

近傍制御点を用いた Bezier 曲線洗練化のための ユーザインタフェースの設計

佐藤 信^{†1} 三輪 譲 二^{†1}

曲線形状の概要を示す制約条件に基づいた形状洗練化のために、Bezier 曲線の形状操作のユーザインタフェースを提案する。この方式では、通過制御点と近傍制御点について、最適化計算により曲線を変換する。これにより、制御点操作による形状変化から、直感的に制御点の操作方法を理解可能である。この直感性により、専門家だけでなく一般のコンピュータユーザが、曲線図形を操作するインタフェースとして適した方式である。

Design of User Interface for Refinement of Bezier Curves with Neighbouring Control Points

MAKOTO SATOH^{†1} and JOUJI MIWA^{†1}

This paper presents a new interaction method for manipulating Bezier curves in order to refine the shape using the constraints for the simple outline of the shape. In the method, the criteria to optimize the shape are the distance from the anchor or neighbouring control points to the curve. Therefore the simple relationship between the position of control points and the shape of curves enable the user to manipulate curves intuitively. The method is suitable not only for the experts but also for the average users of the graphics applications.

1. はじめに

本稿では、曲線形状を直感的に作成または洗練化するために、通過制御点と近傍制御点を

用いた曲線操作のためのユーザインタフェースを提案する。ここでの洗練化 (refinement) の意味は、Bartels 等により文献 7) において説明されているものと同様である。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。また、通過制御点とは、その点を曲線が通過すべき制御点であり、近傍制御点とは、その点の可能な限り近くを曲線が通過すべき制御点である。

ここで始めに、提案するユーザインタフェースを用いて形状を操作する曲線について考える。この曲線は、Bezier 曲線の表現形式を変換したものであり、通過制御点と近傍制御点を用いて曲線を表現する。この表現形式を用いた曲線の特徴を、以下に示す。

形状操作の直感性 目的とする曲線の通過点または近傍点を指定するという直感的な操作により、曲線形状を作成または洗練化することが可能である。

形状概要の表現性 曲線の通過点または近傍点を用いて目的形状の概要を表現することが可能である。例えば、目的形状を明確に決定していない段階であっても、形状概要の近傍点を指定して形状を作成し、その後、曲線の通過点または近傍点を変更または追加することにより、形状を洗練化することが可能である。

形状微調整の容易性 近傍点の位置または個数を変更することにより、形状を微調整することによる洗練化が可能である。特に、近傍点の個数は、曲線表現の数学的自由度に制限されないため、目的形状に合わせて近傍点の個数を増減することが可能である。

提案するユーザインタフェースを用いるユーザと、その場合の典型的な曲線図形の作成方法を以下のとおり想定する。

ユーザ 職業として曲線図形を作成する方、または一般のコンピュータユーザ
作画方式 初めに形状の概要を作成して、その後、形状洗練化を繰り返すことにより目的形状を得る作画方式

このユーザと典型的な作画方式に合わせて、以下の条件を満たしたユーザインタフェースを設計する。なお、一般のコンピュータユーザを、このユーザインタフェースのユーザに含めるので、使用する入力装置は、マウス、タッチパッドまたはペンタブレットとする。

- 曲線形状の概要を示す点列を入力することにより、その点列に含まれる点を通過制御点または近傍制御点とする曲線を作成する。
- 通過制御点または近傍制御点の座標または個数を変化することにより、曲線形状を洗練化する。
- 曲線形状を微調整して洗練化することを可能とする。特に、入力装置の座標入力解像度が、洗練化の精度を直接的に制限しないようにする。

^{†1} 岩手大学
Iwate University

このインタフェースを用いることにより、直感的な操作で曲線図形を作成または洗練化することが可能となる。

ここで、関連研究との比較をする。点列からパラメトリック曲線を作成するための手法に関する研究としては、Plass 等⁹⁾の研究があり、変更後の曲線の形状をストローク入力により与えて曲線形状を洗練化するための手法に関する研究としては、Baudel 等³⁾とFleisch 等⁶⁾の研究がある。また、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、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適化して曲線を変形するアルゴリズムを、Bartels 等^{1),2)}が提案している。そして、曲面と近傍点との距離、そして曲面の制御点の変化量を最適化することにより曲面を洗練化する手法を、Welch 等¹⁰⁾が提案している。

