂点の更新による移動距離の最大値が一定の値以下になったとき、収束とみなすことにする。この値は、ユーザが任意に定めるものとする。なお、この値が小さければ小さいほど、入力された形状に近づく。図 6 は、このアルゴリズムを行なった結果である。耳の端の形状を保ったままその他のメッシュの形状が入力された形状に近づいていることが分かる。



図 6 修正結果.

3. 計算機実験

本節では、提案したアルゴリズムを実装し、計算機実験を行なった結果を示す。まず、実験環境を表 1 に記す。

本実験では、manequin モデル (頂点数 6743, 面数 13424), kitten モデル (頂点数 134448, 面数 268896) Chinese dragon モデル (頂点数 15806, 面数 305608) の 3 つのメッシュを入力として用いた。manequin モデルの変形については、前節に示した。

ここでは、kitten モデル, Chinese dragon モデルに対する実験について述べる。それぞれの入力を、図 7, 図 9, 出力を、図 8, 図 10 に示す。どの入力に対しても、要求した変形が行えていることがわかる。

次に、計算時間について述べる。提案したアルゴリズム

表 1 実験環境.

OS	Windows XP (SP2)
CPU	Pentium M (2.10 GHz)
メモリ	1.00 GB RAM

ムを用いてそれぞれのモデルを変形した際にかかった計算時間を表 2 に示す。この際、メッシュの修正を行なう際の反復の収束は、頂点の移動距離がメッシュの平均辺長の 10 分の 1 以下になったときとした。表 2 には、

- 対象領域に含まれる頂点数.
- 形状の保存に要した時間.
- 曲線を元に変形するのに要した計算時間.
- メッシュの修正に要した反復回数.
- メッシュの修正に要した時間.
- 計算時間の合計

を示す。

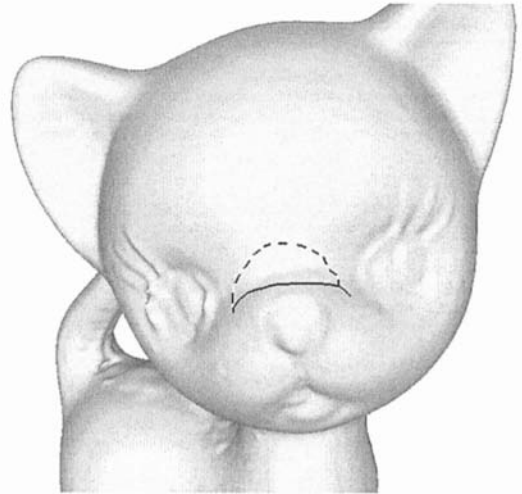


図 7 kitten モデルに対する入力.

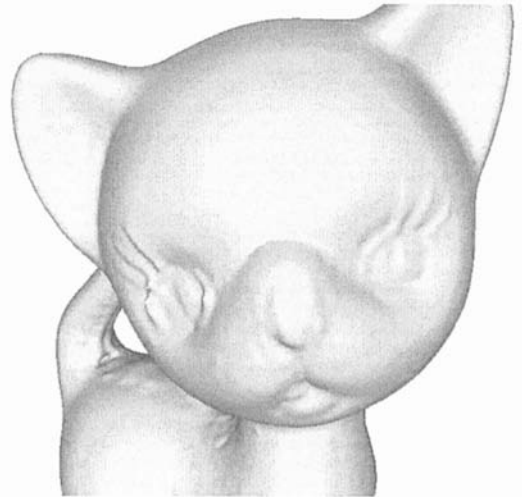


図 8 kitten モデルを変形した結果.

表 2 計算時間.

	各ステップに要した時間					
	頂点数	形状保存 [秒]	変形 [秒]	反復回数	修正 [秒]	合計 [秒]
manequin モデル	1671	0.109	0.031	11	0.438	0.578
kitten モデル	2736	0.609	0.531	17	2.985	4.125
Chinese dragon モデル	7312	0.718	0.875	14	6.032	7.625



図 9 Chinese dragon モデルに対する入力.



図 10 Chinese dragon モデルを変形した結果.

表 2 より, 計算時間は, 長くとも数秒程度であるので, 短時間で変形が可能であると考えられる. また, 要する計算時間は, ほとんどがメッシュの修正に要するものであるということがわかる.

4. 結 論

本研究では, Kho らのアルゴリズムを改良して, スケッチを基にメッシュの表面特徴を変形するアルゴリズムを提案した. さらに, Sheffer らのアルゴリズムを適用することによって変形結果を, 入力された形状を考慮することによって修正するアルゴリズムも提案した. また, 計算機実験によって提案したアルゴリズムの有効性を示した.

今後の課題としては, 変形の対象となる表面特徴の認識が考えられる. それによって, 対象領域の決定やコントロール頂点の選択方法を適切に定めることができ, より良い結果が得られると考えられる. また, メッシュの修正の計算時間の短縮による対話性の向上も考えられる.

参 考 文 献

- [1] M. S. Floater, "Mean value coordinate," *Computer Aided Geometric Design*, Vol. 20, Issue. 1, pp. 19-27, 2003.
- [2] Y. Kho and M. Garland, "Sketching mesh deformations," *Proceeding of the 2005 Symposium on Interactive 3D Graphics and Games*, pp. 147-154. 2005.
- [3] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rossil, H.-P. Seidel "Laplacian surface editing," *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry processing*, pp. 179-188, 2004.
- [4] A. Nealen, O. Sorkine, M. Alexa and D. Cohen-Or, "A sketch-based interface for detail-preserving mesh editing," *ACM Transactions on Graphics*, Vol. 24, No. 3, pp. 1142-1147, 2005.
- [5] T. Sederberg and S. R. Parrrt, "Free-form deformation of solid geometric models," *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151-160, 1986.
- [6] A. Sheffer and V. Krayevoy, "Shape preserving mesh deformations," *ACM SIGGRAPH 2004 Sketches*, pp. 39. 2004.