

# 形状の類似性を用いた曲線洗練法

## Curve Refinement Method Using Shape Similarity

佐藤 信†  
Makoto Satoh

三輪 譲二†  
Jouji Miwa

### 概要

形状変形のための制約条件に基づいて、曲線形状を洗練するためのアルゴリズムを提案する。本アルゴリズムでは、形状特徴の類似性の基準として、形状の相似性と制御点の変化量を使用することにより、変形前の形状的特徴を維持したままで曲線変形が可能である。そのため、制約条件として、スケーリングが変化する相似性を維持するための制約条件を提案する。また、大局的に形状に制約を加えるために、弱制約条件が有効であることを示す。適用分野には、グラフィックス・コンテンツ作成アプリケーションのためのユーザインタフェースがある。特に、初期形状から目的形状に徐々に形状変形が可能であるので、既存のクリップアート等の修正に適する。

### 1 はじめに

本稿では、曲線形状を洗練するためのアルゴリズムを提案する。本アルゴリズムの特徴は以下のとおりである。

1. 変形前の形状的特徴を維持しながら、形状変形のための制約条件に基づいて曲線形状を変形可能である。
2. Bezier 曲線などのパラメトリック曲線を対象とする。
3. 形状的特徴の類似性の基準として、相似性と制御点の変化量を使用する。曲線のパラメータ位置での曲率の変化を使用して、形状の相似性を計算する。
4. 曲線制御点の変化量を独立変数とする線形方程式により、曲線変形のための制約条件を表現する。弱制約により解くことにより、大局的な制約を実現する。

本アルゴリズムの適用分野としては、グラフィックス・コンテンツ作成アプリケーションのユーザインタフェースがある。特に、対話的な操作により、初期形状から徐々に修正を加えて、目的形状を得るグラフィックス・コンテンツ作成方法に適する。なお、Bezier 曲線を対象とした理由は、XML グラフィックスのための規格である SVG(Scalable Vector Graphics)[3] で表現したクリップアートの形状の洗練に使用するためである。

†岩手大学, Iwate University

### 2 形状の類似性を用いた曲線洗練法

#### 2.1 既存アルゴリズムとの比較

形状変形のための制約条件に基づいて、制御点の変化量を最小化して曲線を変形するアルゴリズムは、Bartels 等 [1, 2] により提案されている。その研究では、制御点の変化量を独立変数とする線形方程式により、曲線変形のための制約条件を表現している。そして、その独立変数のノルムの最小値を求めて、制御点の変化量とする。

その制約条件について、具体的に説明する。説明のために、 $n$  次のパラメトリック曲線の  $i$  番目のセグメントの  $k$  階導関数  $Q_i^{(k)}(u)$  ( $k \geq 0$ ) を、以下のとおりに表現する。

$$\sum_{j=0}^n V_{i,j} B_{i,j}^{(k)}(u) = Q_i^{(k)}(u) \quad (1)$$

ここで、 $V_{i,j}$  と  $B_{i,j}^{(k)}(u)$  は、それぞれ制御点と基底関数である。(1) 式から、制御点の変化量とパラメータ  $u$  の位置での導関数の変化量の関係を表す制約条件を、以下のように導出できる。

$$\sum_{j=0}^n \Delta V_{i,j} B_{i,j}^{(k)}(u) = \Delta Q_i^{(k)}(u) \quad (2)$$

本研究で提案するアルゴリズムでは、(2) 式の制約条件に加えて、形状的特徴の類似性の基準としてスケーリングが変化する相似性を使用することにより、形状的特徴を維持して曲線を変形可能である。また、大局的に形状制約を実現するために、弱制約条件を使用する。

#### 2.2 相似性制約条件

相似形状の曲率で変形するための制約条件を、以下に示す。ここで、 $Q_i^{(1)}(u)$  と  $Q_i^{(2)}(u)$  は変形前のベクトルである。

1.  $\|Q_i^{(1)}(u)\|$  の変形でのスケーリングと、 $\|Q_i^{(2)}(u)\|$  の変形でのスケーリングが等しい。
2.  $Q_i^{(1)}(u)$  と  $Q_i^{(2)}(u)$  のなす角度と、変形後のそれらのなす角度が等しい。