これらの研究に対して、提案する手法の特徴は、以下のとおりである。

- 座標点を入力することにより、曲線形状を作成または洗練化する。これは、ストロークにより点列を入力することに比較すると、個々の点を1個づつ入力することは、希望する座標値を入力することが容易であることによる。
- 近傍制御点と曲線の距離のみを最小化することにより、曲線形状を作成または洗練化する。このことにより、複数の種類の変数を最適化する場合と比較して最適化の重みを調節する必要がないだけでなく、曲線形状作成のための形状基準の入力インタフェースとして、近傍制御点を使用し易いものとなっている。
- 曲線の変形のための座標値に関する制約としてだけでなく、変形前の曲線形状を維持するための座標値に関する制約としても、近傍制御点を使用する。

提案するユーザインタフェースは、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として適用可能である。例としては、クリップアート、挿絵、説明図の作成または洗練化、または2D曲線を使用した3D曲線の作成^{4),8)}などに使用可能である。

2. 近傍制御点を用いた Bezier 曲線

2.1 曲線作成方式の概要

通過制御点と近傍制御点を制約点として、曲線と近傍制御点との距離を最小化することにより、Bezier 曲線を作成または洗練化する。以下に、その概要を示す。

Step1 曲線の表現形式を、Bezier 制御点による表現から曲線と近傍制御点の距離を用いた表現に変換する。

Step2 通過制御点と近傍制御点を制約点として、曲線と近傍制御点との距離を最小化する。

Step3 曲線の表現形式を、曲線と近傍制御点との距離を用いた表現から Bezier 制御点による表現に変換する。

以下で、各段階について詳しく説明する。

2.2 Bezier 曲線の表現形式の変換

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。

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

次に、曲線の複数の通過点に関して、(2) 式を、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する。それを以下のとおりに表現する。

$$P = MV \quad (3)$$

ここで、 V は $V_{i,j}$ を要素とするベクトル、 M は $B_{i,j}^{(0)}(u)$ を要素とする行列、そして P は $P_i(u)$ を要素とするベクトルである。この (3) 式から、以下の逆行列 M^{-1} を求める。

$$V = M^{-1}P \quad (4)$$

(4) 式の $P_i(u)$ を、近傍制御点 $N_i(u)$ と、近傍制御点と曲線との距離 $\Delta N_i(u)$ の和とする。

$$P_i(u) = N_i(u) + \Delta N_i(u) \quad (5)$$

(4) 式と (5) 式から、次式を得る。

$$V = M^{-1}N + M^{-1}\Delta N \quad (6)$$

ここで、 N は $N_i(u)$ を要素とするベクトルそして ΔN は $\Delta N_i(u)$ を要素とするベクトルである。

2.3 曲線と近傍制御点との距離の最小化を用いた曲線作成

Bezier 曲線の表現形式を変換する (6) 式を用いて、近傍制御点と曲線との距離を最小化する手順を以下に示す。

Step1 (6) 式の行数分の個数の通過制御点と近傍制御点の座標値を、(6) 式の N に代入する。これにより、(6) 式の右辺は、 ΔN のみを変数とする連立方程式となる。

Step2 (2) 式を用いて、通過制御点に関する制約方程式を作成する。この制約方程式を、Step1 で座標値を代入した (6) 式を用いて、 ΔN のみを変数とする制約方程式とする。
Step3 次式を用いて、Step1 で使用していない近傍制御点に関する制約方程式を作成する。

$$\sum_{j=0}^n V_{i,j} B_{i,j}^{(0)}(u) + \Delta N_i(u) = N_i(u) \quad (7)$$

この制約方程式を、Step1 で座標値を代入した (6) 式を用いて、 ΔN のみを変数とする制約方程式とする。

Step4 Step2 と Step3 で作成した制約条件の連立方程式を、独立変数 ΔN のノルムが最小値を持つように解く。

2.4 Bezier 曲線への表現形式の逆変換

2.3 節の Step1 で座標値を代入した (6) 式に、2.3 節の Step4 で求めた ΔN を代入して、曲線の Bezier 制御点を求める。

3. 近傍制御点を用いた Bezier 曲線洗練化のためのユーザインタフェースの設計

3.1 ユーザインタフェース

曲線形状の作成と洗練化をするために、以下に示すモードを設ける。

形状作成モード 曲線の初期形状を作成する。このとき、入力した点列の端点を通過制御点とし、端点以外の点を近傍制御点とする。

形状洗練化モード 曲線形状を洗練化する。通過制御点または近傍制御点の座標値変更などの操作をおこない、それに合わせて曲線を作成する。

漸近的形状洗練化モード 曲線の位置変化が小さいように、曲線を近傍制御点に少しずつ近づける。

3.2 形状作成アルゴリズム

入力した点列 P から、その点列に含まれる点および点列の端点を結ぶ直線形状を基準として、曲線形状を作成するアルゴリズムを以下に示す。このとき、以下の基準を使用することが可能であるようにアルゴリズムを設計した。

