

導関数ベクトルの非均一相似性制約に基づく 曲線洗練化法

佐藤 信^{†1} 三輪 譲 二^{†1}

曲線を形状類似性を維持したままで、直感的な操作により形状を洗練化する手法を提案する。この手法では、曲線の制約点での2つの導関数ベクトルについて、制約点ごとに非均一な倍率と回転で相似変換した形状との差を最適化により調整することにより、Bezier 曲線を形状洗練化する。このとき、洗練化の特性に合わせて選択した導関数ベクトルのみを変数とするように Bezier 曲線の表現形式を変換し、その変換後の表現変数を最適化することで形状を洗練化する。形状類似性維持のための制約条件としては、曲率の類似性を維持するための制約条件と曲率の変化の類似性を維持するための制約条件を用いる。適用分野としては、曲線形状を直感的な操作により対話的に洗練化する分野がある。特に、クリップアート、挿絵、説明図またはスケッチなどのグラフィックス・コンテンツの形状洗練化のためのインタフェースとして適した手法である。

A Curve Refinement Method Using Non-uniform Similarity of Derivative Vectors

MAKOTO SATOH^{†1} and JOUJI MIWA^{†1}

This paper presents a method for refining Bezier curves intuitively while keeping the similarity in shape. In the method, the criteria to optimize the shape deformation are the similarities of two derivative vectors at each constraining position. The scaling factors and the rotations of the similarity at each position of curves non-uniformly result from optimizing the constraints for curve refinement. The Bezier representation is converted into the representation using only the derivative vectors that are suitably selected for refinement. The shape refinement is achieved by optimizing these converted representation variables. The constraints for similarity of curvature or its variation are employed to keep the similarity in shape. The method is suitable for interactive curve refinement by intuitive manipulation: e.g. interfaces for shape refinement of clip arts, illustrations or sketches.

1. はじめに

本稿では、曲線の導関数ベクトルの非均一相似性制約を用いることにより、曲線をその形状類似性を維持したままで、直感的な操作により形状を洗練化する手法を提案する。ここで洗練化 (refinement) の意味については、Bartels 等により文献 5) に説明がある。ここでも、同様の意味で洗練化という用語を使用する。具体的には、曲線の初期形状を変形して、デザイナーの意図する目的形状を得ることが、曲線の洗練化である。

提案手法では、曲線の制約点での2つの導関数ベクトルについて、制約点ごとに非均一な倍率と回転で相似変換した形状との差を最適化により調整することにより、Bezier 曲線を形状洗練化する。そのために、洗練化の特性に合わせて選択した導関数ベクトルのみを変数とするように Bezier 曲線の表現形式を変換し、その変換後の表現変数を最適化することで形状を洗練化する。この手法では、対話型システムに使用することを目的とするために、最適化計算に用いる制約条件として線形制約条件を用いる。これらの制約条件は、曲線を変形するための制約条件と、曲線の形状類似性を維持するための制約条件により構成される。形状類似性維持のための制約条件としては、曲率の類似性を維持するための制約条件と曲率の変化の類似性を維持するための制約条件の2つの制約条件を用いる。

提案手法を対話的曲線形状洗練化に用いると、Bezier 曲線を変形前の形状的特徴を維持したままで変形可能であることから、直感的な操作による曲線形状洗練化が可能である。また、曲線作成過程での洗練化だけでなく、既に作成してあるクリップアートなどの曲線図形をデータベースから検索して、その目的に合わせて洗練化して使用する機会も増えている。特に、グラフィックス・コンテンツ作成の専門家だけでなく一般のコンピュータ・ユーザも、既に作成してある曲線図形を洗練化する機会が増加していることを考えると、曲線形状を直感的に洗練するためのアルゴリズムの必要性が増加しているといえる。

この手法の適用分野としては、曲線形状を直感的な操作により対話的に洗練化する分野がある。例としては、音響フィルタのイコライザ特性の操作、モーションパスの形状作成またはクリップアート、挿絵、説明図またはスケッチなどのグラフィックス・コンテンツ作成のための対話的なインタフェースがある。

