形状と位相の編集が可能な手書きCADインターフェースの実現に関する研究

形状と位相の編集が可能な
手書き CAD インターフェースの実現に関する研究

伊藤 友彦

2022 年 6 月 30 日

目次

概要

デジタルペンタブレットやタッチディスプレイの普及に伴い，スケッチベースのインタラクティブな CAD（Computer-aided design）ユーザインターフェースが注目を集めている．しかしながら，既存のインターフェース SKIT（Sketch Input Tracer）では，重ね書きストロークの入力による編集操作は，一度に単一の幾何オブジェクトしか対象とすることができないという限界がある．この限界を克服するため本研究ではまず準備として，「自由曲線整形」，「サブ曲線同定」，「IFGS」という要素技術を提案し，SKIT の要素技術を拡充する．その上で，2 次元 CAD システムで使用するための汎用的なスケッチベースのインターフェースを新たに提案する．これにより，重ね書きによって複数の幾何オブジェクトの形状と位相を同時に修正することができるスケッチベースの編集操作が実現される．提案インターフェースは，SKIT のファジィ論理ベースの戦略に基づいて開発される．これにより，汎用手書き CAD インターフェースで必要となる手書き操作が網羅されることとなる．提案インターフェースを使用することで，ユーザはラフスケッチから始めて，重ね書きを繰り返しながら詳細なデザインを作成するといった作図を創造的に行うことが可能となる．実験により，熟練者が提案インターフェースを制御することができ，初心者がその制御に慣れることが可能であることを示す．

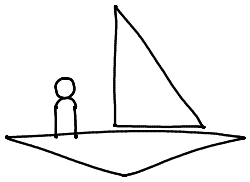
第 1 章

序論

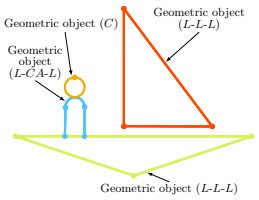
1.1 本研究の背景と目的

従来の 2D の CAD システムでは，幾何図形は基本的に，線分や円弧などのいくつかのクラスの幾何プリミティブの列として表される幾何オブジェクトで構成される．実際，標準的な 2D ベクターグラフィックス形式である SVG（Scalable Vector Graphics）では，幾何図形は矩形，ポリライン，ポリゴン，パスなどの幾何オブジェクトで構成され，これらのそれぞれは，線分，楕円弧，2 次または 3 次 Bézier 曲線といった幾何プリミティブの列として表現される．また，CAD 分野のデファクトスタンダードである DXF（Drawing Interchange Format）では，2D の幾何図形はポリラインやリージョンを含む幾何オブジェクトで構成され，それぞれが線分，円，円弧，楕円，楕円弧，スプライン曲線といった幾何プリミティブの列として表現される．これらの表現は，幾何オブジェクトを，互いに論理的に接続された幾何プリミティブで構成された構造として扱う際に重要な役割を果たしている．CAD システムのユーザは，メニュー選択やコマンド入力などのユーザインターフェースを介して幾何オブジェクトの入力や編集を行う．

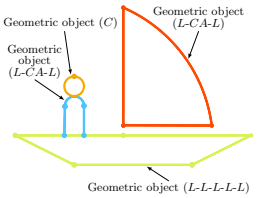
デジタルペンタブレットやタッチディスプレイの普及に伴い，スケッチベースのインタラクティブな CAD（Computer-aided design）ユーザインターフェースが注目を集めている．既存のスケッチベースインターフェースの 1 つである SKIT（Sketch Input Tracer）[1] では，ユーザは 7 クラスの幾何プリミティブ（線分（L），円（C），円弧（CA），楕円（E），楕円円弧（EA），閉自由曲線（FC），開自由曲線（FO））の列として構成された幾何オブジェクトを入力することができる．たとえば，図 1.1(a) の手書きストロークを描画することにより図 1.1(b) に示されているような幾何オブジェクトを入力することができる．ここでユーザはさらに図 1.1(b) の幾何オブジェクトを重ね書きによって変形操作，分割操作，統合操作といった編集操作を繰り返すことで図 1.1(c) のように修正することができる．しかしながら，SKIT では重ね書きストロークの入力による編集操作は，一度に単一の幾何オブジェクトしか対象とすることができないという限界があるため，複数の幾何オブジェクトを横断する連結操作や分解操作を行うことができない．したがってこれらの操作を繰り返すことにより図 1.1(c) の幾何オブジェクトを図 1.1(d) の幾何オブジェクトに修正することはできない．



