

# メッシュ上への制御可能な曲線生成手法

金 井 崇<sup>†</sup>

本論文では、三角形メッシュ上で制御可能な曲線を定義するための一手法として、パラメトリック曲線を生成するための手法について提案する。メッシュ上の曲線は、メッシュモデリングやテクスチャマッピング、リメッシング、モーフィングなどの応用において、領域特定の境界曲線を定義するためによく用いられる。ここで定義する曲線は、厳密な曲線ではなくその区分線形近似であるが、メッシュの上に完全に乗っていることが保証される。基本的なアイデアは、直接メッシュ上に曲線を定義するかわりにその球体パラメータ上に定義することにある。曲線の計算は、球体パラメータ上に指定された制御点だけを用いて行われ、メッシュの頂点数に依存しない。これより、密なメッシュにおいてもインタラクティブな曲線の生成・編集が可能である。

## A Method for Creating Controllable Curves on Meshes

TAKASHI KANAI<sup>†</sup>

In this paper, we propose a method for creating parametric curves as one of methods for generating controllable curves on triangular meshes. A curve on a mesh is frequently used as a boundary curve of a specific region of a mesh in mesh modeling and applications such as texture mapping, remeshing or morphing. Although the curve defined in this paper is a piecewise linear approximation of a strict parametric curve, it is guaranteed that such a curve is just on a mesh. The basic idea is creating a curve on a spherical parameterization instead of direct definition on a mesh. The computation of this curve is done by using only the control points on a spherical parameterization which does not depend on the number of vertices in a mesh. This enables interactive creation/modification of curves even for dense meshes.

### 1. はじめに

非構造の三角形メッシュ(以下メッシュと呼ぶ)は、任意の幾何や位相を扱うことができるという点で、コンピュータグラフィックス(CG)の中でも用途の広い曲面表現である。特に最近では、レンジスキャナなどの形状測定技術の発達により、大容量のメッシュが比較的簡単に得られるようになった。しかしながら、計測情報によって得られるメッシュはそのまま利用できるわけではなく、通常何らかの加工を施してから利用する場合が多い。

ここでは特に、メッシュ上に曲線を定義することを考える。メッシュ上への曲線は、テクスチャマッピング、リメッシング、あるいはモーフィングにおける境界線としてよく用いられている。これらのアプリケーションにおいては、曲線としての厳密性、すなわち、メッシュ上の曲線自体が数式により厳密に定義されて

いること、は特に求められていない。そのかわり、滑らかな曲線であることと、ユーザが自由に曲線を生成・修正できるだけのインタラクティブ性を求められる。

本研究では、メッシュ上に自由に生成・修正できるパラメトリック曲線の定義手法について提案する。このようなことを実現する過去の研究としては、まず Krishnamurthy らの方法<sup>14)</sup>があげられる。文献 14)では、メッシュ上のサンプル点を B スプライン曲線で補間することを基本とした、メッシュ上の密なサンプル点群の生成手法について述べている。ただし、メッシュ上に定義されるのはサンプル点群のみであり、厳密には曲線はメッシュ上に乗っていない。また Kanai らは、メッシュ上に近似直線を定義するための方法を提案した<sup>8)</sup>。この方法は、Dijkstra のアルゴリズムに基づき、メッシュのグラフ構造を適応的詳細化することで、近似的に 2 点間の最短経路を計算することを基本とする。ただし、Dijkstra のアルゴリズムを一度形状全体に施す必要があり、近似精度を高めるためにはその分計算時間がかかる。

他に、だれでも思いつきそうな手法として、メッシュ

<sup>†</sup> 理化学研究所ものづくり情報技術統合化研究プログラム  
Integrated Volume-CAD System Research Program,  
RIKEN

上に定義された制御点を使って空間上に曲線（たとえば B スプライン曲線）を生成し、その曲線を投影することでメッシュ上の曲線を生成する手法が考えられる。しかし、この方法の問題点として、曲線をどの方向に投影すべきかを決定する必要がある。仮に曲線上のサンプル点をメッシュ上の最近点に投影する場合、サンプル点に対するメッシュの最近点の探索が必要になる。この探索には付加的なデータ構造（たとえば八分木構造のような空間分割構造）を必要とし、メッシュの頂点数や面数が多くなると計算コストがかかる。我々は曲線生成・編集におけるインタラクティブ性を重要視しているため、計算コストのかかる操作はなるべく避けたい。また、すべてのサンプル点に対し同一方向に投影する場合、形状によっては（特に凹凸が激しい部位で）投影に失敗する可能性がある。

本手法の基本的なアイデアは、メッシュそのものではなく、メッシュから生成されるパラメータ空間上に曲線を定義することにある。この考えは、B スプライン曲面や NURBS などのパラメトリック曲面形式において、パラメータ空間上での幾何操作を行うことに似ている。たとえば、等パラメータ線は、この考え方に基づくものである。メッシュに関する研究でもこの考え方を応用しているものは存在する。たとえば Sheffer らのリメッシングに関する研究<sup>23)</sup>では、メッシュから作成したパラメータ空間上でフェアリングを適用することで、パラメータを歪みのないよう修正する手法を提案している。

