情報処理学会論文誌 Vol.59 No.12 2288–2298 (Dec. 2018)

インタラクティブな手書き幾何作図のための自由曲線整形法

伊藤 友彦[1,a)] 神谷 葉月[1] 油谷 凜[1] 佐賀 聡人[1]

受付日 2018年2月23日, 採録日 2018年9月7日

概要：タブレットデバイスなどで手書き入力された自由曲線の整形技術が求められている．実際，局所的特徴点における曲線分割によるもの，美的曲線セグメント列への曲線分割によるもの，デザインの原理を用いた大局的な曲線修正によるもの，などの研究が進められている．しかし，これらは線分，円，円弧，楕円，楕円弧といった円錐曲線を主な構成要素とする幾何作図との調和を必ずしも意図したものではない．一方，我々はインタラクティブな手書き幾何作図インタフェースを実現するために手書きストロークを7クラスの幾何曲線（線分，円，円弧，楕円，楕円弧，閉自由曲線，開自由曲線）のいずれかにリアルタイムで同定する手法としてファジィスプライン曲線同定法（FSCI）を提案した．しかし，ここでは自由曲線（閉自由曲線，開自由曲線）と同定されたものに対する整形法は論じられていない．本論文では，インタラクティブな手書き幾何作図に適した自由曲線整形法を提案する．提案手法は手書きストロークが入力されるごとに逐次的に即応し，同定された自由曲線を「部分的に円錐曲線でありながら全体として G^1 連続な曲線」として整形する．これにより，自由曲線を含めた7クラスの幾何曲線すべてが調和した幾何作図を行える手書きインタフェースが実現される．

キーワード：自由曲線整形，幾何作図，手書き入力，図形認識，ファジィ理論

Free Curve Shaping for Interactive Sketch-based Drawing

Tomohiko Ito[1,a)] Hazuki Kamiya[1] Rin Aburaya[1] Sato Saga[1]

Received: February 23, 2018, Accepted: September 7, 2018

Abstract: Shaping techniques for freehand curves are required while drawing with tablet devices. In fact, several techniques have been proposed, such as the one that splits curves at local feature points, the one that splits curves into aesthetic curve segments, and the one that is based on design principles. However, these techniques are not necessarily intended to be harmonized with geometric drawings that consist of mainly conic sections such as line, circle, circular arc, ellipse, or elliptic arc. We have proposed the Fuzzy Spline Curve Identifier (FSCI) for interactive sketch-based drawing interfaces, where FSCI identifies each freehand stroke in real time as one of the seven classes of geometric curves, namely, line, circle, circular arc, ellipse, elliptic arc, closed free-curve, or open free-curve. However, FSCI includes no method to shape curves that are identified as free curves, namely, closed free-curves or open free-curves. This paper proposes a free-curve shaping method which is suitable for interactive sketch-based drawing interfaces. Our method immediately responds to each drawn stroke and shapes an identified free-curve into a G^1-continuous curve, which includes conic sections. This realizes a sketch-based interface for geometric drawing in which all the seven classes of geometric curves, which include free curves, are harmonized with each other.

Keywords: free curve shaping, geometric drawing, sketch-based drawing, figure recognition, fuzzy theory

1. 緒言

ペンタブレットやタッチディスプレイを用いた手書き入力インタフェースが普及するにともない手書き自由曲線の整形技術が求められている．実際，手書き自由曲線の整形に関して，角や変曲点などの局所的特徴点における曲線分割によるもの [1]，曲率と弧長から定義される美的曲線セグメント列への曲線分割によるもの [2]，複数の曲線間の平行性・滑らかさ・形状類似性・共曲線性などのデザインの原理を用いた大局的な曲線修正によるもの [3]，描画ストロークの

[1] 室蘭工業大学
Muroran Institute of Technology, Muroran, Hokkaido 050–8585, Japan
[a)] 17043006@mmm.muroran-it.ac.jp

速度情報を利用することで書き手の意図した詳細な形状を保存しつつ書き手の意図しないノイズを取り除くもの [4]，などの研究が進められている．しかし，これらは線分，円，円弧，楕円，楕円弧といった円錐曲線を主な構成要素とする幾何作図との調和を必ずしも意図したものではない．

一方，我々はインタラクティブな手書き幾何作図インタフェースを実現するために，手書き自由曲線をその描画軌跡だけではなく，描画の丁寧さの程度も考慮して 7 クラスの幾何曲線（線分（L），円（C），円弧（CA），楕円（E），楕円弧（EA），閉自由曲線（FC），開自由曲線（FO））[*1]のいずれかに同定する手法としてファジィスプライン曲線同定法（FSCI）[5] を提案した．また，複数の描画ストロークの重ね書きに即応してファジィスプライン曲線を逐次修正しつつ生成する手法として，逐次型ファジィスプライン曲線生成法（S-FSCG）[6] を提案した．さらに，これらを組み合わせて，インタラクティブな手書き幾何作図インタフェース [7] を実現した．しかし，文献 [5] では FC または FO と同定されたものに対する整形法について論じられておらず，これらの同定結果には手書き描画のうねりがそのまま残る．そのため，1 つの作図の中で，FC，FO といった自由曲線が L，C，CA，E，EA といった円錐曲線と調和しないという問題があった．

本論文ではインタラクティブな手書き幾何作図に適する自由曲線整形法として，以下の条件を満たす手法を提案する．

(1) **整形形状**：整形形状が部分的に円錐曲線でありながら全体として G^1 連続な曲線となる．
(2) **整形処理時間**：処理時間が 1 秒程度以下となる．
(3) **整形形状の詳細度の制御機能**：描画の丁寧さの程度を変化させることで整形形状の詳細度を制御できる．
(4) **整形形状の修正機能**：重ね書きによって整形結果を逐次的に修正できる．

(1) は FC，FO の整形形状が L，C，CA，E，EA と調和するための条件，(2) はインタフェースのインタラクティブ性を確保するための条件，(3) は描画の丁寧さの程度も考慮して認識を行う FSCI のストラテジーとの一貫性を保つための条件，(4) は S-FSCG による重ね書き修正機能との一貫性を保つための条件である．ここで，手書き幾何作図のための自由曲線整形法として，文献 [8] に手書きストロークを線分，円弧，クロソイド曲線の 3 クラスの幾何曲線から構成されるクロソイドスプライン曲線に変換するものが提案されている．しかし，これは形状の情報のみを用いているため (3) を満たさない．また，クロソイド曲線を含み，楕円弧を含まないため (1) を満たさない．そこで，上のすべての条件を満たす自由曲線整形法を実現するために，FSCI の幾何曲線同定機能を利用した自由曲線の円錐曲線列化アルゴリズムとその円錐曲線列の平滑化アルゴリズムを構築する．さらに，提案手法が上の条件を満たすことを実験により示す．

2. インタラクティブな手書き幾何作図の概要

3 章で提案する自由曲線整形法は，FSCI による手書き幾何曲線同定機能および S-FSCG による重ね書き幾何曲線修正機能を基盤としたインタラクティブな手書き幾何作図システムの内部で利用されることを前提とするものである．一方で，提案する自由曲線整形法ではさらにその内部で FSCI の幾何曲線同定機能を利用する．そこで，以下ではインタラクティブな手書き幾何作図システムについて FSCI と S-FSCG に焦点を置いて概説する．

2.1 FSCI による幾何曲線の手書き入力

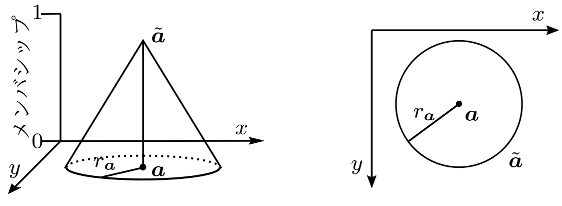
FSCI は手書きストロークが入力されるたびに，その形状と描画動作をもとにファジィスプライン曲線（FSC）を生成し，それを 7 クラスの幾何曲線（L，C，CA，E，EA，FC，FO）のいずれかとして同定する．これによりインタラクティブな幾何曲線の手書き入力が実現される．