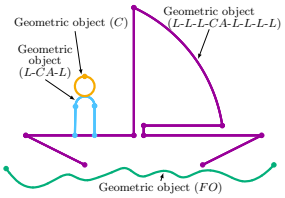
(a) 手書きストローク



(b) 作成された幾何オブジェクト



(c) 個別の幾何オブジェクトに対する編集操作の結果



(d) 複数の幾何オブジェクトを横断する編集操作の結果

図 1.1: 幾何オブジェクトに対する編集操作

この限界を克服するため本研究では，2 次元 CAD システムで使用するための汎用的なスケッチベースのインターフェースを新たに提案する．これにより，重ね書きによって複数の幾何オブジェクトの形状と接続関係を同時に修正することができるスケッチベースの編集操作が実現される．提案インターフェースは，ファジィ論理に基づく SKIT の戦略に基づいて開発される．提案インターフェースのユーザはラフスケッチから始めて，重ね書きを繰り返しながら詳細なデザインを作成するといった創造的な作図を行うことが可能となる．また，提案インターフェースにより，汎用手書き CAD インターフェースで必要となる手書き操作が網羅されることとなる．

1.2 関連研究と本研究の位置付け

イラストレーションや幾何図形を入力および編集するために，多数の手書きインターフェースが模索されてきた．いくつかの研究 [2, 3, 4, 5, 6] は描画された複数の手書きストロークを集約することによって美化されたイラストを作成するための手書きインターフェースを提案した．さらに，重ね書きストロークが描画されるたびに既存のイラストレーションをインタラクティブに修正するための編集操作を実現するためにいくつかの手書きのインターフェース [7, 8, 9, 10, 11, 12] が提案された．ただし，これらのインターフェースは 2D の CAD システムでの使用を目的としたものではない．

2D の CAD システムでの汎用使用を目的とした手書きインターフェースは，7 クラスの幾何プリミティブ，すなわち，L，C，CA，E，EA，FC，FO，のすべてを包括的に入力できる必要がある．このことは，SVG と DXF のそれぞれの幾何プリミティブが基本的にこれらの 7 つのクラ