このような考えの正当性として、近年のメッシュモデリングの研究の大部分が、パラメータ化を必要としていることがあげられる。先にあげたようなテクスチャマッピング、リメッシング、モーフィングなどのアプリケーションにおいて、そのほとんどがパラメータ化を経由した研究であることは今や疑いのない事実である。したがって、本研究のようなパラメータ化を前提としたアプローチは、今後普遍的な考え方になるものと確信する。

本手法の特徴として、パラメータ空間上に定義される曲線は、厳密な数式に基づくパラメトリック形式の曲線であるため、制御性は維持される。また本手法は、制御点に相当するメッシュ上の頂点群のみを計算に使うため、メッシュの頂点数に依存しない局所的処理により計算可能であることが特徴である。このため、インタラクティブに曲線を生成・修正することが可能である。また本研究では、このメッシュ上のパラメトリック曲線に基づく 2 つの応用として、メッシュ上への曲線の編集およびメッシュの領域分割についても論じる。

本論文の構成は以下のとおりである。2 章には、本手法の概要を示す。3 章には本手法を構成するそれぞれのアルゴリズムの詳細を関連研究とともに述べ、4 章には応用例と議論を示す。最後に 5 章で結論と展望を述べる。

## 2. 概要

図 1 に本手法全体の概要を示す。入力は、任意位相のメッシュであり、出力はメッシュ上に描かれる曲線となる。手法は以下のとおりである。

- (1) メッシュ  $M$  に対し、球体パラメータ化  $\phi: M \mapsto P$  を施し、パラメータ  $p \in P$  をメッシュ上の頂点  $v \in M$  に保存しておく。この計算は前処理によって行われ、一度計算するだけでよい。
- (2) メッシュ上の点群を指定した後に、対応する球体上の頂点のパラメータを用いて、球体上に四元数曲線 (quaternion curve)  $L(t)$  を生成する。
- (3) この曲線を球体パラメータ上に投影し、区分線形近似  $p_i \in P$  ( $i = 1 \dots n$ ) を得る。 $p_i$  をメッシュ上へ逆写像することにより、メッシュ上の区分線形近似曲線  $v_i \in M$  ( $i = 1 \dots n$ ) が生成される。

本手法では、パラメータ化に球体パラメータ化 (spherical parameterization) を用いることとする。しかしながら、1 つの問題として、この球体パラメータ化の利用が、本手法の適用範囲を種数 0 のメッシュに限定することとなる。

ただし、任意の種数のメッシュに適用する方法がな

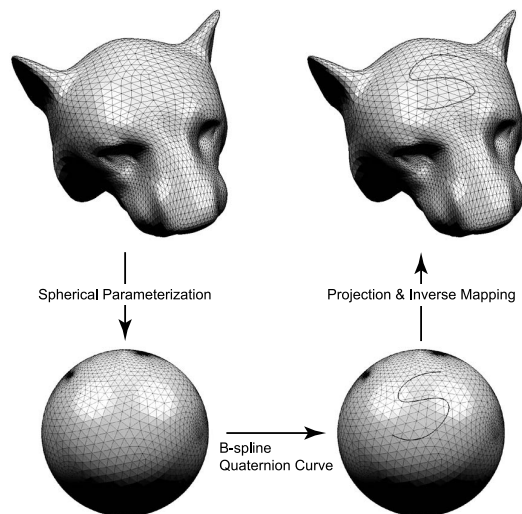


図 1 本手法の概要

Fig. 1 Overview of our basic procedure for curve generation.

いわけではない．1つの方法としては，平面パラメータ化を利用することが考えられる．この場合，まず形状全体をパラメータ化してテクスチャアトラスを生成し<sup>17),22)</sup>，アトラス上で二次元の平面パラメトリック曲線を適用することで，上記の手順と同様のことが実現できるものと思われる．ただし，この方法における問題点としては，全体形状をいくつかのバッチに分割し，さらにテクスチャアトラスを作成するという，少々複雑な手順が必要となることである．また，アトラスの中でバッチをまたがるような曲線を描くときには，境界における連続性を考慮する必要がある．

他の可能性としては，Leeらの研究<sup>15),16)</sup>に見られるような，局所的なパラメータ化を行うことである．これは，線が置かれる（少々大きめの）領域を抽出し，その領域だけをパラメータ化する．こうすることで，上記の平面パラメータ化の場合に起こる連続性の問題は解決できる．しかしながら，線の描画ごとにパラメータ化を計算するために，計算時間が多少かかることが予想される．これは特に4.1節で述べる曲線の移動や回転の際のインタラクティブ性を損なう可能性がある．

メッシュ上に生成される曲線は，厳密には曲線の区分線形近似（*piecewise linear approximation*）である．なぜならばメッシュ自体が区分線形曲面として定義されるからである．このメッシュ上に厳密に曲線を定義することは不可能である．ただし，パラメータ上での曲線から区分線形近似への変換により，線分  $\overline{v_i v_{i+1}}$  はメッシュ  $M$  上に乗ることが保証される．