2.1.1 FSC 生成

時系列点列（点の位置と時刻との組を要素とする列）として入力された手書きストロークをファジィスプライン補間 [9] することで，手書きストロークの描画軌跡およびその位置のあいまいさを表現するファジィな曲線 FSC $\tilde{\boldsymbol{s}} : [a, b] \to F$ を生成する．ただし，a は手書きストロークの開始時刻，b は終了時刻，F は円錐型ファジィ点全体の集合であり，FSC $\tilde{\boldsymbol{s}}$ は時刻 $t \in [a, b]$ に対応する円錐型ファジィ点 $\tilde{\boldsymbol{s}}(t)$ を返す写像である．円錐型ファジィ点 $\tilde{\boldsymbol{a}} \in F$ は位置のあいまいな点のモデルであり，底面の中心を $\boldsymbol{a} \in \mathbb{E}^2$，底面の半径を $r_{\boldsymbol{a}} \in \mathbb{R}$ とする**図 1** のような円錐型のメンバシップ関数で特徴づけられるファジィ集合で定義され，$\tilde{\boldsymbol{a}} = \langle \boldsymbol{a}, r_{\boldsymbol{a}} \rangle$ と記述される．ここで，$r_{\boldsymbol{a}}$ はファジィ点の位置のあいまいさの程度（ファジネス）を表しており，これが大きければあいまいな，小さければ厳密なファジィ点となる．

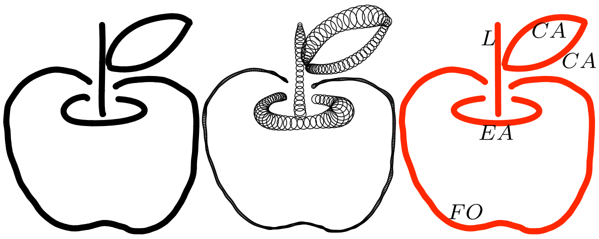
ここで，文献 [10] の手法で FSC $\tilde{\boldsymbol{s}}$ を生成すれば，この FSC $\tilde{\boldsymbol{s}}$ は時刻 $t \in [a, b]$ の変化とともに円錐型ファジィ点の移動軌跡を描くことになり，たとえば，**図 2** (a) の 5 つの手書きストロークに対して図 2 (b) の 5 つの FSC が生成される．このとき，**図 3** (a) に示す形状の似た手書きストロークであっても，図 3 (b) に示すように，描画動作の違いに応じて，素早く雑な描画部分ではファジネスの大きな FSC が生成され，逆に，ゆっくりとした丁寧な描画部分ではファ

[*1] 一般に，「線分」，「円」，「円弧」，「楕円」，「楕円弧」，「閉自由曲線」，「開自由曲線」という幾何曲線クラスによってクラス分けを行った場合，幾何曲線クラス間には包含関係が存在する．たとえば，「楕円弧」は「円弧」や「楕円」をその特殊な場合として含む．これに対して，本論文では文献 [5] の定義に従い，L，C，CA，E，EA，FC，FO というラベルを用いて包含関係の存在しないクラス分けを行う．たとえば，EA は「楕円弧」の性質を持ち，かつ，「円弧」の性質を持たず，かつ，「楕円」の性質を持たないものを指す．



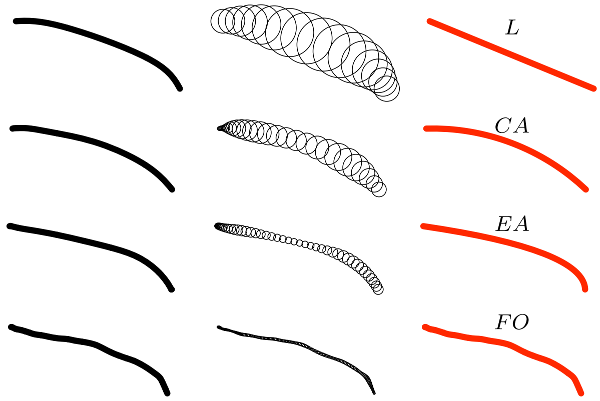
(a) 射影図　(b) 上面図

図 1　円錐型ファジィ点のメンバシップ関数
Fig. 1　Membership function of a conical fuzzy point.



(a) 手書きストローク　(b) FSC　(c) 同定幾何曲線

図 2　FSCI による幾何作図の例
Fig. 2　Geometric drawing by FSCI.



(a) 手書きストローク　(b) FSC　(c) 同定幾何曲線

図 3　FSC のファジネスの変化による幾何曲線同定結果の違い
Fig. 3　Difference of identification results by fuzziness variation of FSCs.

ジネスの小さな FSC が生成されるという効果が得られる．

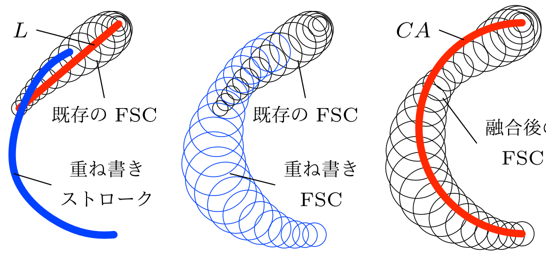
2.1.2　幾何曲線同定

得られた $\tilde{s}$ に対して，**表 1** のファジィ推論規則に従って，7 つの曲線クラスのグレード値 $\mu(L), \mu(C), \mu(CA), \mu(E), \mu(EA), \mu(FC), \mu(FO) \in [0,1]$ を算出する．ここで，P^{Line}, $P^{Circle}, P^{Ellipse} \in [0,1]$ はそれぞれ $\tilde{s}$ が線形，円形，楕円形である可能性値，$P^{Closed} \in [0,1]$ は $\tilde{s}$ が閉曲線である可能性値である．これらの可能性値とは文献 [11] のファジィ理論における可能性測度で求められる値であり，P^{Line}, P^{Circle}，$P^{Ellipse}$，P^{Closed} は，具体的には文献 [5] で提案されている FSCI の内部処理で計算される．$\neg$ は論理否定であり，可能性値 P に対して $\neg P = 1 - P$ と計算される．$\wedge$ は論理積で min 演算で計算される．

次に，L，C，CA，E，EA，FC，FO の中で最も高いグレード値を得た曲線クラスを選出し，その曲線クラスに対応した幾何曲線を同定幾何曲線として出力する．ここで，同定幾何曲線は，L，C，CA，E または EA の場合は 2 次有理 Bézier 曲線形式で，FC または FO の場合はスプライン曲線形式で出力される．

表 1　7 クラスの幾何曲線同定のためのファジィ推論規則
Table 1　Fuzzy rules to identify seven classes of geometric curves.

$\mu(L)$	$= P^{Line}$
$\mu(C)$	$= \neg P^{Line} \wedge P^{Circle} \wedge P^{Closed}$
$\mu(CA)$	$= \neg P^{Line} \wedge P^{Circle} \wedge \neg P^{Closed}$
$\mu(E)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge P^{Ellipse} \wedge P^{Closed}$
$\mu(EA)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge P^{Ellipse} \wedge \neg P^{Closed}$
$\mu(FC)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge \neg P^{Ellipse} \wedge P^{Closed}$
$\mu(FO)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge \neg P^{Ellipse} \wedge \neg P^{Closed}$



(a) 重ね書きストロークの入力　(b) 重ね書きストロークの FSC 生成　(c) 重ね書き後の幾何曲線同定

図 4　S-FSCG による幾何曲線の重ね書き修正の例
Fig. 4　Update of a geometric curve using S-FSCG.

以上の処理により，たとえば，図 2 (b) のような 5 つの FSC に対して図 2 (c) のように 5 つの幾何曲線が同定される．ここで，表 1 のファジィ推論規則はファジネスによる可能性の広がりが許す限り最も単純な曲線を推論しようとするものとなっている．このため，図 3 に示すように，形状の似た手書きストロークから生成された FSC であっても，ファジネスの大きさの違いに応じて，ファジネスの大きな FSC に対しては L といった自由度の低い単純な幾何曲線が優先的に同定され，一方で，ファジネスの小さな FSC に対しては FO といった自由度の高い詳細な幾何曲線が優先的に同定されるという効果が得られる．ユーザがこの効果を利用すれば，描画の丁寧さの程度を変化させることで同定結果の単純さや詳細度を自在に制御できる．

2.2　S-FSCG による幾何曲線の重ね書き修正

S-FSCG は，既存の FSC に対する重ね書きストロークが入力されるたびに，既存の FSC と重ね書きストロークの FSC を融合した FSC を新たに生成し，これで既存の FSC を更新する．S-FSCG を FSCI と併用すれば，たとえば**図 4** に示すように，既存の FSC に対して重ね書きストロークを入力することで幾何曲線の同定結果をインタラクティブに修正することが可能となる．