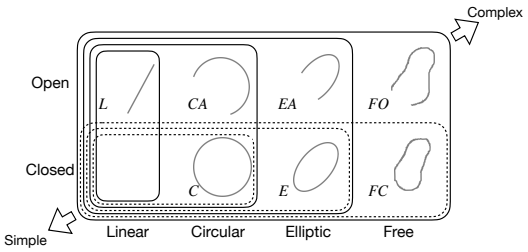


図 1.2: 曲線クラス間の自由度に関する包含関係

スに分類できるという事実から理解できる[*1]．幾何オブジェクトの入力または編集操作を実現するために 2D の CAD システムで使用できるいくつかの手書きインターフェースが提案されている．Hse と Newton[13] は，シンボル認識に基づいて事前定義された幾何オブジェクトの入力操作を実現した．Arvo と Novins[14] は，カーブフィッティングに基づいて，円，軸に沿ったボックス，線分の入力操作を実現した．Jorge と Fonseca[15] は，描かれた手書きストロークの凸包の長さや面積などの幾何学的特徴に基づいて，線分，円，三角形，長方形，楕円，ダイヤモンドの入力操作を実現した．Yu と Cai[16] は，カーブフィッティングと手書きストロークのセグメンテーションを組み合わせて，線分，円，円弧，楕円，ポリライン，ヘリックスで構成される複雑な幾何オブジェクトの入力操作を実現した．Paulson と Hammond[17] は，この手法を拡張して 5 次曲線やスパイラルを含む幾何オブジェクトの入力操作を実現した．五十嵐ら [18] や Fiser ら [19] は，幾何学的制約に基づいて整形および配置された幾何オブジェクトの入力操作を実現した．[18] の手書きインターフェースでは線分のみで構成される幾何オブジェクトの入力が可能であるのに対し，[19] の手書きインターフェースでは線分，円，円弧，3 次 Bézier 曲線で構成される幾何オブジェクトの入力が可能である．McCrae と Singh[20] ならびに Baran ら [21] は曲率の連続性に基づいて線分，円弧，クロソイド曲線からなる滑らかな幾何オブジェクトの入力操作と，重ね書きによる個々の幾何オブジェクトの編集操作を実現する手書きインターフェースを提案した．しかしながら，以上のインターフェースはいずれも 7 クラスの幾何プリミティブをカバーしておらず，手書き操作についても包括的にはサポートしていない．それゆえに，これらは 2D の CAD システムでの汎用使用には不十分である．

一方，佐賀らによって提案された FSC 同定法 [22, 23] はユーザの描画動作からユーザの描画意図を推測する．FSC 同定法は，手書きストロークを 7 クラスの幾何プリミティブのいずれかとして同定することができる．ここで図 1.2 に示すように，7 つの曲線クラスの間には，（たとえば，EA には特殊なケースとして CA と E が含まれるといった）自由度に関する包含関係が存在することに注意する．したがって，手書きストロークの形状のみを考慮すると，手書きストロークをユーザの意図に応じて 7 クラスのいずれかに分類する際に本質的な問題が発生する．FSC 同定法は，雑な手書きストロークを単純な曲線クラスの幾何プリミティブとして，丁寧な手書きストロー

[*1] SVG と DXF は幾何プリミティブとして Bézier 曲線やスプライン曲線を含む場合があるが，それらは閉自由曲線（FC）または開自由曲線（FO）として分類でき，NURBS として統一的に表現できるものである．

クを複雑な曲線クラスの幾何プリミティブとして分類するという戦略を通じて，これらの問題を克服する．この戦略を実装するために，FSC 同定法はファジィ論理を使用して，手書きストロークの形状だけでなく，ペンの速度や加速度などの描画の動的特性から推定されたあいまいさに関する情報も使用する．[13, 14, 15, 16, 17, 18, 19, 20, 21] の手書きインターフェースにはそのような戦略が含まれていないため，上記の問題を克服することはできない．

その後，河合ら [1] は FSC 同定法を基にした一連の研究 [22, 24, 23, 25, 26, 27] を通じて，SKIT と呼ばれる汎用手書き CAD インターフェースを開発し，前述した編集操作 1 から 5 までを実現した．SKIT を使用すると，ユーザは，重ね書きストロークで描画意図を表現しながら，7 クラスの幾何プリミティブで構成される個別の幾何オブジェクトを作成できる．

Qin ら [28] の手書きインターフェースと，FSC 同定法を基に開発された FFDS（Fuzzy Freehand Drawing System）[29] は，手書きストロークを 7 クラスの幾何プリミティブに分類できる手書きインターフェースとして提案されている．しかしながら，重ね書き編集操作については議論されていない．Wang ら [30] の手法は，重ね書きされた手書きストロークから線分と円錐曲線（円，円弧，楕円，楕円円弧を含む）で構成される幾何オブジェクトを生成できる手書きインターフェースとして提案された．しかしながら，それぞれの手書きストロークを個別に幾何プリミティブとして認識し，それらをグループ化して統合するため，幾何オブジェクトを変形，分割するような編集操作が実行できない．したがって，この手法は，図 1.1(b) を図 1.1(c) に変化させるタイプの幾何オブジェクトの修正には適用できない．