複数の曲線を定義する場合には，上記(2)，(3)の手順を繰り返し実行すればよい．

### 3. アルゴリズムの詳細

本章では，前章で述べた曲線の定義手法で利用する3つの項目，球体パラメータ化，四元数曲線，曲線から区分線形近似への変換の詳細についてそれぞれ説明する．

#### 3.1 球体パラメータ化

これまで，球体パラメータ化として様々な手法が提案されている．古くは，Kentら<sup>11)</sup>が，モーフィングの際メッシュを球体に投影するための手法として提案した．ただしこの方法は，星型形状にしか適用できないという問題があった．しかしごく最近，種数0のメッシュに対するいくつかの実用的な方法が提案されている<sup>1),6),20)</sup>．

ここでは，代表的な球体パラメータ化に関する3つの方法として，Alexaパラメータ化<sup>1)</sup>（以下 Alexa），Tutteパラメータ化<sup>6)</sup>（以下 Tutte），および等角

（*conformal*）パラメータ化<sup>6)</sup>（以下 Conformal）を試し，曲線描画に対する結果の違いについて比較・検討する．これら3つの手法はいずれも，球体上の頂点座標の更新操作を繰り返すことによる収束計算となる．エネルギーを計算し，その各ステップにおけるエネルギーの差分がある閾値以内になったとき，処理を終了する．頂点  $p_i \in P$  におけるオペレータ  $q_i$ ，および，更新後の頂点  $p'_i$  は手法によって異なる．それぞれ次式により表される：

Alexaパラメータ化

$$q_i = c_{alexa} \frac{\sum_{\{i,j\} \in e}^{\nu} (p_i - p_j) |p_i - p_j|}{\nu}, \quad (1)$$

$$p'_i = \frac{p_i + q_i}{|p_i + q_i|}.$$

Tutteパラメータ化

$$q_i = p_i - (\Delta p_i \cdot p_i) p_i,$$

$$\Delta p_i = \sum_{\{i,j\} \in e}^{\nu} (p_j - p_i), \quad (2)$$

$$p'_i = p_i + c_{tutte} q_i.$$

等角パラメータ化

$$q_i = p_i - (\Delta p_i \cdot p_i) p_i,$$

$$\Delta p_i = \sum_{\{i,j\} \in e}^{\nu} (a_{j,i}^{\alpha} + a_{j,i}^{\beta}) (p_j - p_i), \quad (3)$$

$$p'_i = p_i + c_{conf} q_i,$$

$$p'_i = \frac{p'_i - \sum_j^P p_j}{|p'_i - \sum_j^P p_j|}.$$

ここで  $\nu$  は頂点の価数， $c_{alexa}$ ， $c_{tutte}$ ， $c_{conf}$  はそれぞれ収束定数を示す． $a_{j,i}^{\alpha}$ ， $a_{j,i}^{\beta}$  はエッジ  $e = \{i, j\}$  に隣接する2つの面  $\alpha$ ， $\beta$  の角度に関する量である．

この球体パラメータ化の計算は，階層的な解法<sup>26)</sup>を利用して行う．まず，入力メッシュを簡略化し，プログレッシブ・メッシュ形式<sup>7)</sup>として保存する．疎なメッシュ（我々の例では100頂点）からスタートし，頂点数を一定の割合，たとえば1つ前のメッシュの10倍の頂点，になるように段階的に増加させる．この増加は頂点分割操作を繰り返すことで行われる．各段階のメッシュにおいて最適化によるパラメータ化処理を実行する．各頂点分割操作において，新しく生成された頂点のパラメータは，周りの頂点より形成されるカーネルの中に入るように決定され，その値をパラメータ化処理における初期値とする．各段階におけるパラメータ化処理は，初期値が最適解に近い値であるため，より少ないステップ回数で収束する．

図2に，異なる球体パラメータにおいて本手法を通

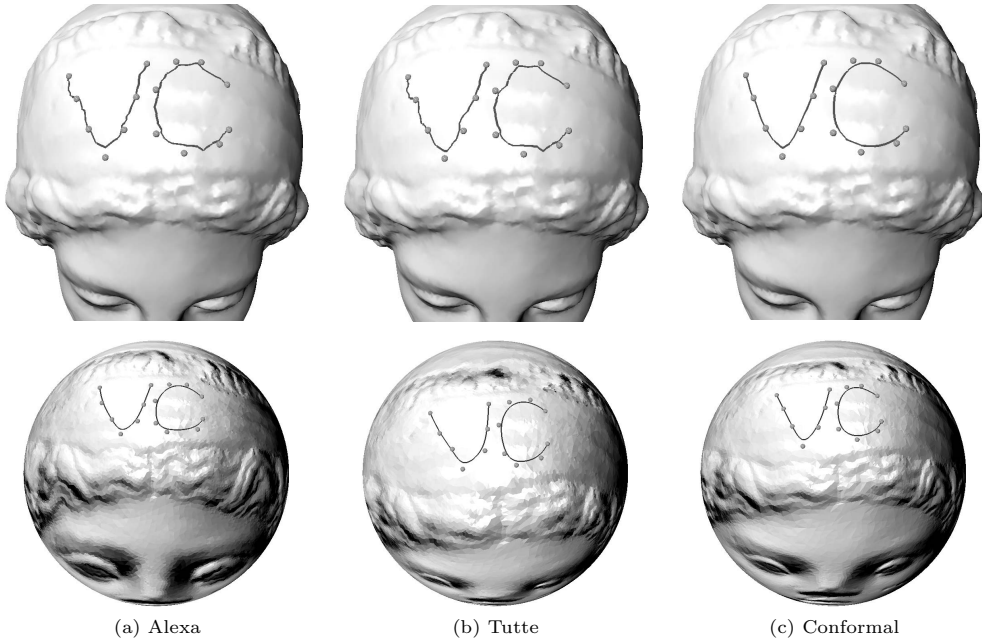


図 2 “Igea” メッシュ (33,591 頂点, Cyberware Inc. の提供による) 上での異なる球体パラメータ化に対する曲線描画結果の比較. 球は制御点を示す. (a) Alexa パラメータ化. (b) Tutte パラメータ化. (c) 等角パラメータ化. 上段: メッシュ上の曲線. 下段: 球体パラメータ上の曲線