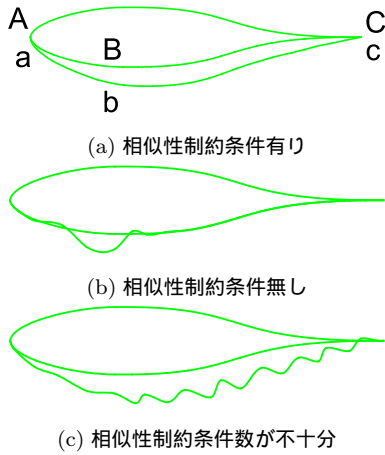


図 1: 相似性制約条件の効果の例

具体的な制約条件を、以下に示す。ここで、 $E_l$  は、 $l$  番目の弱制約条件の誤差である。弱制約条件とは、制約値との誤差が最小である解を求めるための制約条件である。強制制約条件の場合は、この項を含めない。

$$\begin{aligned} & \left\| Q_i^{(2)}(u) \right\|^2 \sum_{j=0}^n Q_i^{(1)}(u) \cdot \Delta V_{i,j} B_{i,j}^{(1)}(u) - \\ & \left\| Q_i^{(1)}(u) \right\|^2 \sum_{j=0}^n Q_i^{(2)}(u) \cdot \Delta V_{i,j} B_{i,j}^{(2)}(u) + \\ & \left\| Q_i^{(1)}(u) \right\|^2 \left\| Q_i^{(2)}(u) \right\|^2 E_l = 0 \end{aligned} \quad (3)$$

$$\begin{aligned} & \left\| Q_i^{(2)}(u) \right\|^2 \sum_{j=0}^n \text{sign}(Q_i^{(1)}(u) \times \Delta V_{i,j} B_{i,j}^{(1)}(u)) \\ & \left\| Q_i^{(1)}(u) \times \Delta V_{i,j} B_{i,j}^{(1)}(u) \right\| - \\ & \left\| Q_i^{(1)}(u) \right\|^2 \sum_{j=0}^n \text{sign}(Q_i^{(2)}(u) \times \Delta V_{i,j} B_{i,j}^{(2)}(u)) \\ & \left\| Q_i^{(2)}(u) \times \Delta V_{i,j} B_{i,j}^{(2)}(u) \right\| + \\ & \left\| Q_i^{(1)}(u) \right\|^2 \left\| Q_i^{(2)}(u) \right\|^2 E_l = 0 \end{aligned} \quad (4)$$

### 2.3 曲線洗練アルゴリズム

洗練する曲線に対して、(2) 式の導関数値と、(3,4) 式の相似性に関する制約条件を作成する。その制約条件の線形連立方程式の独立変数  $\Delta V_{i,j}$  と  $E_l$  のノルムの最小値を持つ  $\Delta V_{i,j}$  を、曲線制御点の最適変化量とする。

## 3 実装と実行結果

### 3.1 実装

Java 言語を使用して実装した。使用する Bezier 曲線を、SVG で表現して変形形状を確認した。

### 3.2 相似性制約条件の効果

相似性制約条件による変形形状の違いの例を、図 1 に示す。曲線 abc は、3 次 Bezier 曲線 (20 セグメン



図 2: 4 点の移動量を制約による形状洗練の例

表 1: 形状洗練での曲率変化の結果

位置	A(a)	B(b)	D(d)	C(c)
変形前の曲率	0.222	0.009	-0.005	0.014
(変形後の曲率)	0.191	0.013	-0.008	0.015

ト) により構成される曲線 ABC の変形形状である。点 A, 点 B, 点 C の移動制約は、それぞれ  $(\Delta x, \Delta y) = (0, 0), (0, -10), (0, 0)$  である。セグメントの接続制約は、導関数の変化が  $C^2$  接続である。形状相似性を制約している位置は、各セグメントについて、図 1(a) ではパラメータ  $u = 0$  と  $0.5$  の位置であり、図 1(c) では  $u = 0.5$  の位置である。

### 3.3 形状洗練での曲率の変化

図 2 の曲線 abdc は、3 次 Bezier 曲線 (20 セグメント) により構成される曲線 ABDC の変形形状である。図 1(a) で使用している制約条件に、点 D の移動制約  $(\Delta x, \Delta y) = (0, -2)$  を加えて形状を洗練している。表 1 に、変形前後での曲率の変化を示す。これより、各位置での曲率の変化が小さく、曲率の符号に変化がないことにより、形状の類似性が保たれていることが分かる。

## 4 おわりに

曲線形状を洗練するためのアルゴリズムをとって、大域的な形状の類似性を維持して変形するために、スケーリングが変化する相似形状の曲率を維持するための弱制約条件を提案した。今後の課題には、相似性からの差を制御することにより、多様な変形形状を得るための研究を上げることができる。

## 参考文献

- [1] Bartels, R. and Forsey, D.: Constraint Based Curve Manipulation, *Tutorial Notes: Splines in Computer Graphics prepared for Eurographics '94*, pp. 31–36 (1994).
- [2] Bartels, R. H. and Beatty, J. C.: A Technique for the Direct Manipulation of Spline Curves, *Graphics Interface 89*, pp. 33–39 (1989).
- [3] W3C: *Scalable Vector Graphics (SVG) 1.1 Specification*, <http://www.w3.org/> (2003).