3. FSCI を利用した自由曲線整形法の提案

図 2 (c) のスプライン曲線形式の FO には手書きストロークのうねりが整形されずに残っており，L，CA，EA といった周囲の円錐曲線と調和していない．このような問題を解決するには自由曲線（FC，FO）に対する整形が必要となるが，文献 [5] ではこれについて論じられていない．そこで，インタラクティブな手書き幾何作図に適した自由曲線の整形法として 1 章で述べた条件 (1) から (4) までを満たす手法を提案する．提案手法は，FSCI で FC，FO と同定されたものを対象とする．ここでは，FSCI で FC，FO と同定された手書きストロークに対して再度 FSCI を適用することでこれを円錐曲線列に分割し，さらにその円錐曲線列を平滑化するという流れで，「部分的に円錐曲線でありながら全体として G^1 連続な曲線」を生成する（条件 (1) の達成）．また，これらの処理では FSCI の同定特性を利用した効率的アルゴリズムにより 1 秒程度以下の処理時間を達成する（条件 (2) の達成）．ここで，FSCI を用いているため，描画の丁寧さの程度を変化させることによる整形形状の詳細度の制御が可能となる（条件 (3) の達成）．また，2.2 節で述べたように FSCI は S-FSCG と併用可能であるため，重ね書きによる整形形状の修正も可能となる（条件 (4) の達成）．

以下では，まず FSCI を利用して円錐曲線を同定する方法を 3.1 節で示したうえで，円錐曲線列化と平滑化のアルゴリズムを 3.2 節および 3.3 節で示す．

3.1 FSCI による円錐曲線の同定

提案手法では，入力された手書きストロークを円錐曲線の列に分割する必要がある．ここで，L，CA，EA の 3 つの開いた幾何曲線クラスをまとめて**円錐曲線**と呼び[*2]，これを CS と表記することとする．円錐曲線列への分割を実現するために，まず，FSCI のファジィ推論規則を表 1 に示したオリジナルのものから**表 2** に示すものに置き換える．これにより，FSCI は L，CA，EA，FO の 4 クラスの幾何曲線のいずれかの同定結果を出力することになる[*3]．ここで，同定結果の曲線クラスが L，CA または EA である場合は CS に，それ以外の場合は FO に分類することにすれば，FSCI は分類結果として CS または FO を出力することになる．また，同定結果の曲線クラスが CS であるグレード値を**円錐曲線性** $\mu(CS) \in [0,1]$ と呼ぶこととすると，$\mu(CS)$ は同定結果の曲線クラスが FO ではないグレード値として，$\mu(CS) = \neg\mu(FO)$ と計算されることになる．

表 2　4 クラスの幾何曲線同定のためのファジィ推論規則

Table 2　Fuzzy rules to identify four classes of geometric curves.

$\mu(L)$	$= P^{Line}$
$\mu(CA)$	$= \neg P^{Line} \wedge P^{Circle}$
$\mu(EA)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge P^{Ellipse}$
$\mu(FO)$	$= \neg P^{Line} \wedge \neg P^{Circle} \wedge \neg P^{Ellipse}$

3.2 FSC の円錐曲線列化

円錐曲線列化では，手書きストロークから生成された FSC を部分 FSC の列に最適分割したうえで，それぞれの部分 FSC を FSCI で同定することで，一筆書きの手書きストロークを円錐曲線列に変換する．ただし，最適分割とは以下の条件を満たす分割である．

(1) すべての部分 FSC が FSCI により CS と同定される．
(2) 条件 (1) を満たしたうえで，分割数（分割後の部分 FSC の数）が最小となる．
(3) 条件 (2) を満たしたうえで，円錐曲線列性（以下で定義）が最大となる．

ここで，(2) は 2.1.2 項の FSCI のストラテジに呼応するものであり，円錐曲線列としてもファジネスによる可能性の広がりが許す限り最も単純な構造のものを選択するための条件である．このような FSC の分割を実現する具体的な手法を提案する準備として以下の定義を行う．

部分 FSC $\tilde{\boldsymbol{s}}_{a',b'}$：FSC $\tilde{\boldsymbol{s}}$ の定義域を $[a',b'] \subseteq [a,b]$ で制限したものを部分 FSC $\tilde{\boldsymbol{s}}_{a',b'}$ と定義する．

探索点時刻列 T：FSC $\tilde{\boldsymbol{s}}$ の定義域 $[a,b]$ を $n-1$ 等分する時刻の列を探索点時刻列 $T = \{t_i \mid a = t_0 < t_1 < \cdots < t_{n-1} = b\}_{i \in I}$（$I = \{0,\ldots,n-1\}$）と定義する．

部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の曲線クラスの分類結果 $r_{i,j}$：部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ を FSCI に入力したときに出力される同定結果の曲線クラスの分類結果を $r_{i,j} \in \{CS, FO\}$ と定義する．

部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の円錐曲線性 $\mu_{i,j}(CS)$：部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ を FSCI に入力したときに出力される円錐曲線性 $\mu(CS)$（3.1 節参照）を $\mu_{i,j}(CS) \in [0,1]$ と定義する．

部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の分割点インデックス列：部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ に対してその定義域 $[t_i,t_j]$ 内に存在する探索点時刻のインデックスを要素とする列 $P = \{p_k \mid p_k \in I,\ i = p_0 < \cdots < p_m = j\}_{k=0}^{m}$ を考える．ここで，$\forall k \in \{1,\ldots,m\}$，$r_{p_{k-1},p_k} = CS$ が満たされるとき，P を $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の分割点インデックス列と呼ぶ．すなわち，分割点インデックス列は分割後の部分 FSC $\tilde{\boldsymbol{s}}_{t_{p_{k-1}},t_{p_k}}$（$1 \le k \le m$）がすべて FSCI によって CS と同定されるような分割を表す．

FSC $\tilde{\boldsymbol{s}}$ の分割点インデックス列：$\tilde{\boldsymbol{s}}_{t_0,t_{n-1}}$ の分割点インデックス列を $\tilde{\boldsymbol{s}}$ の分割点インデックス列と呼ぶ．これは $\tilde{\boldsymbol{s}}_{t_0,t_{n-1}} = \tilde{\boldsymbol{s}}$ であることによる．

部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の円錐曲線列性 $\mu_{i,j}^{P}$：部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ とその分割点インデックス列 P が与えられたとき $\mu_{i,j}^{P} = \bigwedge_{k=1}^{m} \mu_{p_{k-1},p_k}(CS) \in [0,1]$ を部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$

[*2] ここでは双曲線および放物線を扱わないこととする．
[*3] 曲線列の構成要素として C，E，FC といった閉曲線は用いないことに注意する．

情報処理学会論文誌 Vol.59 No.12 2288–2298 (Dec. 2018)

の円錐曲線列性と定義する．これは部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ を P に基づいて分割して得られる m 個の部分 FSC $\tilde{\boldsymbol{s}}_{t_{p_{k-1}},t_{p_k}}$（$1 \le k \le m$）それぞれの円錐曲線性 $\mu_{p_{k-1},p_k}(CS)$ すべての論理積をとったものであり，部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ が円錐曲線列の性質を満たすグレード値を表す．

FSC $\tilde{\boldsymbol{s}}$ の円錐曲線列性 μ^P：$\mu^P = \mu^P_{0,n-1}$ を $\tilde{\boldsymbol{s}}$ の円錐曲線列性と定義する．これは $\tilde{\boldsymbol{s}}_{t_0,t_{n-1}} = \tilde{\boldsymbol{s}}$ であることによる．

FSC $\tilde{\boldsymbol{s}}$ の円錐曲線列 Q^P：FSC $\tilde{\boldsymbol{s}}$ をその分割点インデックス列 P に基づいて分割して得られる m 個の部分 FSC $\tilde{\boldsymbol{s}}_{t_{p_{i-1}},t_{p_i}}$（$1 \le i \le m$）のそれぞれを FSCI で同定した結果得られる幾何曲線（円錐曲線）を，$\boldsymbol{q}_i^P : [0,1] \to \mathbb{E}^2$（$0 \le i \le m-1$）とする．このとき，列 $Q^P = \{\boldsymbol{q}_i^P\}_{i=0}^{m-1}$ を円錐曲線列と定義する．ここで，$\boldsymbol{q}_i^P$ は重み w_i^P，制御点 $\boldsymbol{b}^P_{i\ j}$（$0 \le j \le 2$）を用いて，以下の 2 次有理 Bézier 曲線形式で表される[*4]．

$$\boldsymbol{q}_i^P(t) = \frac{B_0^2(t)\boldsymbol{b}^P_{i\ 0} + B_1^2(t)w_i^P\boldsymbol{b}^P_{i\ 1} + B_2^2(t)\boldsymbol{b}^P_{i\ 2}}{B_0^2(t) + B_1^2(t)w_i^P + B_2^2(t)}$$