^{†1} 岩手大学
Iwate University

2. 非均一相似性制約を用いた曲線洗練化法

2.1 提案アルゴリズムと既存アルゴリズムとの比較

既に作成してある曲線の形状を洗練化するための手法に関する研究としては、曲線の形状を変更するための制約条件を与えて形状を洗練化する Bartels 等^{1),2)} の研究がある。その研究では、曲線上の点の移動量または導関数値を制約条件として、曲線制御点の変化量を最適化することで曲線形状を洗練化する。そして、制約点について非均一な相似形状の曲率を基準として、形状類似性を維持しながら曲線形状を洗練化するように Bartels 等^{1),2)} の手法を拡張した佐藤等^{6),7)} の手法がある。本稿での提案手法は、佐藤等^{6),7)} の手法を改良したものである。その改良点は、形状類似性として着目している特徴のみを最適化変数とするように曲線表現形式を変換する点、および、形状類似性の基準として、制約点について非均一な相似形状の曲率または曲率変化を用いる点である。

形状洗練化での最適化計算に用いる制約点を、張力と曲げを用いた曲線形状洗練化でのハンドルとして用いる研究としては、Welch 等⁸⁾ の手法がある。本稿での提案手法では、曲線を定義するの制御点のみを用いて、変形前の形状の特徴を維持しながら、曲線の通過点のみを用いて曲線形状を変形可能であるという特色を持つ。

変更後の曲線の形状を与えて形状を洗練化する研究としては Baudel 等³⁾ と Fleisch 等⁴⁾ の手法がある。これらの手法では、変更する部分の変更後の形状をストロークにより与えて、その与えた形状の両端と曲線の変更しない部分をアルゴリズムにより接続する。この手法では、変更する形状を自由に与えることが可能である。一方では、曲線の一部分をある条件に基づいて変更し、その周辺の形状の変更は最小限ならば良い場合でも、変更する部分の全体の形状を与えなければならない。本稿での提案手法では、変形前の形状の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ので、アルゴリズムへのユーザ入力として変更のための制約条件のみを与えればよいという特色を持つ。

2.2 曲線洗練化アルゴリズムの概要

提案するアルゴリズムでは、平面上の Bezier 曲線を対象として、曲線の形状特徴制約点での 2 つの導関数ベクトルの始点を一致させてできる形状を相似変換した形状との差を変数として、その変数の値を最適化することにより、形状の類似性を維持したままで、形状変形のための制約条件を満たすように、Bezier 曲線を洗練化する。ここで、この表現形式変換後の変数を形状類似性変数と定義する。以下に、アルゴリズムの概要を示す。

Step 1 曲線の表現形式を、Bezier 制御点による表現から、形状類似性変数による表現に

変換する。

Step 2 形状類似性変数を変数として、形状変形のための線形制約条件を作成する。この制約条件のもとで、形状類似性変数の値を最小化する。

Step 3 最小化した形状類似性変数の値から、Bezier 制御点による曲線表現に逆変換する。以下で、各段階について詳しく説明する。

2.3 曲率の類似性に基づく Bezier 曲線の表現形式変換

ここで、説明のために、 n 次の Bezier 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、 u は曲線パラメータである。また、 $V_{i,j}$ と $B_{i,j}^{(k)}(u)$ は、それぞれ Bezier 制御点と基底関数である。これにより、(1) 式から、曲線通過点 $P_i(u)$ は、以下のとおりである。