- アルゴリズムの全段階で、点列 P に含まれる全ての点を基準とする。
- アルゴリズムの段階に合わせて、基準とする点を点列 P から選択する。

Step1 点列 P の端点を通過制御点 A 、そして端点以外の点を近傍制御点 N とする。

Step2 点列から選択した点を基準とする場合には、Douglas 等⁵⁾ のアルゴリズムを用い

て、点列 P から数点を選択する。これらの点をからなる点列を D とする。

Step3 D の各点と作成する曲線との距離を最適化するために、 D の各点に対応する曲線上の各点の曲線パラメータを決定する。ここでは、点列 P の各点を結ぶ直線の距離を基にして曲線パラメータを作成する。

Step4 D の各点に対応する曲線 (繰り返しの 1 回目では点列 P の端点を結ぶ直線) 上の各点を、近傍制御点 M とする。

Step5 2 節のアルゴリズムを用いて、 D そして M の各点と曲線との距離を最適化することにより、曲線を作成する。このとき、点列 D の端点を通過制御点 A 、そして端点以外の点が近傍制御点 N である。

Step6 P の各点と曲線との距離が、あらかじめ定めた値以下の場合には、Step7 に進む。そうでない場合には、Step4 に進む。

Step7 P の各点と作成する曲線との距離を最適化するために、 P の各点に対応する曲線上の各点の曲線パラメータを決定する。ここでは、点列 P の各点について、その点と曲線との距離が最短である曲線上の点の曲線パラメータを用いる。

Step8 P の各点に対応する曲線 (繰り返しの 1 回目では点列 P の端点を結ぶ直線) 上の各点を、近傍制御点 M とする。

Step9 2 節のアルゴリズムを用いて、 P そして M の各点と曲線との距離を最適化することにより、曲線を作成する。このとき、点列 P の端点を通過制御点 A 、そして端点以外の点が近傍制御点 N である。

Step10 P の各点と曲線との距離が、あらかじめ定めた値以下の場合には、点列 P を基にした曲線が作成出来たので、アルゴリズムを終了する。そうでない場合には、Step7 に進む。

以上のアルゴリズムで作成した曲線が、点列 P を十分近似していない場合には、点列 P に Douglas 等⁵⁾ のアルゴリズムを適用して得られた各線分に対して、上記のアルゴリズムを実行する。このとき、各線分に対応する各曲線セグメントの接続に制約を加える場合には、Bartels 等^{1),2)} と同様の制約条件に 2 節の変数変換を行ったものを、上記のアルゴリズムの最適化計算の制約条件として追加する。

3.3 形状洗練化アルゴリズム

曲線形状の洗練化アルゴリズムを、以下に示す。

Step1 通過制御点または近傍制御点に、以下の操作を行う。

- 制御点の座標値の変更

- 制御点の追加
- 制御点の削除
- 制御点の種類の変更 (その制御点が通過制御点であるのか、または、近傍制御点であるのかを変更)

Step2 各制御点と曲線との距離を最適化するために、各制御点に対応する曲線上の各点の曲線パラメータを決定する。ここでは、各制御点について、その点と曲線との距離が最短である曲線上の点の曲線パラメータを用いる。

Step3 2節のアルゴリズムを用いて、各制御点と曲線との距離を最適化することにより、曲線を作成することにより、曲線形状を洗練化する。

3.4 漸近的形状洗練化アルゴリズム

漸近的に曲線形状を洗練化するアルゴリズムを、以下に示す。

Step1 通過制御点または近傍制御点を入力する。

Step2 形状を維持するための近傍制御点を、曲線上に自動生成する。

Step3 各制御点と曲線との距離を最適化するために、各制御点に対応する曲線上の各点の曲線パラメータを決定する。ここでは、各制御点について、その点と曲線との距離が最短である曲線上の点の曲線パラメータを用いる。

Step4 2節のアルゴリズムを用いて、各制御点と曲線との距離を最適化することにより、曲線を作成する。曲線形状を洗練化する。

入力装置のボタンを押してドラッグを開始することにより、Step1を開始する。その後、入力装置のボタンを離すまで、Step2からStep4までを繰り返す。

4. 実装と結果の検討

4.1 実装

Java 言語を使用して実装した。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変形形状を確認した。使用したのは、3 次の Bezier 曲線 1 セグメントである。