FSC の円錐曲線列への分割は，以下に示すとおり，FSC を離散化して探索点を生成し，その中から分割点の候補を絞り込んだうえで，二分探索を繰り返して $\tilde{\boldsymbol{s}}$ の円錐曲線列性 μ^P を最大化する $\tilde{\boldsymbol{s}}$ の分割点インデックス列を決定することにより実現する．なお，この探索アルゴリズムでは FSCI の同定における以下の 2 つの特性を利用する．

(1) ある部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ の定義域 $[t_i,t_j]$ が拡大すると，その円錐曲線性は減少する．すなわち，$[t_i,t_j] \subseteq [t_{i'},t_{j'}]$ ならば $\mu_{i,j}(CS) \ge \mu_{i',j'}(CS)$ となる．

(2) 同定結果が FO となる部分 FSC $\tilde{\boldsymbol{s}}_{t_i,t_j}$ は，その定義域 $[t_i,t_j]$ を拡大しても FO と同定される．すなわち，$[t_i,t_j] \subseteq [t_{i'},t_{j'}]$ かつ $r_{i,j} = FO$ ならば $r_{i',j'} = FO$ となる．

3.2.1 FSC の生成

一筆書きの手書きストロークから 2.1.1 項と同様に FSC $\tilde{\boldsymbol{s}}$ を生成する．たとえば図 **5** の手書きストロークから図 **6** のような FSC が生成される．

3.2.2 探索点列の生成

FSC $\tilde{\boldsymbol{s}}$ の定義域 $[a,b]$ 上に n 個の時刻を等時間間隔でサンプリングすることで，探索点時刻列 $T = \{t_i\}_{i \in I}$ を生成する．ここで，$\forall i \in \{0, n-2\}$，$r_{i,i+1} = CS$ が満たされるように探索点数 n を十分大きく設定する[*5]．図 **7** に探索点時刻列に対応した FSC 上の点列（探索点列）の例を示す．

3.2.3 分割点候補の絞り込みと最小分割数 m の決定

最適分割の条件 (1)，(2) を満たす分割点候補を絞り込み，最小分割数 m を決定する．まず，分割数が最小となる $\tilde{\boldsymbol{s}}$ の分割点インデックス列の中で各要素に対応する分割点が終点側へ最も偏った分割点インデックス列 $P^L = \{p_i^L\}_{i=0}^m$ を求める．具体的には $p_0^L = 0$ と始点時刻のインデックスで初期化し，終点自体が分割点となるまで順次

$$p_i^L = \max\{p \mid p \in I,\ p > p_{i-1}^L,\ r_{p_{i-1}^L,p} = CS\} \qquad (1)$$

と P^L の要素を求める．この処理が完了した時点で最小分割数 m が決定される[*6]．

次に，分割数が最小となる $\tilde{\boldsymbol{s}}$ の分割点インデックス列の中で各要素に対応する分割点が始点側へ最も偏った分割点インデックス列 $P^R = \{p_i^R\}_{i=0}^m$ を求める．具体的には $p_m^R = n-1$ と終点時刻のインデックスで初期化し，始点自身が分割点となるまで順次

$$p_i^R = \min\{p \mid p \in I,\ p < p_{i+1}^R,\ r_{p,p_{i+1}^R} = CS\} \qquad (2)$$

と P^R の要素を求める．

さらに，求めた P^L と P^R から最適な分割点候補のインデックス集合をそれぞれ $A_i = \{p \mid p_i^R \le p \le p_i^L\}$（$0 \le i \le m$）と絞り込む[*7]．たとえば，図 7 の探索点列から分割点候補を絞り込むと図 **8** のようになる．この例では最小分割数は $m = 4$ となっている．

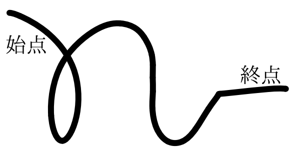


図 **5** 手書きストローク
Fig. 5 Freehand stroke.

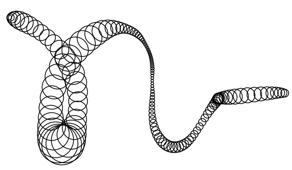


図 **6** FSC
Fig. 6 FSC.

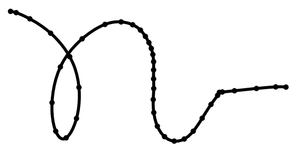


図 **7** 探索点列
Fig. 7 Search points.

*4 $B_j^k : [0,1] \to \mathbb{R}$ は k 次のバーンスタイン基底関数である．

*5 本論文では $n = 40$ とした．十分な探索点数は最大描画時間に依存するが，15 秒程度であればこれで十分であることを予備実験により確認した．

*6 式 (1) では p_i^L として，部分 FSC $\tilde{\boldsymbol{s}}_{t_{p_{i-1}^L},t_p}$ が CS と同定される範囲で，最も終点側にある p を求めている．そのため，m は最小分割数となることが保証される．

*7 式 (2) では，式 (1) とは逆に，p_i^R として，部分 FSC $\tilde{\boldsymbol{s}}_{t_p,t_{p_{i+1}^R}}$ が CS と同定される範囲で，最も始点側にある p を求めている．P^L，P^R の要素をそのように求めると，A_i（$0 \le i \le m$）は互いに共通の要素を持たないことになる．

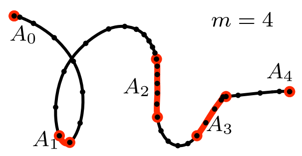


図 8　分割点の候補
Fig. 8　Candidates for segmentation points.

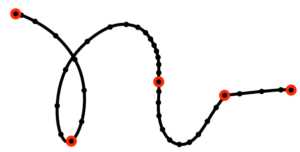


図 9　分割点列
Fig. 9　Segmentation points.

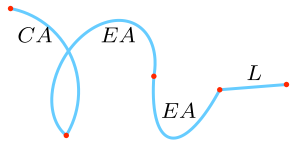


図 10　円錐曲線列
Fig. 10　Conic sections.

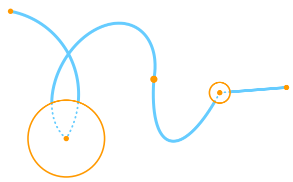


図 11　刈り込み円
Fig. 11　Pruning circles.

3.2.4　$\tilde{s}$ の円錐曲線列性を最大化する $\tilde{s}$ の分割点インデックス列 P^O の探索

3.2.3 項で絞り込んだ分割点候補の中から最適分割の条件 (3) を満たす $\tilde{s}$ の分割点インデックス列 P^O を探索により決定する．この探索では，始点から部分 FSC を拡大しながら，順次これを最適分割し，その結果を保存する処理を繰り返す．各ステップでは前のステップの結果を利用することで動的計画法と同様に再計算を避けることができる．

具体的には，すべての $j \in \bigcup_{i=1}^{m} A_i$ に対して j の小さい方から順に，P_j および $\mu_{0,j}^{P_j}$ を以下のように求めて，結果を保存していく．ただし，P_j は部分 FSC $\tilde{s}_{t_0,t_j}$ の最適な分割点インデックス列である．

$j \in A_1$ の場合

$$P_j = \{p_i^j \mid p_0^j = 0,\ p_1^j = j\}_{i=0}^{1}$$
$$\mu_{0,j}^{P_j} = \mu_{0,j}(CS)$$

$j \in A_i$ $(1 < i \leq m)$ の場合

$$P_j = \mathrm{append}(P_k, j)$$
$$\mu_{0,j}^{P_j} = \mu_{0,k}^{P_k} \wedge \mu_{k,j}(CS)$$

ただし，$\mathrm{append}(X, x)$ は列 X の末尾に要素 x を追加した列を返す．また，$k \in I$ として

$$k = \mathop{\arg\max}_{k' \in K} \mu_{0,k'}^{P_{k'}} \wedge \mu_{k',j}(CS),$$
$$K = \{k' \mid 0 < k' < j,\ r_{k',j} = CS\}$$

を満たすインデックスを探索する．ここで，3.2.3 項の絞り込みにより $k \in A_{i-1}$ であることが分かっており，さらに，FSCI の同定特性 (1) より，$p_{i-1}^R \leq k \leq p_{i-1}^L$ の範囲で $\mu_{0,k}^{P_k} \wedge \mu_{k,j}(CS)$ は単峰となるため，二分探索により効率的に k を探索することができる[*8]．具体的には目的関数 $f : I \to [0,1]$ を

$$f(k) = \begin{cases} \mu_{0,k}^{P_k} \wedge \mu_{k,j}(CS) & (p_{i-1}^R \leq k \leq p_{i-1}^L) \\ 0 & (\text{otherwise}) \end{cases}$$