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

また、1 階導関数と 2 階導関数を用いた形状類似性変数 $S_{(12)_i}(u) = (D_{(12)_i}(u), C_{(12)_i}(u))$ を、以下のとおりに表現する。ここで、 $Q_i^{(1)}(u)$ および $Q_i^{(2)}(u)$ は変形前の曲線の導関数ベクトルである。例えば、 $S_{(12)_i}(u) = (0, 0)$ の場合には、その位置での曲率変化は、相似形状の変形での曲率変化である。

$$\frac{\sum_{j=0}^n Q_i^{(1)}(u) \cdot V_{i,j} B_{i,j}^{(1)}(u)}{\|Q_i^{(1)}(u)\|^2} - \frac{\sum_{j=0}^n Q_i^{(2)}(u) \cdot V_{i,j} B_{i,j}^{(2)}(u)}{\|Q_i^{(2)}(u)\|^2} = D_{(12)_i}(u) \quad (3)$$

$$\frac{\sum_{j=0}^n \text{sign}(Q_i^{(1)}(u) \times V_{i,j} B_{i,j}^{(1)}(u)) \|Q_i^{(1)}(u) \times V_{i,j} B_{i,j}^{(1)}(u)\|}{\|Q_i^{(1)}(u)\|^2} - \frac{\sum_{j=0}^n \text{sign}(Q_i^{(2)}(u) \times V_{i,j} B_{i,j}^{(2)}(u)) \|Q_i^{(2)}(u) \times V_{i,j} B_{i,j}^{(2)}(u)\|}{\|Q_i^{(2)}(u)\|^2} = C_{(12)_i}(u) \quad (4)$$

曲線の表現で使用するために、通過制御点 $P_i(u)$ および形状類似性制御点での $S_{(12)_i}(u)$ に関して、それぞれ (2) 式および (3, 4) 式を、表現する曲線の自由度に合わせて、一意に形状

を決定可能な個数を用意し、それを以下のとおりに行列表現する．

$$U_{(12)} = M_{(12)} V \quad (5)$$

ここで、 V は $V_{i,j}$ を要素とするベクトル、 $M_{(12)}$ は $B_{i,j}^{(k)}(u)$ または (3,4) 式で $B_{i,j}^{(k)}(u)$ に定数を演算した値を要素とする行列、そして $U_{(12)}$ は $P_i(u)$ および $S_{(12)_i}(u)$ を要素とするベクトルである．この (5) 式から、以下の曲線表現形式変換行列 $M_{(12)}^{-1}$ を求める．

$$V = M_{(12)}^{-1} U_{(12)} \quad (6)$$

2.4 曲率変化の類似性に基づく Bezier 曲線の表現形式変換

曲率変化の類似性に基づいた表現形式変換では、形状類似性変数 $S_{(23)_i}(u) = (D_{(23)_i}(u), C_{(23)_i}(u))$ を、以下のとおりに表現する．ここで、 $Q_i^{(1)}(u)$ および $Q_i^{(2)}(u)$ は変形前の曲線の導関数ベクトルである．例えば、 $S_{(23)_i}(u) = (0, 0)$ の場合には、その曲線の 1 階導関数ベクトルの始点を一致させた場合にそのベクトルの終点の軌跡により作成される曲線 H について、その位置での曲率変化は、曲線 H を相似変換した曲率変化である．

$$\frac{\sum_{j=0}^n Q_i^{(2)}(u) \cdot V_{i,j} B_{i,j}^{(2)}(u)}{\|Q_i^{(2)}(u)\|^2} - \frac{\sum_{j=0}^n Q_i^{(3)}(u) \cdot V_{i,j} B_{i,j}^{(3)}(u)}{\|Q_i^{(3)}(u)\|^2} = D_{(23)_i}(u) \quad (7)$$

$$\frac{\sum_{j=0}^n \text{sign}(Q_i^{(2)}(u) \times V_{i,j} B_{i,j}^{(2)}(u)) \|Q_i^{(2)}(u) \times V_{i,j} B_{i,j}^{(2)}(u)\|}{\|Q_i^{(2)}(u)\|^2} - \frac{\sum_{j=0}^n \text{sign}(Q_i^{(3)}(u) \times V_{i,j} B_{i,j}^{(3)}(u)) \|Q_i^{(3)}(u) \times V_{i,j} B_{i,j}^{(3)}(u)\|}{\|Q_i^{(3)}(u)\|^2} = C_{(23)_i}(u) \quad (8)$$