4.2 近傍制御点を用いた曲線作成の例

近傍制御点を用いて曲線を作成する例を、図 1 に示す。ここで、図 1(a-1,a-2) は、入力点列 ABC の 3 点から、点 A, C を通過制御点、点 B を近傍制御点として曲線を作成している。なお、点 D は、点 B に対応するようにアルゴリズムにより作成している近傍制御点である。また、図 1(a-1) の繰り返し回数の 1 回目とは、3.2 節のアルゴリズムの 1 個目の繰り返し

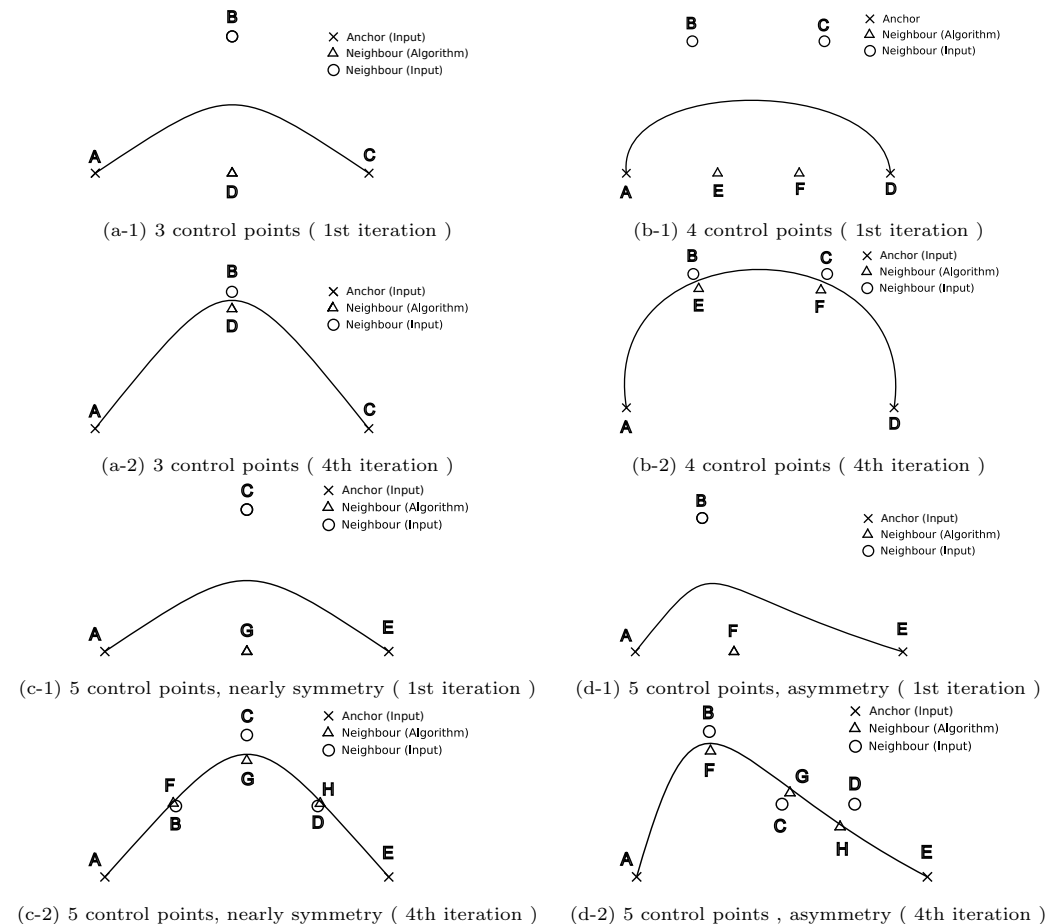


図 1 近傍制御点を用いた曲線作成の例
Fig.1 Examples of curve creation using neighbouring control points.