本研究は，SKIT のファジィ論理ベースのストラテジを継承しつつ SKIT を拡張的に再構成することで，複数の幾何オブジェクトを横断して自在に編集することが可能な手書きインターフェースを提案する．したがって，本研究の主なコントリビューションは，汎用手書き CAD インターフェースで必要となる単一または複数の幾何オブジェクトの入力操作，形操作，分割操作，統合操作，連結操作，分解操作およびそれらの複合操作といった手書き操作を網羅するインターフェースをはじめて実現することである．

1.3 本論文の構成

本論文では，新たなインターフェースの提案に先立ち，まず 2 章から 4 章までで以下の提案を行い，従来の SKIT の要素技術を拡充する．

2 章 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：
閉自由曲線または開自由曲線が整形されず他の円錐曲線と調和しないという問題を解決するため，手書き幾何作図に適する自由曲線整形の提案する．

3 章 FSC 同定法に基づく $n/4$ 円弧および $n/4$ 楕円弧の同定：
円弧／楕円弧が浅い時に中心角の量子化が困難となり，正方グリッドに合わせて整形できないという問題を解決するため，円弧／楕円弧を $\frac{n}{4}$ 円弧／楕円弧または一般円弧／楕円弧として同定するサブ曲線同定を提案する．

4 章 ファジィ理論に基づく無限解像度グリッドスナッピング技術：

MFGS（Multi-resolution fuzzy grid snapping）[31] では有限範囲の解像度のグリッドしか扱えず，その範囲外ではスナッピングが適正に動作しないという問題を解決するため，無限解像度グリッドから適切な解像度のスナッピング先を効率的に探索する IFGS（Infinite-resolution fuzzy grid snapping）を提案する．

その後，5 章で以下の提案を行うことで，汎用手書き CAD インターフェースで必要となる手書き操作を網羅する．

5 章 ファジィ論理に基づいて複数のオブジェクトを横断したジオメトリ・トポロジ編集を実現するインタラクティブな手書き *CAD* インターフェース：

従来の SKIT の要素技術を拡張的に再構成することで，複数の幾何オブジェクトを横断的に編集することが可能な手書きインターフェースを新たに提案する．さらにここでは実験により，提案されたインタラクティブな描画インターフェースが熟練者にとって制御可能であり，初心者でもこの制御に慣れることが可能であることが示される．

最後に，6 章では，2 章から 5 章で得られた結果をまとめ，本研究の結論を示す．

第 2 章

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

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，佐賀ら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 クラスの幾何プリミティブ（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 (FSC 同定法) 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．本章では，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 クラスの幾何プリミティブ全てが調和した幾何作図を行える手書きインターフェースが実現される．

2.1 本章の目的

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伴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 [28]，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 [32]，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 [33]，手書きストロークの速度情報を利用

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 [34]，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，佐賀ら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 クラスの幾何プリミティブ（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（FSC 同定法）[22] を提案した．また，複数の手書き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）[25] 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ーフェース [1] を実現した．しかし，文献 [22] 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．

本章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) はインター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 FSC 同定法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 [21] に手書きストロークを線分，円弧，クロソイド曲線の 3 クラスの幾何プリミティブ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，FSC 同定法の幾何プリミティブ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

[*1] 一般に，「線分」，「円」，「円弧」，「楕円」，「楕円弧」，「閉自由曲線」，「開自由曲線」という幾何プリミティブクラスによってクラス分けを行った場合，幾何プリミティブ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，本章では文献 [22] 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．

示す．

2.2 インタラクティブな手書き幾何作図の概要

2.3 で提案する自由曲線整形法は，FSC 同定法による手書き幾何プリミティブ同定機能および S-FSCG による重ね書き幾何プリミティブ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 FSC 同定法の幾何プリミティブ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 FSC 同定法と S-FSCG に焦点を置いて概説する．