曲線の表現に用いるために、通過制御点 $P_i(u)$ および形状類似性制御点での $S_{(23)_i}(u)$ に関して、それぞれ (2) 式および (7, 8) 式を、表現する曲線の自由度に合わせて、一意に形状を決定可能な個数を用意し、それを以下のとおりに行列表現する．

$$U_{(23)} = M_{(23)} V \quad (9)$$

ここで、 V は $V_{i,j}$ を要素とするベクトル、 $M_{(23)}$ は $B_{i,j}^{(k)}(u)$ または (7,8) 式で $B_{i,j}^{(k)}(u)$ に定数を演算した値を要素とする行列、そして $U_{(23)}$ は $P_i(u)$ および $S_{(23)_i}(u)$ を要素とするベクトルである．この (9) 式から、以下の曲線表現形式変換行列 $M_{(23)}^{-1}$ を求める．

$$V = M_{(23)}^{-1} U_{(23)} \quad (10)$$

2.5 曲線が複数セグメントにより構成される場合

曲線が複数セグメントにより構成される場合には、 C^1 接続、 C^2 接続または接続形状の類似性を維持した接続⁷⁾ などの接続条件にあわせて、接続に関する条件を追加して用いる． C^1 接続、 C^2 接続の場合には、以下の式を用いる．なお、1 セグメントの曲線パラメータ u は $0.0 \leq u \leq 1.0$ としている．

$$\sum_{j=0}^n V_{i,j} B_{i,j}^{(1)}(1.0) - \sum_{j=0}^n V_{i+1,j} B_{i+1,j}^{(1)}(0.0) = 0 \quad (11)$$

$$\sum_{j=0}^n V_{i,j} B_{i,j}^{(2)}(1.0) - \sum_{j=0}^n V_{i+1,j} B_{i+1,j}^{(2)}(0.0) = 0 \quad (12)$$

これにあわせて、2.3, 2.4 節で、曲線接続条件にあわせて、曲線表現形式変換行列のための連立方程式に (11) 式または (12) 式を含める必要がある．

2.6 曲率の類似性に基づく Bezie 曲線洗練化アルゴリズム

2.3 節で説明した曲線表現形式変換行列 $M_{(12)}^{-1}$ が計算してあるとして、アルゴリズムを説明する．

Step 1 (2) 式および (6) 式を用いて、 $U_{(12)}$ を独立変数として通過制御点に関する制約条件を作成する．また、必要な場合には、 $U_{(12)}$ を独立変数として曲線接続に関する制約条件を作成する．

Step 2 追加の形状類似性制御点が必要な場合は、(3, 4) 式の $S_{(12)_d}$ または $S_{(12)_e}$ を左辺に移項した方程式を、(6) 式により変換することにより、 $U_{(12)}$ 、 $S_{(12)_d}$ そして $S_{(12)_e}$ を独立変数として形状類似性に関する制約条件を作成する．この制約条件では、 $S_{(12)_d}$ および $S_{(12)_e}$ が弱制約項となる．

Step 3 文献 7) で述べている制約条件および (6) 式を用いて、 $U_{(12)}$ を独立変数として形状を変形するための制約条件を作成する．

Step 4 作成した制約条件の線形連立方程式を、独立変数 $U_{(12)}$ と追加の制約条件の弱制約項のノルムが最小値を持つように解く．

Step 5 (6) 式に $U_{(12)}$ を代入して、ベジェ曲線の制御点を求める．

2.7 曲率変化の類似性に基づく Bezie 曲線洗練化アルゴリズム

2.4 節で説明した曲線表現形式変換行列 $M_{(23)}^{-1}$ が計算してあるとして、アルゴリズムを説明する．

Step 1 (2) 式および (10) 式を用いて, $U_{(23)}$ を独立変数として通過制御点に関する制約条件を作成する. また, 必要な場合には, $U_{(23)}$ を独立変数として曲線接線に関する制約条件を作成する.