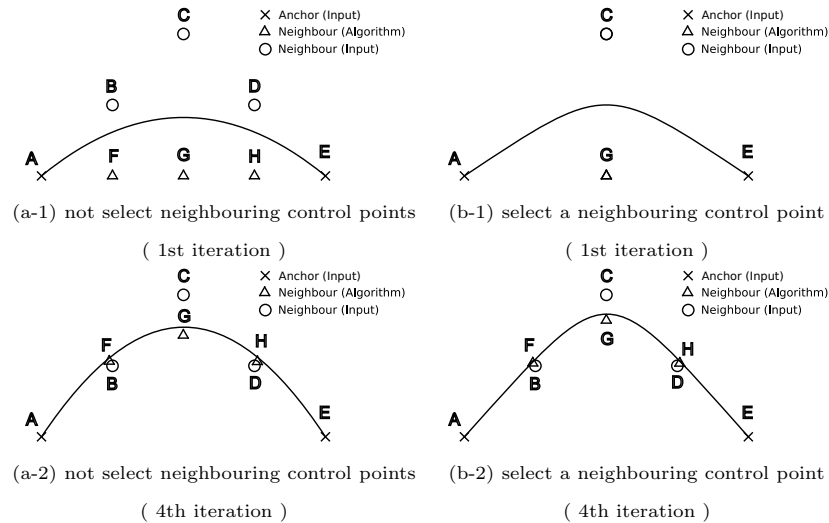


図2 選択的に近傍制御点を用いた効果の例
Fig. 2 Examples of selection of neighbouring control points.

返しの1回目のことであり、図1(a-2)の繰り返し回数の4回目とは、3.2節のアルゴリズムの1回目の繰り返しを2回実行してからの、2回目の繰り返しの2回目のことである(以下同様)。また、図1(b-1,b-2)、(c-1,c-2)そして(d-1,d-2)は、それぞれ入力点列 $ABCD$ の4点、 $ABCDE$ の5点そして $ABCDE$ の5点から通過制御点と近傍制御点を基にした曲線を作成している。図1(d-1,d-2)は、非対称形状を作成する例である。これらを比較すると、繰り返しにより、曲線形状が入力点列の形状に近づいていることが分かる。なお、入力点列が3点から構成される場合には、曲線の表現形式変換行列を作成するための方程式としては、通過制御点である点 A, C に対応する4個の方程式と、近傍制御点である点 B に対応する2個の方程式に加えて、近傍制御点である点 B に対応する2個の方程式を重複して使用している。

4.3 選択的に近傍制御点を用いる効果の例

選択的に近傍制御点を用いる効果の例を、図2に示す。この例では、入力点列 $AFCHB$ から、点 A, B を通過制御点、点 F, C, H を近傍制御点として曲線を作成している。図2(a-1,a-2)は、形状最適化の始めの繰り返しから、全近傍制御点を使用する例であり、図

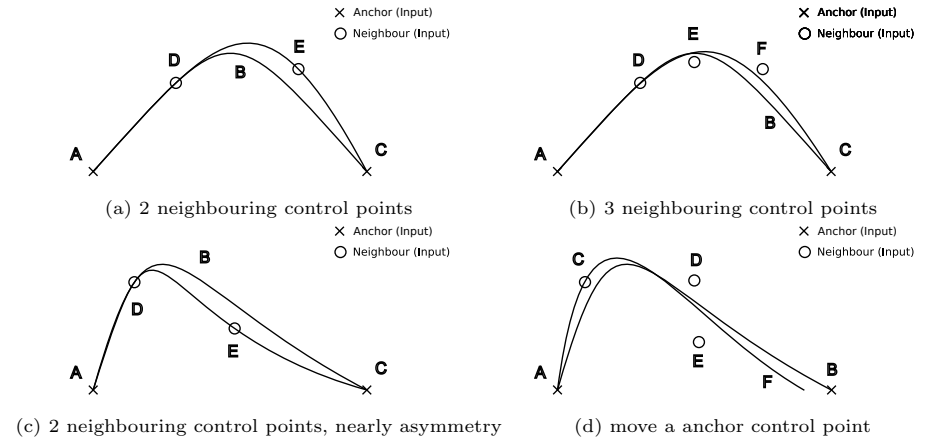


図3 近傍制御点を用いた曲線形状洗練化の例
Fig. 3 Examples of curve refinement using neighbouring control points.