と定義し，これを最大化する k を以下の二分探索により $k = \mathrm{find}(p_{i-1}^R, p_{i-1}^L)$ と求める．

$$\mathrm{find}(l, r) = \begin{cases} \mathrm{find}(c+1, r) & (r_{c,j} \neq CS \text{ or } \\ & \ f(c) < f(c+1)) \\ \mathrm{find}(l, c-1) & (f(c) \leq f(c-1)) \\ c & (\text{otherwise}) \end{cases}$$

ただし，$c = \lceil \frac{r+l}{2} \rceil$ である．

最終的に $j = n-1$ となった時点で $\tilde{s}$ の最適な分割点インデックス列が $P^O = P_{n-1}$ と得られる[*9]．得られた分割点インデックス列に対応した FSC 上の点列（分割点列）を**図 9** に示す．

3.2.5　円錐曲線列の生成

3.2.4 項で得られた $\tilde{s}$ の最適な分割点インデックス列 P^O に対応する円錐曲線列 $Q^{P^O} = \{\boldsymbol{q}_i^{P^O}\}_{i=0}^{m-1}$ を生成する．円錐曲線列を**図 10** に示す．

3.3　円錐曲線列の平滑化

円錐曲線列 Q^{P^O} は分割点において G^1 連続ではない．そこで，**図 11** に示すようにファジィ分割点のファジネスに応じた刈り込み円によって角とその周囲の円錐曲線の一部を取り除く．その後，不連続となった部分を 3 次 Bézier 曲線によって再接続することで，全体として G^1 連続に整形された 1 本の曲線を**図 12** のように生成する．以下では FO の刈り込みと再接続の具体的な方法を示すが，FC についても同様に処理することができる．ただし，FC では始終点においても刈り込みおよび再接続を行うこととする．

*8　一般に k の探索に線形探索を用いれば，円錐曲線列化の時間計算量は $O(n^2)$ となる．しかし，$\mu_{0,k}^{P_k} \wedge \mu_{k,j}(CS)$ は単峰であるため，k の探索には二分探索を用いれば十分である．これにより円錐曲線列化の時間計算量は $O(n \log n)$ となる．

*9　$\tilde{s}$ の最適な分割点インデックス列は一般に複数個存在するが，P^O はそのうちの 1 つとなる．

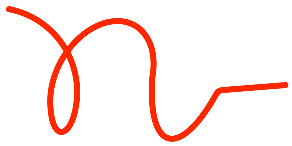
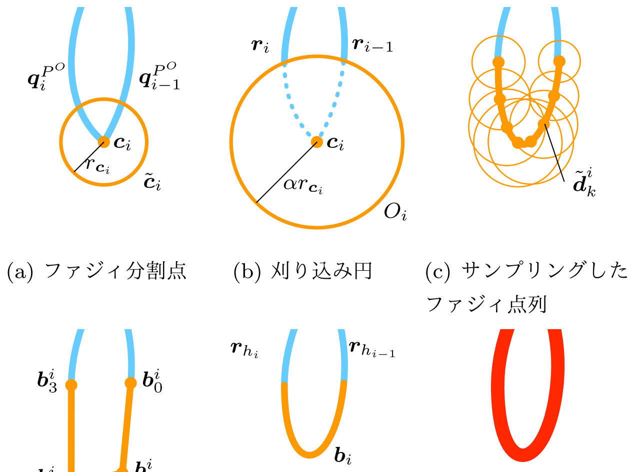


図 12 整形結果
Fig. 12 Shaped result.



(a) ファジィ分割点　(b) 刈り込み円　(c) サンプリングしたファジィ点列

(d) 制御点　(e) 3 次 Bézier 曲線　(f) 再接続後の曲線

図 13 ファジィ分割点における刈り込みと再接続
Fig. 13 Pruning and re-connecting at a fuzzy segmentation point.

3.3.1 ファジィ分割点における刈り込み

3.2.4 項で得られた $\tilde{s}$ の分割点インデックス列 $P^O = \{p_i^O\}_{i=0}^m$ からファジィ分割点列を $C = \{\tilde{c}_i \mid \tilde{c}_i = \tilde{s}(t_{p_i^O})\}_{i=0}^m$ と求める．図 **13** (a) のファジィ分割点の例を示す．

次に，$\tilde{c}_i$（$0 \le i \le m$）の中心 $\boldsymbol{c}_i$，ファジネス $r_{\boldsymbol{c}_i}$ をもとにして，中心 $\boldsymbol{c}_i$，半径 $\alpha r_{\boldsymbol{c}_i}$ の刈り込み円 O_i を設定する．だたし，$\alpha > 0$ は刈り込み円の大きさを調節するパラメータである[*10]．ここで，FO の始終点では刈り込みを行わないため，O_0，O_m の半径を 0 とする．刈り込み円の例を図 13 (b) に示す．

さらに，すべての $i \in \{1, \ldots, m-1\}$ に対して，円錐曲線 $\boldsymbol{q}_{i-1}^{P^O}$，$\boldsymbol{q}_i^{P^O}$ から O_i に含まれる部分を取り除く．これにより，刈り込み後の円錐曲線の列 $R = \{\boldsymbol{r}_i \mid \boldsymbol{r}_i = \boldsymbol{q}_i^{P^O}{}_{u_0^i, u_1^i}\}_{i=0}^{m-1}$ が得られる．ただし，$\boldsymbol{q}_i^{P^O}{}_{u_0^i, u_1^i}$（$0 \le i < m$）は $\boldsymbol{q}_i^{P^O}$ の定義域を $[u_0^i, u_1^i] \subseteq [0,1]$ に制限した円錐曲線であり，

$$u_0^i = \min(\{u \mid u \in [0,1], \|\boldsymbol{q}_i^{P^O}(u) - \boldsymbol{c}_i\| = \alpha r_{\boldsymbol{c}_i}\} \cup \{1\}),$$
$$u_1^i = \max(\{u \mid u \in [0,1], \|\boldsymbol{q}_i^{P^O}(u) - \boldsymbol{c}_{i+1}\| = \alpha r_{\boldsymbol{c}_{i+1}}\} \cup \{0\})$$

である．

以上の刈り込み処理により，あいまいなファジィ分割点の周囲では円錐曲線が大きく削られ，厳密なファジィ分割点の周囲では円錐曲線の大部分が残ることとなる．

[*10] 本論文では $\alpha = 2$ とした．

3.3.2 3 次 Bézier 曲線による再接続

3.3.1 項で求めた $\boldsymbol{r}_i$ のうち，$u_0^i > u_1^i$ となるものは円錐曲線全体が刈り込まれて実質消滅している．そこで，消滅せずに残存している円錐曲線 $\boldsymbol{r}_i$ のインデックスを取り出し，昇順に並べた列を $H = \{h_i\}_{i=0}^{m'-1}$ とおく．ここで，m' は残存している円錐曲線の個数である．

次に，残存した円錐曲線間の部分 FSC $\tilde{\boldsymbol{s}}_{a_{h_{i-1}+1}, b_{h_i}}$（$0 \le i \le m'$）を等時間間隔でサンプリングして，ファジィ点列 $D_i = \{\tilde{\boldsymbol{d}}_k^i\}_{k=0}^{N-1}$ を求める[*11]．ここで，$h_{-1} = -1$，$h_{m'} = m$ であり，$\forall h \in \{0, \ldots, m\}$ に対して，

$$a_h = \max(\{a' \mid a' \in [a, t_{p_h}], \|\boldsymbol{s}(a') - \boldsymbol{c}_h\| = \alpha r_{\boldsymbol{c}_h}\} \cup \{a\}),$$
$$b_h = \min(\{b' \mid b' \in [t_{p_h}, b], \|\boldsymbol{s}(b') - \boldsymbol{c}_h\| = \alpha r_{\boldsymbol{c}_h}\} \cup \{b\})$$

である．図 13 (c) にサンプリングしたファジィ点列の例を示す．

最後に，残存した円錐曲線間の部分 FSC を近似する 3 次 Bézier 曲線 $\boldsymbol{b}_i : [0,1] \to \mathbb{E}^2$（$0 \le i \le m'$）を求め，これで残存する円錐曲線を再接続する．具体的には，3 次 Bézier 曲線を制御点 $\boldsymbol{b}_j^i$（$0 \le j \le 3$）を用いて

$$\boldsymbol{b}_i(t) = \sum_{j=0}^{3} B_j^3(t) \boldsymbol{b}_j^i$$

と表すこととし，評価関数

$$E_i = \sum_{k=0}^{N-1} w_k^i \left\| \boldsymbol{d}_k^i - \boldsymbol{b}_i(t_k^i) \right\|^2 \tag{3}$$