Fig. 2 Comparison of curve drawing results on “Igea” mesh (33,591 vertices, courtesy of Cyberware Inc.) with different spherical embeddings. Control points are represented as spheres. (a) Alexa mapping. (b) Tutte mapping. (c) Conformal mapping. Top: Curves on a mesh. Bottom: Curves on its spherical embedding.

用した結果の比較を示す. 3つの図はいずれも, 同じ頂点を用いた曲線による文字を描画しており, 異なるのは球体パラメータのみである. 結果を見て分かる通り, 本手法はパラメータ化の「品質」に大きく依存する. Alexa と Tutte では, メッシュ上の曲線に細かい振動が見られる一方, Conformal では滑らかな曲線が生成されている. このことは, 球体パラメータ化の「等角性」が明確に現れていることを示している. また等角パラメータ化では, メッシュの接続性に依存しない品質を保証することが文献 4) より示されている.

球体パラメータ上の曲線は, どれも滑らかなのは当然だが, 異なる形状が生成されている. これは, 頂点の持つパラメータ値が, 球体パラメータによって異なるからである. ただし, 実際に利用する側からみれば, 球体パラメータ上の曲線はあくまで陰的なものであり, 実際に目に触れることなく操作を行うことができる.

### 3.2 四元数曲線

我々は, 球体上の曲線の定義のために, 四元数曲線 (Quaternion Curve) を利用する. 四元数曲線は, もともとコンピュータアニメーションやロボティクス

の分野での方向制御に用いられる古典的なツールである. これまでに多くの四元数曲線が提案されている<sup>3),13),19),24)</sup>.

この中で我々は Kim らの単位 B スプライン四元数曲線<sup>13)</sup> を用いている. その理由として, 滑らかに接続された複合曲線の生成が簡単であることがあげられる. また, 計算は制御点の数にのみ依存し, かつ, B スプライン曲線の代数的, 微分学的性質を世襲している. さらに, 他のベジエやエルミート形式などの曲線と違い, 任意の数の制御点に対して定式化されていることから, 曲線的设计の自由度は高い. 我々のこの目的は, 滑らかで制御可能な曲線を生成することであり, その意味で適した手法であるといえる.

曲線上の点  $q(t)$  は以下の式で求められる:

$$q(t) = q_0 \prod_{i=1}^n \exp(w_i \tilde{B}_{i,k}(t)), \quad (4)$$

$$w_i = \log(q_{i-1}^{-1} q_i),$$

ここで  $q_i$  は四元数の制御点,  $\exp, \log$  はそれぞれ四元数の指数写像, 対数写像である. また  $\tilde{B}_{i,k}(t)$  は B ス

ブライン基底関数  $B_{j,k}(t)$  の蓄積的形式 (cumulative form)

$$\tilde{B}_{i,k}(t) = \sum_{j=i}^n B_{j,k}(t),$$

である .

上記の単位 B スプライン四元数曲線を使って球体上に曲線を生成するとき、考慮すべき 1 つの問題がある . もともと単位四元数  $q = (\cos \theta, \sin \theta \cdot (a, b, c)) \in S^3$  は方向  $(a, b, c)$  と回転  $\theta$  により規定される . しかしながら、Miura の論文<sup>19)</sup>でも指摘しているように、球面上に曲線を生成する問題は、与えられたパラメータ値に対応する「方向」を定義することとなる . したがって、球面上の 1 点は無限個の単位四元数に対応することになる . 我々のケースにあてはめてみると、球面上の 1 点から単位四元数を規定するとき、どのようにして回転に相当する量を算出するかが問題となる .

我々のこの問題に対する単純な解法は、すべての四元数への変換に対し、 $\cos \theta = 0, \sin \theta = 1$  のように固定してしまうことである . この固定を正当化する理由として、

- (1)  $\theta$  は 0 から  $\pi$  の間で定義されるが、 $\theta$  を 0 もしくは  $\pi$  に近づけると、 $\sin \theta$  は 0 に近づくため、 $\sin \theta \cdot (a, b, c)$  の項よりも  $\cos \theta$  の項の方が相対的な値は大きくなり、結果として曲線はその制御点から遠ざかる傾向にある . しかしながら、その差は視覚的に非常にわずかなものである .
- (2) 単位四元数は  $|q| = 1$  が成り立つため、 $\cos \theta$  が 0 以外の値を持つと、 $\sin \theta \cdot (a, b, c)$  の長さは 1 よりも小さくなり、結果として曲線上の点  $q(t)$  は球面上には乗らない . よって、球面上に乗るようになるためには、 $(a, b, c)$  を正規化する必要があるが、上記のように固定することでその手間が省ける .