Step 2 追加の形状類似性制御点が必要な場合は, (7, 8) 式の $S_{(23)_d}$ または $S_{(23)_c}$ を左辺に移項した方程式を, (10) 式により変換することにより, $U_{(23)}, S_{(23)_d}$ および $S_{(23)_c}$ を独立変数として形状類似性に関する制約条件を作成する. この制約条件では, $S_{(23)_d}$ および $S_{(23)_c}$ が弱制約項となる.

Step 3 文献 7) で述べている制約条件および (10) 式を用いて, $U_{(23)}$ を独立変数として形状を変形するための制約条件を作成する.

Step 4 作成した制約条件の線形連立方程式を, 独立変数 $U_{(23)}$ と追加の制約条件の弱制約項のノルムが最小値を持つように解く.

Step 5 (10) 式に $U_{(23)}$ を代入して, ベジエ曲線の制御点を求める.

2.8 曲線形状から直線形状への形状洗練化

2.4, 2.7 節において説明した, 曲率変化の類似性に基づく形状洗練化では, 2 階導関数ベクトルと 3 階導関数ベクトルの始点を一致させた形状が, 非均一な相似形状に変化するという制約条件を使用している. また, 1 階導関数ベクトルに関する制約が必要でない場合には, Bezier 曲線の表現形式変換行列および最適化計算で用いる制約条件に, 1 階導関数を含める必要がない. これにより, 曲線の形状類似性を維持したままで, 徐々に直線形状に近づくように形状洗練化が可能であり, 直線に形状洗練化することも可能である.

3. 実装と結果の検討

3.1 実装

提案アルゴリズムを, Java 言語を使用して実装した. 形状の洗練化を確認するための曲線には, Bezier 曲線を使用した. その Bezier 曲線のグラフィックスデータとしての表現形式には, SVG を使用した. SVG を取り扱い可能なドローイング・ツールを使用して, 洗練化前後の曲線の变形形状を確認した.

3.2 曲率の類似性に基づく Bezie 曲線洗練化の例

曲率の類似性に基づく形状洗練化の例を, 図 1, 2 に示す. Bezier 曲線 1 セグメントで表現した曲線 AB の形状を基にして, それと類似形状の曲線 AC に形状洗練化している. ここで, ×印は, 曲線が通過する座標を制約する点である. これらの洗練化前後の形状を比較すると, 曲線形状の類似性を維持したままで, 制約条件にあわせて形状を洗練化可能なこと

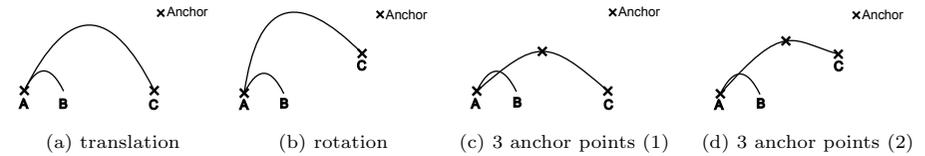


図 1 曲率の類似性に基づく Bezie 曲線洗練化の例 (対称形状, 1 セグメント)

Fig. 1 Examples of curve refinement using curvature similarity (symmetry, 1 segment).

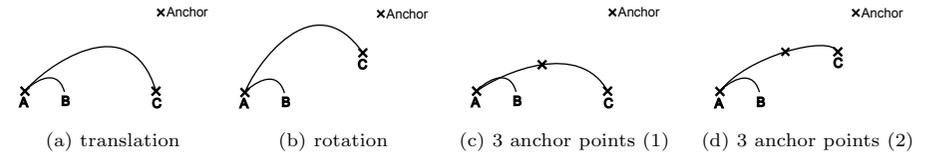


図 2 曲率の類似性に基づく Bezie 曲線洗練化の例 (非対称形状, 1 セグメント)

Fig. 2 Examples of curve refinement using curvature similarity (asymmetry, 1 segment).