を最小化する制御点 $\boldsymbol{b}_j^i$ を求める．ただし，$t_k^i = \frac{\sum_{l=1}^{k} \|\boldsymbol{d}_l^i - \boldsymbol{d}_{l-1}^i\|}{\sum_{l=1}^{N-1} \|\boldsymbol{d}_l^i - \boldsymbol{d}_{l-1}^i\|}$ はパラメータ，$w_k^i = \frac{1}{r_{\boldsymbol{d}_k^i}}$ は重み[*12]である．また，円錐曲線との接続部分を G^1 連続とするために制御点には以下の拘束条件を課す．

端点一致条件

$$\boldsymbol{b}_0^i = \boldsymbol{r}_{h_{i-1}}(u_1^{h_{i-1}}) \quad (0 < i \le m') \tag{4}$$
$$\boldsymbol{b}_3^i = \boldsymbol{r}_{h_i}(u_0^{h_i}) \quad (0 \le i < m') \tag{5}$$

端点における接線方向一致条件

$$\boldsymbol{b}_1^i = \boldsymbol{b}_0^i + \beta_i \boldsymbol{r}'_{h_{i-1}}(u_1^{h_{i-1}}) \quad (\beta_i \in \mathbb{R},\ 0 < i \le m') \tag{6}$$
$$\boldsymbol{b}_2^i = \boldsymbol{b}_3^i + \gamma_i \boldsymbol{r}'_{h_i}(u_0^{h_i}) \quad (\gamma_i \in \mathbb{R},\ 0 \le i < m') \tag{7}$$

ここで，$\boldsymbol{r}'_l(t)$（$0 \le l < m$）は $\boldsymbol{r}_l$ の $t \in [u_0^l, u_1^l]$ における接ベクトルである．

このとき，$\boldsymbol{b}_0^i$（$0 < i \le m'$），$\boldsymbol{b}_3^i$（$0 \le i < m'$）は式 (4), (5) より決定され，残りの制御点 $\boldsymbol{b}_0^0$，$\boldsymbol{b}_1^i$（$0 \le i \le m'$），$\boldsymbol{b}_2^i$（$0 \le i \le m'$），$\boldsymbol{b}_3^{m'}$ を決定する未知パラメータは

$i = 0$ のとき　x_0^0，y_0^0，x_1^0，y_1^0，γ_0，

$0 < i < m'$ のとき　β_i，γ_i，

[*11] 本論文では $N = 33$ とした．
[*12] これは，あいまいなファジィ点の重みを小さく，厳密なファジィ点の重みを大きくするための重み設定である．

情報処理学会論文誌　Vol.59　No.12　2288–2298 (Dec. 2018)

$i = m'$ のとき　$\beta_{m'}$，$x_2^{m'}$，$y_2^{m'}$，$x_3^{m'}$，$y_3^{m'}$

となる．ただし，x_j^i，y_j^i はそれぞれ $\boldsymbol{b}_j^i$ の x 座標値，y 座標値である．したがって，未知パラメータベクトルを

$$\boldsymbol{\xi}_0 = (x_0^0, y_0^0, x_1^0, y_1^0, \gamma_0)^T$$
$$\boldsymbol{\xi}_i = (\beta_i, \gamma_i)^T \ (0 < i < m')$$
$$\boldsymbol{\xi}_{m'} = (\beta_{m'}, x_2^{m'}, y_2^{m'}, x_3^{m'}, y_3^{m'})^T$$

とおき式 (3) を $\boldsymbol{\xi}_i$ の関数と見なせば，線形方程式 $\frac{\partial E_i(\boldsymbol{\xi}_i)}{\partial \boldsymbol{\xi}_i} = \mathbf{0}$ を解くことで，すべての未知パラメータが得られる．得られた x_0^0，y_0^0，x_1^0，y_1^0，$x_2^{m'}$，$y_2^{m'}$，$x_3^{m'}$，$y_3^{m'}$ より $\boldsymbol{b}_0^0$，$\boldsymbol{b}_1^0$，$\boldsymbol{b}_2^{m'}$，$\boldsymbol{b}_3^{m'}$ が求められ，β_i（$0 < i \leq m'$），γ_i（$0 \leq i < m'$）をそれぞれ式 (6)，(7) に代入することにより $\boldsymbol{b}_1^i$（$0 < i \leq m'$），$\boldsymbol{b}_2^i$（$0 \leq i < m'$）が求められる．

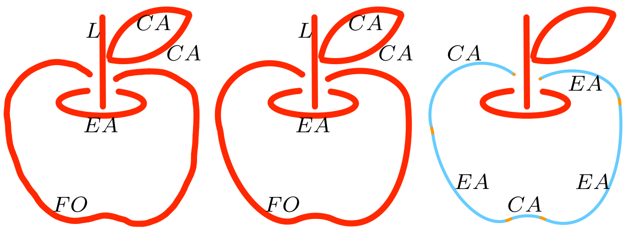
以上より，図 13 (d) に示すような制御点が決定され，図 13 (e) に示すような 3 次 Bézier 曲線 $\boldsymbol{b}_i$ が生成される．これで残存する円錐曲線を再接続し，図 13 (f) に示すように全体として G^1 連続な曲線を生成する[*13]．

4. 動作実験

提案手法が 1 章で述べた 4 つの性質を満たした自由曲線整形法となっていることを確認する動作実験を行った．動作実験には Intel Core i5 (3.1 GHz)，メモリ 16 GB，macOS Sierra 搭載のパーソナルコンピュータに接続した液晶ペンタブレットを用いた[*14]．プログラムは Kotlin を用いて実装した．

4.1 自由曲線整形法の幾何作図への適用例

提案した自由曲線整形法を図 2 の幾何作図へ適用したことによる効果を**図 14** で比較して示す．提案手法によって FO は図 14 (c) のような構成要素からなる曲線として整形された．これにより，図 14 (d) の FO では，図 14 (a) に残っていた手書きストロークのうねりが解消されて，周囲の幾何曲線と調和することが分かる．



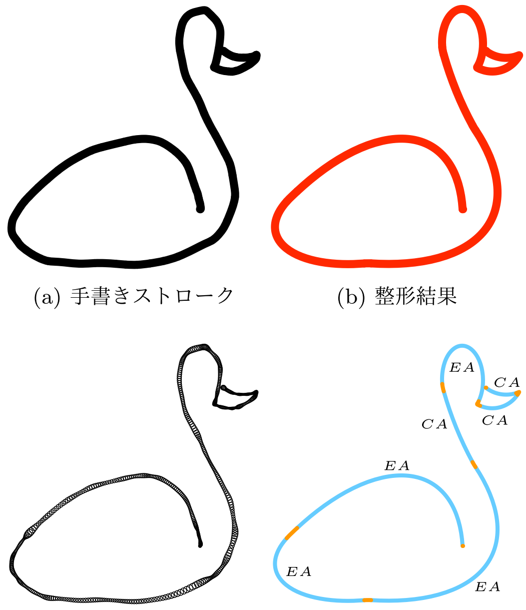
(a) FO への整形を行わない場合の同定幾何曲線　(b) FO への整形を行った場合の同定幾何曲線　(c) FO の整形結果の構成要素

図 14　図 2 の FO に対する自由曲線整形の効果
Fig. 14　Effect of free curve shaping for FO in Fig. 2.