の 2 点があげられる . しかし、(1) に関しては、あくまで視覚的な判断に基づくものであり、曲線の品質を定性的に調査することは、今後の課題としてあげられる .

### 3.3 投影と逆写像

前節で述べた単位 B スプライン四元数曲線は、厳密には球体パラメータ上に乗っていない . そこで、曲線を球体パラメータ上に投影し、曲線の区分線形近似を生成する . 区分線形近似は、メッシュ上のポリラインの形  $l = \{p_1, p_2, \dots, p_n\}$  で表現される . ポリラインの頂点は、メッシュ上の頂点、もしくはメッシュのエッジ上の点となる .

投影は以下のアルゴリズムにより行うことができる :

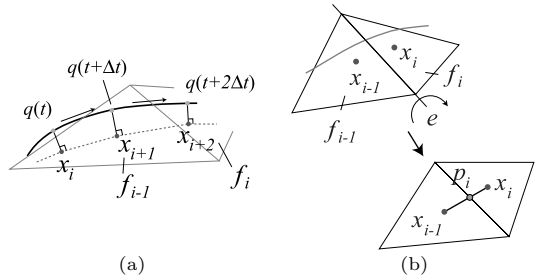


図 3 曲線投影アルゴリズム . (a)  $q(t)$  からの最近点  $x_i$  の計算 . (b) エッジ  $e$  と線分  $\overline{x_i x_{i-1}}$  の交点  $p_i$  の計算

Fig. 3 The curve projection algorithm. (a) Computation of a nearest point  $x_i$  from  $q(t)$ . (b) Computation of an intersection point  $p_i$  between an edge  $e$  and a line segment  $\overline{x_i x_{i-1}}$ .

#### Algorithm1 曲線投影アルゴリズム

入力 : 球体パラメータ  $P$ , 四元数曲線  $q(t)$

出力 :  $P$  上の点列  $p_1, p_2, \dots, p_n$

$t = 0; p_i = q(t);$

while (  $p_i \neq q(t = 1)$  )

$x_{i-1} = p_i;$

while (  $f_{i-1} == f_i$  )

$t = t + \Delta t;$

曲線上の  $q_i = q(t)$  の最近点  $x_i$  と対応する面  $f_i$  を計算 ; ... (1)

if (  $x_i == nil$  )

$\Delta t = \Delta t * 0.1; // \Delta t$  を小さくする

end

end

$f_i$  と  $f_{i-1}$  の隣接エッジ  $e$  と線分  $\overline{x_i x_{i-1}}$  の交点  $p_i$  を計算 ; ... (2)

end

end

アルゴリズムは、 $t = 0$  からスタートし、 $t$  を  $\Delta t$  ずつ追加することによって、曲線上の点  $q(t)$  を求める . この点に対し、メッシュとの最近点  $x_i$  と対応する面  $f_i$  を求める ( 図 3 (a), Algorithm 1 の (1) ). 最近点探索において、 $x_i$  は面  $f_i$  の垂線の足として計算される .

もし、ある最近点  $x_{i-1}$  と、その次のステップにおける最近点  $x_i$  における対応面が異なるとき ( それぞれ  $f_{i-1}, f_i$  とする ), それらの面の隣接エッジ  $e$  と  $\overline{x_i x_{i-1}}$  との交点を計算し、その点をポリライン上の点  $p_i$  とする ( 図 3 (b), Algorithm 1 の (2) ). このとき、 $f_{i-1}$  と  $f_i$  が同一平面上になるよう、 $e$  を軸に回転する . 上記の操作を、 $t = 1$  となるまで繰り返す .

ここでステップ間隔  $\Delta t$  のとり方については気をつける必要がある . (2) においてアルゴリズムが最近点

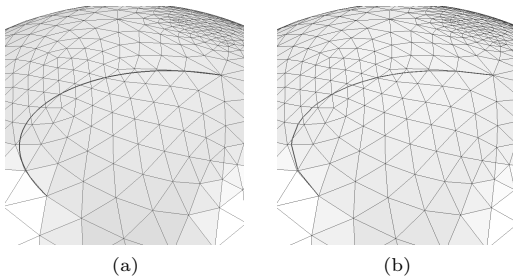


図4 (a) 球体上の B スプライン四元数曲線 . (b) 投影された区分線形近似

Fig. 4 (a) B-spline quaternion curve on sphere. (b) Piecewise linear approximation projected on a spherical embedding.