が分かる. また, 図 1, 2(a,b) から, 相似形状での形状洗練化が可能な形状洗練化制約条件の場合には, 相似形状を維持して形状洗練化が可能なが分かる. これは, 最適化計算において曲線の Bezier 制御点の移動量を最適化変数としないように, 曲線の表現形式を変換していることによる.

曲線セグメントの接続がある場合について, 曲率の類似性に基づく形状洗練化の例を, 図 3, 4 に示す. Bezier 曲線 2 セグメントで表現した曲線 AB の形状を基にして, それと形状が類似な曲線 AC に形状洗練化している. ここで, ×印は, 曲線が通過する座標を制約する点である. また, 形状洗練化で用いた制約条件を, 表 1 に示す. これらの洗練化前後の形状を比較すると, 曲線セグメントの接続がある場合にも 1 セグメントの場合と同様に, 曲線形状の類似性を維持したままで, 制約条件にあわせて形状を洗練化可能なことが分かる.

3.3 曲率変化の類似性に基づく Bezie 曲線洗練化の例

曲率変化の類似性に基づく形状洗練化の例を, 図 5, 6 に示す. Bezier 曲線 2 セグメントで表現した曲線 AB の形状を基にして, それと類似形状の曲線 AC に形状洗練化している. ここで, ×印は, 曲線が通過する座標を制約する点である. また, 形状洗練化で用いた制約条件を, 表 2 に示す. これらの洗練化前後の形状を比較すると, 曲線形状の類似性を維持したままで, 制約条件にあわせて形状を洗練化可能なことが分かる. また, 図 5(a,b,c), 6(a,b,c) から, 曲線形状から直線形状に近づくように形状洗練化が可能なが分かる. 特に, 図 5(c), 6(c) から, 曲線形状から, その曲線と曲率符号が反対である同様の曲率変化の

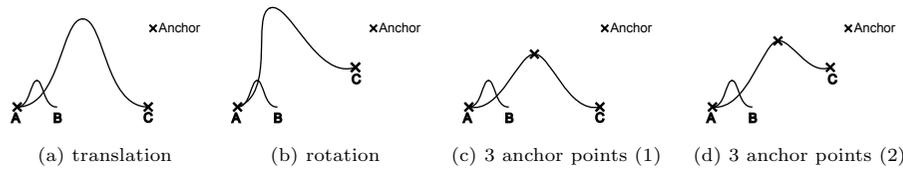


図3 曲率の類似性に基づく Bezie 曲線洗練化の例 (対称形状, 2 セグメント)

Fig. 3 Examples of curve refinement using curvature similarity (symmetry, 2 segments).

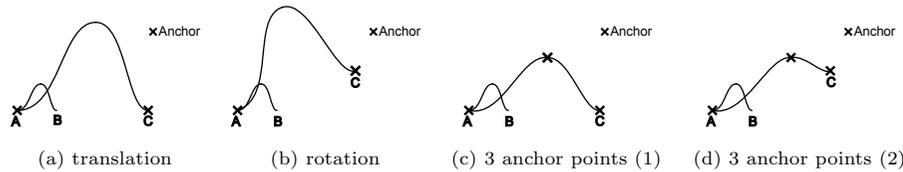


図4 曲率の類似性に基づく Bezie 曲線洗練化の例 (非対称形状, 2 セグメント)

Fig. 4 Examples of curve refinement using curvature similarity (asymmetry, 2 segments).

曲線に形状洗練化が可能なが分かる。

形状類似性基準による洗練化形状の相違の例を、図7に示す。Bezier 曲線 2 セグメントで表現した曲線 AB の形状を基にして、それと類似形状の曲線 ADC, 曲線 AEC または曲線 AFC に形状洗練化している。ここで、x印は、曲線が通過する座標を制約する点である。

表1 図3, 4における曲率の類似性に基づく Bezie 曲線洗練化の例で用いた方程式 (2 セグメント)

Table 1 The equations used in the examples of curve refinement using curvature similarity in Fig.3, 4 (2 segments).