[*13] この曲線は CAD で一般的に用いられる NURBS 曲線形式で表現できるものである．
[*14] 動作実験の様子については下記の動画を参照されたい．https://sites.google.com/view/fscshaping

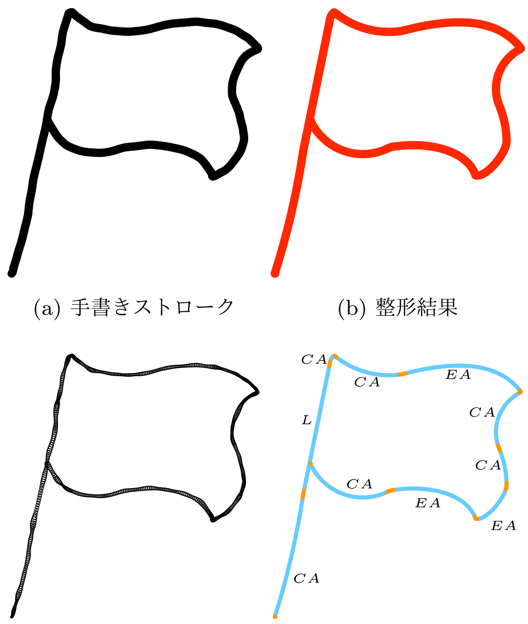
4.2 整形形状と処理時間

図 15，**図 16** および**図 17** に提案手法による自由曲線整形の例を示す．どの例でも，円錐曲線が 3 次 Bézier 曲線によって G^1 連続に接続された形状に整形され，手書きストロークのうねりが解消されている．また，「スワン」，「フラッグ」，「ヨット」の描画時間および整形処理時間を**表 3** に示す．このような十数秒程度の描画時間の長い手書きストロークに対しても，整形処理時間はたかだか 1 秒程度となっており，インタラクティブなインタフェースに十分利



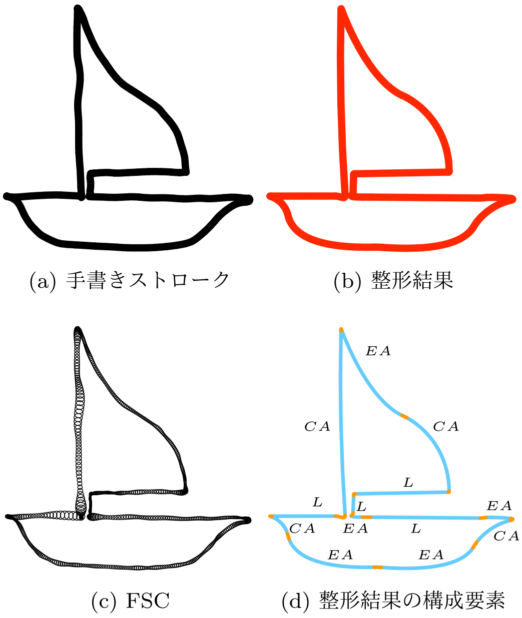
(a) 手書きストローク　(b) 整形結果　(c) FSC　(d) 整形結果の構成要素

図 15　「スワン」の整形
Fig. 15　Shaping for "Swan."



(a) 手書きストローク　(b) 整形結果　(c) FSC　(d) 整形結果の構成要素

図 16　「フラッグ」の整形
Fig. 16　Shaping for "Flag."



(a) 手書きストローク　(b) 整形結果

(c) FSC　(d) 整形結果の構成要素

図 17　「ヨット」の整形
Fig. 17　Shaping for “Yacht.”

表 3　「スワン」,「フラッグ」,「ヨット」の描画時間および整形処理時間
Table 3　Drawing times and processing times of “Swan,” “Flag” and “Yacht.”

	描画時間 [s]	整形処理時間 [s]
「スワン」	11.62	0.97
「フラッグ」	12.10	0.91
「ヨット」	14.90	1.00

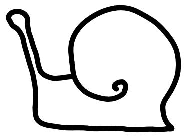


図 18　手書きストローク「かたつむり」
Fig. 18　Freehand stroke “Snail.”

用可能である．ここで，提案手法は局所的特徴点抽出に基づいた曲線分割を行っていないにもかかわらず，角や変曲点では結果的に分割されていることに注意する．

4.3　描画の丁寧さの程度の変化による整形形状の詳細度の制御

描画の丁寧さの程度が整形形状の詳細度にどのように影響するかを示す模擬実験を行った．ここでは**図 18** の手書きストロークの描画速度をそれぞれ 3 倍，1 倍，$\frac{1}{3}$ 倍にした模擬手書きストロークを生成し，それぞれを提案手法で整形した．その結果を**図 19**，**図 20**，**図 21** に示す．図 19 では，描画速度の速い手書きストロークに対してファジネスの大きな FSC が生成され，これに応じて，整形結果を構成する幾何曲線数は減り，手書きストロークが大胆に単純化された整形形状となっている．この結果は素早く雑な

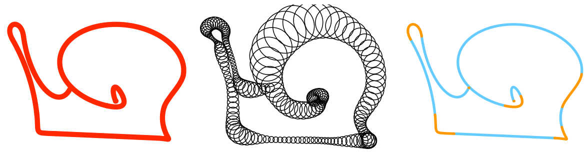


図 19　「かたつむり」（描画速度 3 倍）の整形
Fig. 19　Shaping for “Snail” with 3x drawing speed.

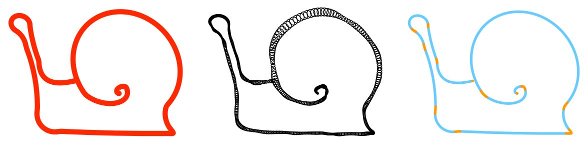


図 20　「かたつむり」（描画速度 1 倍）の整形
Fig. 20　Shaping for “Snail” with 1x drawing speed.

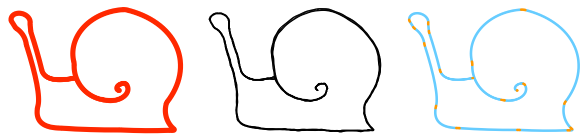
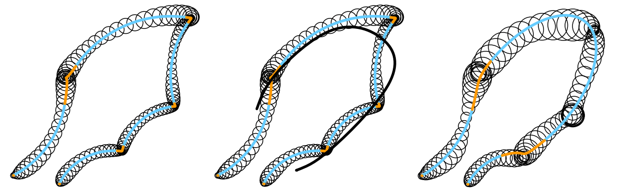


図 21　「かたつむり」（描画速度 $\frac{1}{3}$ 倍）の整形
Fig. 21　Shaping for “Snail” with $\frac{1}{3}$x drawing speed.

手書きストロークに対して，提案手法が手書きストロークの詳細な形状を省略する効果を持つことを示す．一方で，図 21 では，描画速度の遅い手書きストロークに対してファジネスの小さな FSC が生成され，これに応じて，整形結果を構成する幾何曲線数は増え，手書きストロークに忠実な整形形状となっている．この結果はゆっくりとした丁寧な手書きストロークに対して，提案手法が手書きストロークの詳細な形状を整形形状に残す効果を持つことを示す．したがって，ユーザがこれらの効果を利用すれば，描画動作の丁寧さの程度を変化させることで整形形状の詳細度を制御することが可能となる．

4.4　重ね書きによる整形形状の逐次的修正

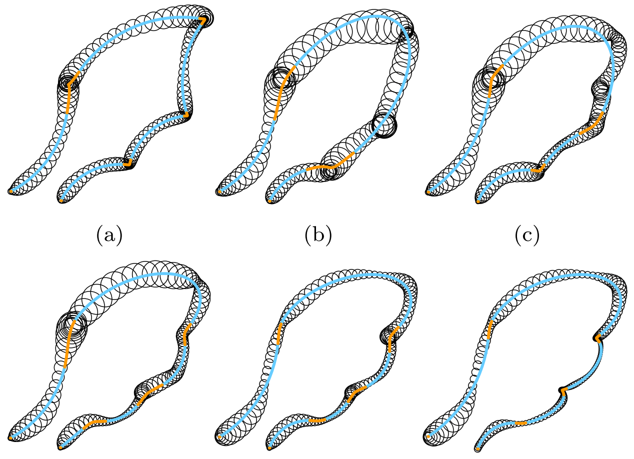
描画ストロークを重ね書きしながら整形結果を逐次的に修正する実験を行った．実験に際して，FSCI による幾何作図機能に提案手法による自由曲線整形機能および文献 [6] の S-FSCG による逐次型 FSC 生成機能を備えた作図アプリケーションを構築し，インタラクティブな作図環境を実現した．このアプリケーションでは，ユーザが 1 本目の手書きストロークを入力すると，その FSC および整形結果が表示される．それ以降は，**図 22** (a) のような既存の FSC に対して，ユーザが図 22 (b) のように重ね書きストロークを入力すると，S-FSCG によって既存の FSC と重ね書きストロークの FSC が融合され，図 22 (c) のように融合後の FSC および整形結果が表示される．重ね書きによる整形結果の修正過程の一部を抜粋して**図 23** に示す．ここではまず，図 23 (a) に対して素早く雑に重ね書きすることで，図 23 (b) のように FSC のファジネスを大きくし，整



(a) 既存の FSC および整形結果　(b) 図 22a に対する重ね書きストローク　(c) 融合後の FSC および整形結果

図 22　作図アプリケーションの動作例

Fig. 22　Behavior of the drawing application.



(a)　(b)　(c)

(d)　(e)　(f)

図 23　重ね書きによる整形結果の逐次的修正の例

Fig. 23　Updates of shaped results by oversketching.