を計算する際、2つの異なる面  $f_{i-1}$ ,  $f_i$  が1つのエッジ  $e$  を共有することが前提となる。  $\Delta t$  を大きくとりすぎると、 $f_{i-1}$ ,  $f_i$  が隣接しない、すなわち、隣接の面を飛び越えて  $f_i$  が見つかるような状況が起こりうる。そのような場合は、最近点を見つけることができない。逆に、 $\Delta t$  を小さくするとステップ数がかかる。ここではいくつかの実験の結果、まず全エッジの平均の長さの1/10を初期値にとり、アルゴリズムの中で、最近点を見つけれないときに限り、ステップ間隔を小さくするという方法をとった。

図4に、球体上のBスプライン四元数曲線と、球体上に投影された区分線形近似を示す。球体パラメータ化によりメッシュとの対応付けがされているため、逆写像は至極簡単である。交点計算の際、対応するメッシュのエッジ上の点をあわせて計算すればよい。

#### 4. 応用

本章では、前章で述べた曲線生成手法の応用について述べる。ここでは2つの応用例として、曲線描画ツールとしての応用と、メッシュ分割してテクスチャマッピングへの応用について議論する。

##### 4.1 曲線の編集

我々の曲線生成・編集ツールは、インタラクティブに曲線を生成できるほか、メッシュ上を自由に移動・回転することが可能である。図5にあるメッシュ上の文字を回転、移動した例を示す。

メッシュ上で曲線文字を動かすときの1つの問題として、メッシュ上の面の大きさと、球体パラメータ上の対応する面の大きさの比率が異なることがある。これより、球体パラメータ上の面の大きさが相対的に小さい部位（たとえば、図5の腕の部分など）を通るとき、文字が大きくなってしまふ。

このことを解決する単純な方法として、文字が存

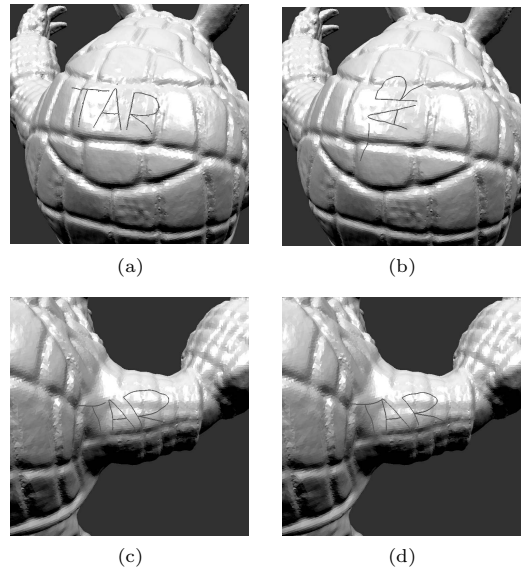


図5 “Armadillo” メッシュ (172,974 頂点, スタンフォード大学の提供による) に対する曲線編集例 . (a) 背中に描かれた文字 “TAR” . (b) 90° 回転した結果 . (c) スケーリング調整なしで腕への平行移動を行った結果 . (d) スケーリング調整ありで腕への平行移動を行った結果

Fig. 5 Curve editing examples on “Armadillo” mesh (172,974 vertices, courtesy of Stanford University). (a) Characters “TAR” on the back. (b) 90°-Rotation. (c) Translation to the arm without scaling effect. (d) Translation to the arm with scaling effect.

在する位置によって文字の大きさを調整する。まず、 $M$  上の面  $f_M$  と球体パラメータ  $P$  上の対応する面  $f_P$  の面積比率係数  $\gamma = \sqrt{\text{area}(f_P)/\text{area}(f_M)}$  を定義する。この係数では、文字が乗っている領域の影響をスケーリングに反映するために、面の面積のスケール比を利用している。一方で、実際のスケーリングに関しては、各制御点に対して文字の重心位置からの距離のスケーリングを行っている。そのため、面積のスケール比の平方根をとることにより次元を揃えている。

実際のスケーリング処理に関して、現在の実装では、まず、移動前と移動後の文字（たとえば図5の “R”）を構成している線群が乗っている面群に対し、面積比率係数の平均  $\gamma_b$ ,  $\gamma_a$  をそれぞれ計算する。線群の重心位置（簡単のため制御点の平均位置を使っている）からの各制御点までの距離を  $\frac{\gamma_a}{\gamma_b}$  倍した位置を計算することで、スケーリングを行っている。なお、この処理はすべて球体パラメータ上で行っており、四元数曲線を計算する処理の中で行われている。図5(d)に、スケーリングを調整した結果を示す。このときの球体パラメータ上の文字は相対的に小さくなる。

また、描いた文字を中心にメッシュを適応的細分割す

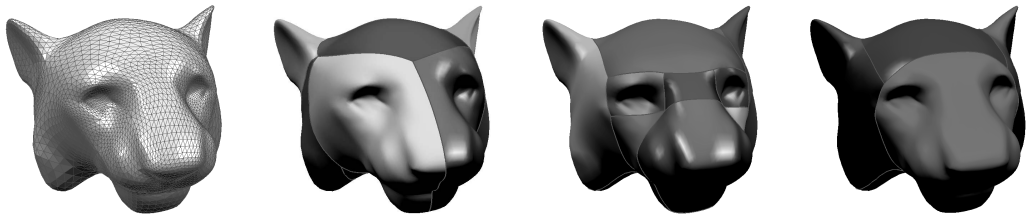


図 6 “tiger’s head” メッシュ (4,034 頂点) における様々なメッシュ分割結果. 各分割における曲線の数はそれぞれ 22, 26, 12 である

Fig. 6 Various mesh decomposition results for “tiger’s head” mesh (4,034 vertices). The number of parametric curves in the decomposition is 22, 26, and 12 (from left to right).