2(b-1,b-2)は、形状最適化の始めの繰り返しでは、近傍制御点 C のみを選択して使用する例である。これらを比較すると、形状最適化の始めの繰り返しで、近傍制御点を選択して使用するとその選択した近傍制御点に重みを置いた形状を作成することが可能であることが分かる。

4.4 近傍制御点を用いた曲線形状洗練化の例

近傍制御点を用いた曲線形状洗練化の例を、図3に示す。図3(a)は、曲線 ACB を、点 A, B を通過制御点、点 C, D を近傍制御点として曲線を洗練化している。図3(b)は、曲線 ACB を、点 A, B を通過制御点、点 C, D, E を近傍制御点として曲線を洗練化している。図3(c)は、曲線 ACB を、点 A, B を通過制御点、点 C, D を近傍制御点として曲線を洗練化している。図3(d)は、曲線 AF を、点 A, B を通過制御点、点 C, D, E を近傍制御点として曲線を洗練化している。これらを比較すると、通過制御点と近傍制御点に合わせて曲線形状が洗練化されていることが分かる。

4.5 近傍制御点を用いた曲線の漸近的形状洗練化の例

近傍制御点を用いた曲線の漸近的形状洗練化の例を、図4に示す。これらはいずれも、曲線 ABC を形状洗練化している。図4(a,b)は、それぞれ通過制御点 A, C と近傍制御点 D 、そして通過制御点 A, C と近傍制御点 D, E に基づいて曲線を漸近的に形状洗練化している。図4(c,d)は、曲線の端点を移動する制約を加えた例であり、それぞれ通過制御点

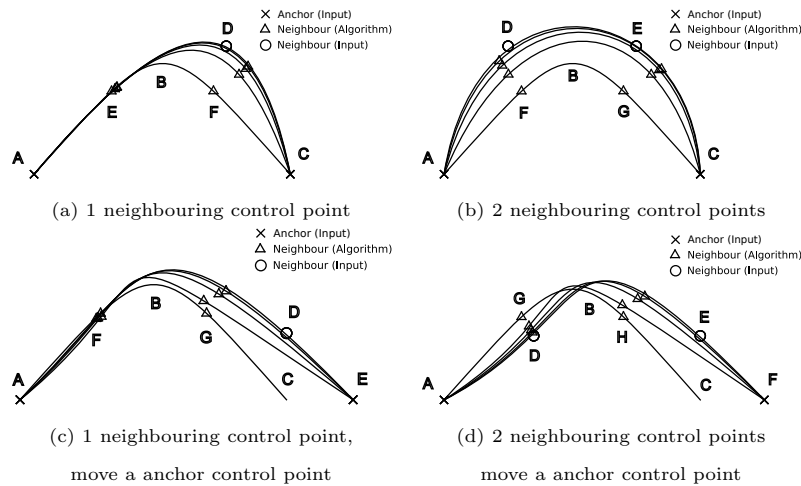


図 4 近傍制御点を用いた曲線の漸近的形状洗練化の例

Fig. 4 Examples of asymptotic curve refinement using neighbouring control points.

A, E と近傍制御点 D, そして通過制御点 A, F と近傍制御点 D, E, に基づいて曲線を漸近的に形状洗練化している。これらのを比較すると、通過制御点の制約を満足して、近傍制御点に漸近的に近づくように形状を変化させることが可能であり、これらの形状から希望の形状を選択可能であることが分かる。このことから、提案方式は、対話的操作による曲線形状操作のインタフェースとして適していると言える。

5. おわりに

曲線形状を直感的に作成または洗練化するためのためのユーザインタフェースを提案した。そのために、その点を曲線が通過するべき通過制御点と、その点の可能な限り近くを曲線が通過するべき近傍制御点に基づいて曲線形状を作成するアルゴリズムとそのためのインタフェースを提案した。この提案では、通過制御点を結ぶ直線から、近傍制御点と通過制御点を近似する曲線に、徐々に形状を近づけて曲線を作成すること、および洗練化前の曲線形状から形状洗練化のための制御点に漸近的に曲線を変化することにより洗練化することが特徴である。そのために、少ない点数の点列により曲線の概形を作成して、その後、微調整をして形状を洗練化するような曲線作成方法に適する。今後の課題には、制御点に対

応する曲線パラメータの決定方法を変更することにより、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) Cohen, J.M., Markosian, L., Zeleznik, R.C., Hughes, J.F. and Barzel, R.: An interface for sketching 3D curves, *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, New York, NY, USA, ACM, pp.17-21 (1999).
- 5) DOUGLAS1, D.H. and PEUCKER, T.K.: ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE, *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol.10, No.2, pp.112-122 (1973).
- 6) 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).
- 7) 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).
- 8) Otaduy, M.A., Igarashi, T. and LaViola, Jr., J.J.: Interaction: interfaces, algorithms, and applications, *SIGGRAPH '09: ACM SIGGRAPH 2009 Courses*, New York, NY, USA, ACM, pp.1-66 (2009).
- 9) Plass, M. and Stone, M.: Curve-fitting with piecewise parametric cubics, *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, ACM, pp.229-239 (1983).
- 10) 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).