stages	equations
rerepresentation (section 2.3)	positional constraint ($i = 0, u = 0.0$) Eq.(2), positional constraint ($i = 1, u = 1.0$) Eq.(2), similarity constraints with 1st and 2nd derivative ($i = 0, u = \frac{1}{3}$) Eq.(3,4), similarity constraints with 1st and 2nd derivative ($i = 0, u = \frac{2}{3}$) Eq.(3,4), similarity constraints with 1st and 2nd derivative ($i = 1, u = \frac{1}{3}$) Eq.(3,4), similarity constraints with 1st and 2nd derivative ($i = 1, u = \frac{2}{3}$) Eq.(3,4), similarity constraints with 1st derivative of segment connection ($i = 0, u = 1.0$ and $i = 1, u = 0.0$)
optimization (section 2.6)	positional constraint A ($i = 0, u = 0.0$) Eq.(2), positional constraint C ($i = 1, u = 1.0$) Eq.(2), additional positional constraint in Fig.3 (c,d) and Fig.4 (c,d) ($i = 0, u = 1.0$) Eq.(2)

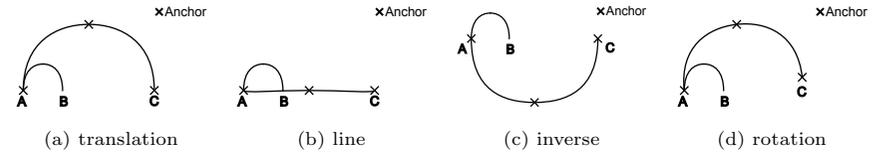


図5 曲率変化の類似性に基づく Bezie 曲線洗練化の例 1(2 セグメント)

Fig. 5 Examples of curve refinement using curvature variation similarity (2 segments).

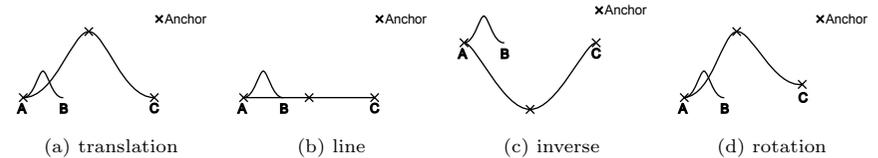


図6 曲率変化の類似性に基づく Bezie 曲線洗練化の例 2(2 セグメント)

Fig. 6 Examples of curve refinement using curvature variation similarity (2 segments).

表2 図5, 6における曲率変化の類似性に基づく Bezie 曲線洗練化の例で用いた方程式 (2 セグメント)

Table 2 The equations used in the examples of curve refinement using curvature variation similarity in Fig.5, 6 (2 segments)..

stages	equations
rerepresentation (section 2.4)	positional constraint ($i = 0, u = 0.0$) Eq.(2), positional constraint ($i = 0, u = 1.0$) Eq.(2), positional constraint ($i = 1, u = 1.0$) Eq.(2), similarity constraints with 2nd and 3rd derivative ($i = 0, u = 0.5$) Eq.(7,8), similarity constraints with 2nd and 3rd derivative ($i = 1, u = 0.5$) Eq.(7,8), C^2 continuity of segment connection ($i = 0, u = 1.0$ and $i = 1, u = 0.0$) Eq.(11,12)
optimization (section 2.7)	positional constraint A ($i = 0, u = 0.0$) Eq.(2), positional constraint ($i = 0, u = 1.0$) Eq.(2), positional constraint C ($i = 1, u = 1.0$) Eq.(2), C^2 continuity of segment connection ($i = 0, u = 1.0$ and $i = 1, u = 0.0$) Eq.(11,12), tangent constraint ($i = 0, u = 0.0$ and $i = 1, u = 1.0$)

これらの洗練化前後の形状を比較すると、1階導関数ベクトルと2階導関数ベクトルを用いた場合には、制約点での相似形状の曲率を維持しているのに対して、2階導関数ベクトルと3階導関数ベクトルを用いた場合には、1階導関数ベクトルを制約していないことにより、曲線の両端を結ぶ直線、またはその直線に対して反転した形状と同様の曲率変化の形状に洗

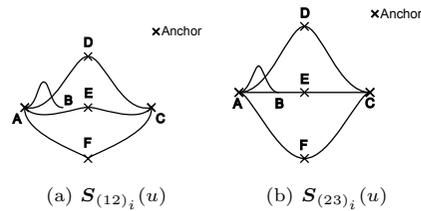


図 7 形状類似性基準の比較

Fig. 7 comparison of shape similarity criteria.