ることでディスプレイマッピング処理を行うことも可能である. 同様のことは Suzuki らの研究<sup>25)</sup>でも行われているが, 文献 25) では, 不可逆処理, すなわち, 移動した後の元形状が損なわれてしまうのに対し, 本手法は完全な可逆処理を, 適応的細分割処理も含めインタラクティブに行うことが可能である.

#### 4.2 メッシュの分割

メッシュ全体のパッチへの分割は, 特にモーフィング<sup>5),9),18),21)</sup>において重要なテーマの 1 つである. モーフィングにおける利用の際, 分割されたパッチ間の境界の滑らかさが, モーフィングの 1 つの重要な要因となる. このメッシュの分割を自動的に行う研究がされている<sup>2),10)</sup>が, 本手法における分割はすべて手動による処理となる. これは利点と欠点がある. 利点としては, ユーザが本当に必要な境界曲線を編集できることである. 本手法は, このようなインタラクティブかつ試行錯誤をともなう処理に耐えうるだけの処理速度を持つ. 欠点としては, すべて手動による処理のため, ユーザによる労力が必要となることである. しかしながら, 自動分割処理と組み合わせて使うことは可能である. たとえば文献 2), 10) の手法であらかじめ自動分割しておいてから, 必要な箇所だけを編集する, といった使い方も可能であろう. 図 6 は 1 つのメッシュに対し, 異なった分割方法で分割した例を示す. この例において, すべて手動で行ったとしても大体 5 分以内で分割処理を行うことが可能である. このような自由な境界曲線の調整は, 自動分割手法では難しい.

本手法では, メッシュ全体だけでなく, 一部の領域に関する抽出も可能である. 図 7(a) は, メッシュの一部の矩形領域を 4 つの曲線により抽出し, テクスチャマッピングを行った例であり, 図 7(b) は境界の一部を修正した例である.

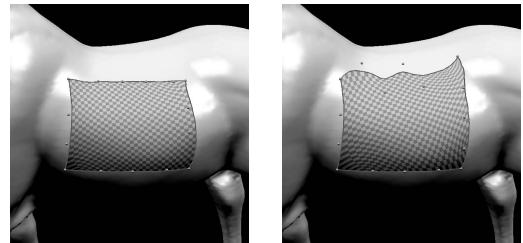


図 7 (a) “horse” メッシュ (48,476 頂点, Cyberware Inc. の提供による) に対する四辺形領域の指定. チェックボードテクスチャを割り当てている. (b) 境界曲線のいくつかの頂点を移動した結果

Fig. 7 (a) Specification of a quadrilateral region in “horse” mesh (48,476 vertices, courtesy of Cyberware Inc.). A check board texture is assigned in this region. (b) Curve editing by moving several control vertices of boundary curves on a mesh.

## 5. 結論と展望

本論文では, 球体パラメータを利用したメッシュ上のパラメトリック曲線の生成手法について提案した. 本手法は, 滑らかで制御可能な曲線をインタラクティブなスピードで生成できることが特徴である. また, メッシュ上での強力な曲線生成・編集ツールとなりうることを, 2 つの応用例を通じて確認した.

今後の課題としては, 主に 2 つの側面からあげられる. 1 つの側面は, 手法の改良であり, そのうちの 1 つは, 種数が 1 以上のメッシュに対する処理である. これは 2 章でも述べたように, テクスチャアトラスを利用する方法が考えられるが, 境界をまたぐような曲線においていかに連続性を維持するかがポイントとなる. もう 1 つは, 点列を補間する補間曲線の生成である. これは球体上の補間四元数曲線<sup>12)</sup>を利用することができると思われる.

もう 1 つの側面としては, 曲線編集のためのインタフェースである. 特に曲線編集のためのタブレットの

利用に興味を持っている。

謝辞 本実験に使用したデータのうち Armadillo はスタンフォード大学, horse, Igea は Cyberware 社のデータである。また, 本研究の一部は (財) 大川情報通信基金によるご支援をいただいた。ここに感謝の意を表する。