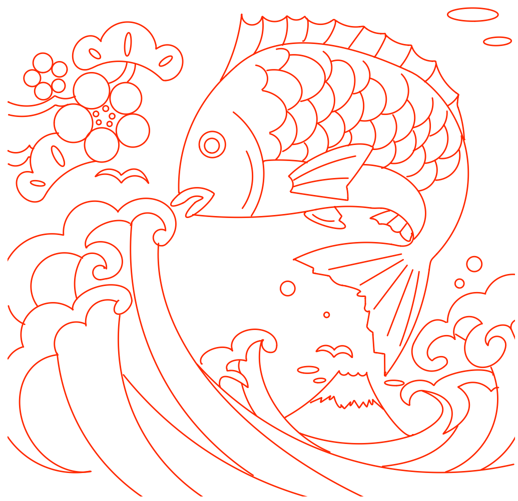
形結果を単純化している．その後，ゆっくりとした丁寧な重ね書きを繰り返すことで，図 23 (c) から図 23 (f) までのように FSC のファジスを局所的に小さくしながら，整形結果の詳細な形状を修正している．この結果は，S-FSCG を併用することで重ね書きによる整形結果の逐次的修正が可能であることを示す．

4.5　実際的な幾何作図における自由曲線整形法の適用例

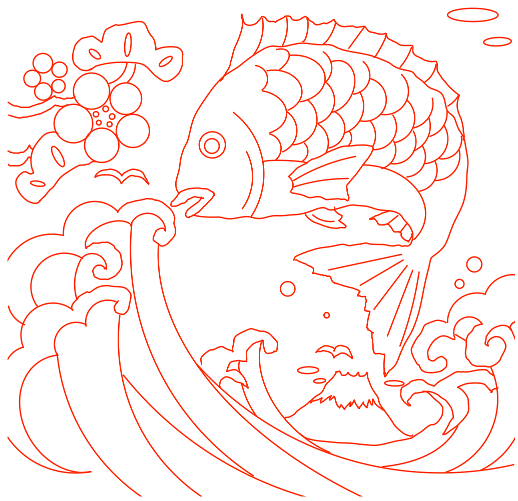
実際的な幾何作図における提案手法の効果を確認する実験を行った．実験ではユーザが 4.4 節の作図アプリケーションで重ね書きを繰り返しつつ，複数の幾何曲線から構成される幾何作図を完成した．入力されたすべての手書きストロークを**図 24** (a) に，幾何作図の結果を図 24 (b) に示す．一方比較のために，*FC*，*FO* に対して提案手法による整形を行わなかった場合の作図結果を図 24 (c) に示す．

図 24 (c) では「魚の輪郭」(*FO*)，「波頭」(*FO*)，「松の葉」(*FC*，*FO*) などの部分に手書きストロークのうねりが残っているが，提案手法による自由曲線整形を行った図 24 (b) ではこれらの *FC*，*FO* のうねりが解消され，周囲の幾何曲線と調和していることが分かる．一方で，「尾びれ」(*FO*) や「冠雪」(*FO*) の部分では整形前の詳細な形状が整形後も保持されている．これはユーザが，ゆっくり丁寧な手書きストロークを入力することで，整形結果が

(a) 入力されたすべての手書きストローク



(b) *FC*，*FO* への整形を行った場合の同定幾何曲線



(c) *FC*，*FO* への整形を行わない場合の同定幾何曲線

図 24　実際的な幾何作図における自由曲線整形の効果

Fig. 24　Effect of free curve shaping in practical geometric drawing.

情報処理学会論文誌　Vol.59　No.12　2288–2298 (Dec. 2018)

手書きストロークに忠実な形状となるように意図的に整形形状の詳細度を制御した結果である．

5. 結言

本論文では，FSCI の幾何曲線同定機能を利用した自由曲線の円錐曲線列化アルゴリズムとその円錐曲線列の平滑化アルゴリズムを構築し，これに基づいた手書き自由曲線整形を提案した．また，整形形状，整形処理時間，整形形状の詳細度の制御機能，整形形状の修正機能の観点から，提案手法がインタラクティブな手書き幾何作図に適した性質を持つことを実験的に示した．

今後，自由曲線整形結果の CS に対するファジィグリッドスナッピング [12] の適用法を検討したうえで，提案手法を手書き作図インタフェース [7] の自由曲線整形機能として本格的に実装する予定である．これにより，実用的な幾何作図の中で提案手法の有用性について評価，検討する．

参考文献

[1] Qin, S.-F., Wright, D.K. and Jordanov, I.N.: On-line segmentation of freehand sketches by knowledge-based nonlinear thresholding operations, *Pattern Recognition*, Vol.34, No.10, pp.1885–1893 (2001).

[2] 八木麻理子，川田洋平，藤澤　誠，三浦憲二郎：ペンタブレット入力による G^1 連続を持つ美的曲線セグメント列の生成，芸術科学会論文誌，Vol.7, No.3, pp.97–101 (2008).

[3] 森本有紀，高橋時市郎：デザインの原理を用いた自由形状のイラスト美化手法，情報処理学会論文誌，Vol.56, No.5, pp.1329–1338 (2015).

[4] Thiel, Y., Singh, K. and Balakrishnan, R.: Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2D curves, *UIST '11 Proc. 24th Annual ACM Symposium on User Interface Software and Technology*, pp.383–392 (2011).

[5] 佐賀聡人，牧野宏美，佐々木淳一：ファジースプライン曲線同定法，電子情報通信学会論文誌 D，Vol.J77-D2, No.8, pp.1620–1629 (1994).

[6] 佐藤洋一，安福尚文，佐賀聡人：スケッチによる作図インタフェースのための逐次型ファジースプライン曲線生成法，電子情報通信学会論文誌 D，Vol.J86-D2, No.2, pp.242–251 (2003).

[7] 河合良太，西川　玲，佐賀聡人：手書きスケッチ入力フロントエンドプロセッサ：SKIT，電子情報通信学会論文誌 D，Vol.J88-D2, No.5, pp.897–905 (2005).

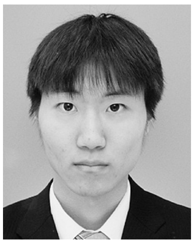
[8] Baran, I., Lehtinen, J. and Popović, J.: Sketching Clothoid Splines Using Shortest Paths, *Computer Graphics Forum*, Vol.29, No.2, pp.655–664 (2010).

[9] 佐賀聡人，牧野宏美，佐々木淳一：手書き曲線モデルの一構成法—ファジースプライン補間法，電子情報通信学会論文誌 D，Vol.J77-D2, No.8, pp.1610–1619 (1994).

[10] 大川哲也，佐賀聡人：手書き曲線同定法 FSCI におけるファジネス生成モデルの精密化，電子情報通信学会論文誌 D，Vol.J82-D1, No.5, pp.634–643 (1999).

[11] Zadeh, L.A.: Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems*, Vol.1, pp.3–28 (1978).

[12] Dematapitiya, S., Kawazoe, M., Nishikawa, A., Sakurai, M. and Saga, S.: Snapping of Fuzzy Objects Using the Multi-Resolution Fuzzy Grid Snapping Technique, *IPSJ*, Vol.50, No.2, pp.904–915 (2009).

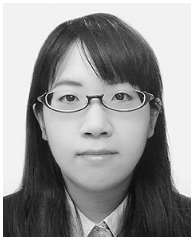


伊藤 友彦　（学生会員）

1995 年生．2017 年室蘭工業大学情報電子工学系学科卒業．同年同大学大学院修士課程進学．手書きインタフェースの研究に従事．電子情報通信学会学生員．

神谷 葉月　（正会員）

1992 年生．2017 年室蘭工業大学大学院工学研究科情報電子工学系専攻修士課程修了．同年（株）朋栄入社．在学中，手書き幾何作図のための自由曲線整形の研究に従事．



油谷 凛

1991 年生．2015 年室蘭工業大学大学院工学研究科情報電子工学系専攻修士課程修了．同年東洋印刷（株）入社．在学中，手書き幾何作図のための自由曲線整形の研究に従事．

佐賀 聡人　（正会員）

1960 年生．1987 年北海道大学大学院工学研究科博士後期課程（電子工学専攻）修了．1987〜1989 年青年海外協力隊員としてフィリピン，ナガ大学でコンピュータサイエンスを指導．1989 年（株）テクノバ入社．図形処理システムのヒューマンインタフェースに関する研究に従事．1994 年室蘭工業大学工学部講師．現在，同大学大学院工学研究科教授．工学博士．ソフトコンピューティングに基づくヒューマンインタフェースの研究に従事．IEEE 会員．