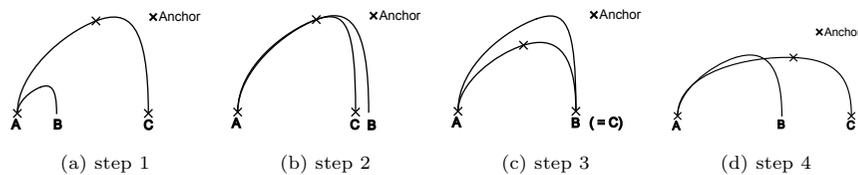


図 8 曲線洗練化を繰り返す例

Fig. 8 Examples of iterative curve refinement

練化可能なことが分かる。

3.4 曲線洗練化を繰り返す例

曲率変化の類似性に基づく形状洗練化を繰り返す例を、図 8 に示す。図 8(a) では、Bezier 曲線 2 セグメントで表現した曲線 AB の形状を基にして、それと類似形状の曲線 AC に形状洗練化している。図 8(b,c,d) では、それぞれの 1 つ前のステップでの洗練化形状を基にして、形状を洗練化している。これらから、各ステップで、1 つ前のステップでの洗練化形状との類似性を維持しながら、形状洗練化のための制約点を通して形状洗練化を繰り返すことが可能であることが分かる。

4. おわりに

曲線の導関数ベクトルの非均一相似性制約を用いることにより、その形状類似性を維持したままで、直感的な操作により曲線形状を洗練化する手法を提案した。提案手法の特徴は、曲線の制約点での 2 つの導関数ベクトルについて、制約点ごとに非均一な倍率と回転で相似変換した形状との差を最適化により調整することにより、Bezier 曲線を形状洗練化することである。そのために、洗練化の特性に合わせて選択した導関数ベクトルを基にして Bezier 曲線の表現形式を変換し、その変換後の表現変数について選択した導関数ベクトル

のみを変数とするようにして最適化計算をすることで形状を洗練化した。これにより、曲率の類似性または曲率変化の類似性を維持することが可能となった。また、曲線形状から直線形状に徐々に近づくように形状洗練化することが可能となった。提案手法を、グラフィックス・コンテンツ作成のためのインタフェースに用いることにより、専門家だけでなく一般のコンピュータ・ユーザが、クリップアート、挿絵、説明図またはスケッチなどのグラフィックス・コンテンツなどの既に作成してある曲線図形を、直感的な操作により洗練化することが可能となった。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挙げることができる。

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) Baudel, T.: A mark-based interaction paradigm for free-hand drawing, *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, New York, NY, USA, ACM, pp.185-192 (1994).
- 4) Fleisch, T., Rechel, F., Santos, P. and Stork, A.: Constraint Stroke-Based Oversketching for 3D Curves, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp.161-165 (2004).
- 5) Forsey, D.R. and Bartels, R.H.: Hierarchical B-spline refinement, *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, ACM, pp.205-212 (1988).
- 6) 佐藤 信, 三輪譲二: 形状の類似性を用いた曲線洗練法, 第 8 回情報科学技術フォーラム講演論文集第 3 分冊, pp.289-290 (2009).
- 7) 佐藤 信, 三輪譲二: 接続の類似性制約を用いた曲線洗練化法, 情報処理学会研究報告, Vol.2009-CG-137, No.7, pp.1-6 (2009).
- 8) Welch, W. and Witkin, A.: Variational surface modeling, *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, ACM, pp.157-166 (1992).