### 参 考 文 献

- 1) Alexa, M.: Morphing Polyhedral Shapes with Scattered Features, *The Visual Computer*, Vol.16, No.1, pp.26–37 (2000).
- 2) Boier-Martin, I.M.: Domain Decomposition for Multiresolution Analysis, *Proc. 1st Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, pp.29–40 (2003).
- 3) Buss, S.R. and Fillmore, J.P.: Spherical Averages and Applications to Spherical Splines and Interpolation, *ACM Trans. Graphics*, Vol.20, No.2, pp.95–126 (2001).
- 4) Desbrun, M., Meyer, M. and Alliez, P.: Intrinsic Parameterizations of Surface Meshes, *Computer Graphics Forum (Eurographics 2002 Proceedings)*, Vol.21, No.3, pp.209–218 (2002).
- 5) Gregory, A., State, A., Lin, M., Manocha, D. and Livingston, M.: Interactive Surface Decomposition for Polyhedral Morphing, *The Visual Computer*, Vol.15, No.9, pp.453–470 (1999).
- 6) Gu, X., Wang, Y., Chan, T.F., Thompson, P.M. and Yau, S.-T.: Genus Zero Surface Conformal Mapping and Its Application to Brain Surface Mapping, *IEEE Trans. Medical Imaging*, Vol.23, No.8, pp.949–958 (2004).
- 7) Hoppe, H.: Progressive Meshes, *Computer Graphics (Proc. SIGGRAPH 96)*, pp.99–108, ACM Press, New York (1996).
- 8) Kanai, T. and Suzuki, H.: Approximate Shortest Path on a Polyhedral Surface and Its Applications, *Computer Aided Design*, Vol.33, No.11, pp.801–811 (2001).
- 9) Kanai, T., Suzuki, H. and Kimura, F.: Metamorphosis of Arbitrary Triangular Meshes, *IEEE Computer Graphics and Applications*, Vol.20, No.2, pp.62–75 (2000).
- 10) Katz, S. and Tal, A.: Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts, *ACM Trans. Graphics (Proc. SIGGRAPH 2003)*, Vol.22, No.3, pp.954–961 (2003).
- 11) Kent, J.R., Carlson, W.E. and Parent, R.E.: Shape Transformation for Polyhedral Objects, *Computer Graphics (Proc. SIGGRAPH 92)*, pp.47–54, ACM Press, New York (1992).
- 12) Kim, M.-J., Kim, M.-S. and Shin, S.Y.: A  $C^2$ -continuous B-spline Quaternion Curve Interpolating a Given Sequence of Solid Orientations, *Proc. Computer Animation '95*, pp.72–81, IEEE CS Press, Los Alamitos CA (1995).
- 13) Kim, M.-J., Kim, M.-S. and Shin, S.Y.: A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives, *Computer Graphics (Proc. SIGGRAPH 95)*, pp.369–376, ACM Press, New York (1995).
- 14) Krishnamurthy, V. and Levoy, M.: Fitting Smooth Surfaces to Dense Polygon Meshes, *Computer Graphics (Proc. SIGGRAPH 96)*, pp.313–324, ACM Press, New York (1996).
- 15) Lee, Y. and Lee, S.: Geometric Snakes for Triangular Meshes, *Computer Graphics Forum (Proc. Eurographics 2002)*, Vol.21, No.3, pp.229–238 (2002).
- 16) Lee, Y., Lee, S., Shamir, A., Cohen-Or, D. and Seidel, H.-P.: Intelligent Mesh Scissoring Using 3D Snakes, *Proc. 12th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2004)*, pp.279–287, IEEE CS Press, Los Alamitos CA (2004).
- 17) Lévy, B., Petitjean, S., Ray, N. and Maillot, J.: Least Squares Conformal Maps for Automatic Texture Atlas Generation, *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, Vol.21, No.3, pp.362–371 (2002).
- 18) Michikawa, T., Kanai, T., Fujita, M. and Chiyokura, H.: Multiresolution Interpolation Meshes, *Proc. 9th Pacific Conference on Computer Graphics and Applications (Pacific Graphics)*, pp.60–69, IEEE CS Press, Los Alamitos CA (2001).
- 19) Miura, K.T.: Unit Quaternion Integral Curve: A New Type of Fair Free-Form Curves, *Computer Aided Geometric Design*, Vol.17, No.1, pp.39–58 (2000).
- 20) Praun, E. and Hoppe, H.: Spherical Parameterization and Remeshing, *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, Vol.22, No.3, pp.340–349 (2003).
- 21) Praun, E., Sweldens, W. and Schröder, P.: Consistent Mesh Parameterizations, *Computer Graphics (Proc. SIGGRAPH 2001)*, pp.179–184, ACM Press, New York (2001).
- 22) Sander, P.V., Snyder, J., Gortler, S.J. and Hoppe, H.: Texture Mapping Progressive Meshes, *Computer Graphics (Proc. SIGGRAPH 2001)*, pp.409–416, ACM Press, New York (2001).
- 23) Sheffer, A. and de Sturler, E.: Smoothing An Overlay Grid To Minimize Linear Distortion



- In Texture Mapping, *ACM Transactions on Graphics*, Vol.21, No.4, pp.874–890 (2002).
- 24) Shoemake, K.: Animating Rotation with Quaternion Curves, *Computer Graphics (Proc. SIGGRAPH 85)*, pp.245–254, ACM Press, New York (1985).
- 25) Suzuki, H., Sakurai, Y., Kanai, T. and Kimura, F.: Interactive Mesh Dragging with an Adaptive Remeshing Technique, *The Visual Computer*, Vol.16, No.3-4, pp.159–176 (2000).
- 26) 金井 崇: 頑健かつ高速な等角球体パラメータ化計算手法, *情報処理学会論文誌*, Vol.46, No.2, pp.649–657 (2005).



金井 崇 (正会員)

昭和 44 年生 . 平成 10 年東京大学大学院工学系研究科精密機械工学専攻博士課程修了 . 平成 12 年慶應義塾大学環境情報学部専任講師 . 平成 17 年理化学研究所ものづくり情報技術統合化研究プログラム研究員 . 現在に至る . 形状モデリングの CAD/CAM および CG への応用に関する研究に従事 . 博士 (工学) . 精密工学会 , 画像電子学会 , 芸術科学会 , ACM , IEEE CS 各会員 . 平成 13 年度山下記念研究賞受賞 .

(平成 17 年 7 月 5 日受付)

(平成 17 年 12 月 2 日採録)

---