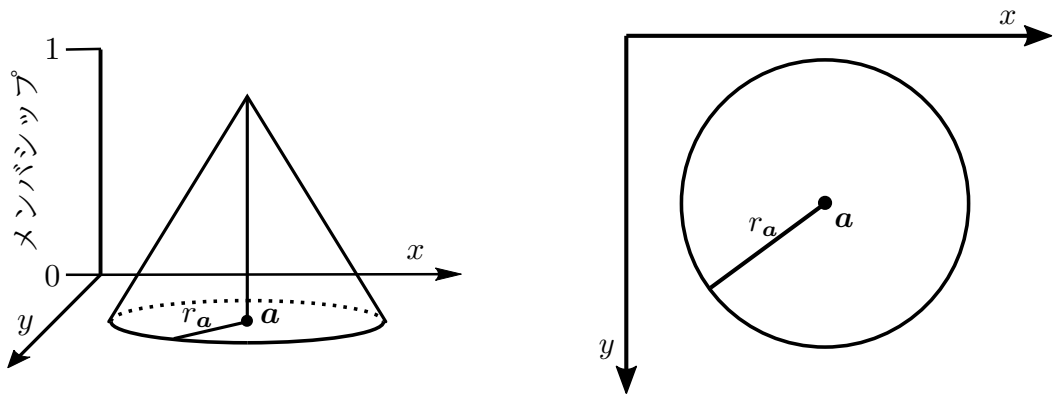
2.2.1 FSC 同定法による幾何プリミティブの手書き入力

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 クラスの幾何プリミティブ（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プリミティブの手書き入力が実現される．

2.2.1.1 FSC の生成

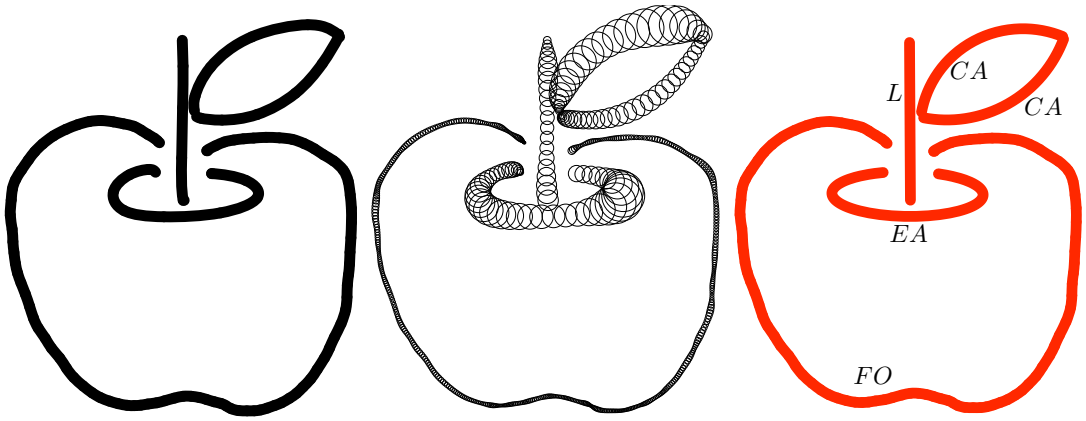
時系列点列（点の位置と時刻との組を要素とする列）として入力された手書きストロークをファジィスプライン補間 [24] することで，手書きストロークの描画軌跡およびその位置のあいまいさを表現するファジィな曲線 FSC $\tilde{\boldsymbol{s}} : [a, b] \to F$ を生成する．ただし，a は手書きストロークの開始時刻，b は終了時刻，F は円錐型ファジィ点全体の集合であり，FSC $\tilde{\boldsymbol{s}}$ は時刻 $t \in [a, b]$ に対応する円錐型ファジィ点 $\tilde{\boldsymbol{s}}(t)$ を返す写像である．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 2.1 のような円錐型のメンバシップ関数により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 [35] 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 2.2a の 5 つの手書きストロークに対して図 2.2b の 5 つの FSC が生成される．このとき，図 2.3a に示す形状の似た手書きストロークであっても，図 2.3b 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ジネスの小さな FSC が生成されるという効果が得られる．



(a) 射影図　　(b) 上面図

図 2.1: 円錐型ファジィ点のメンバシップ関数



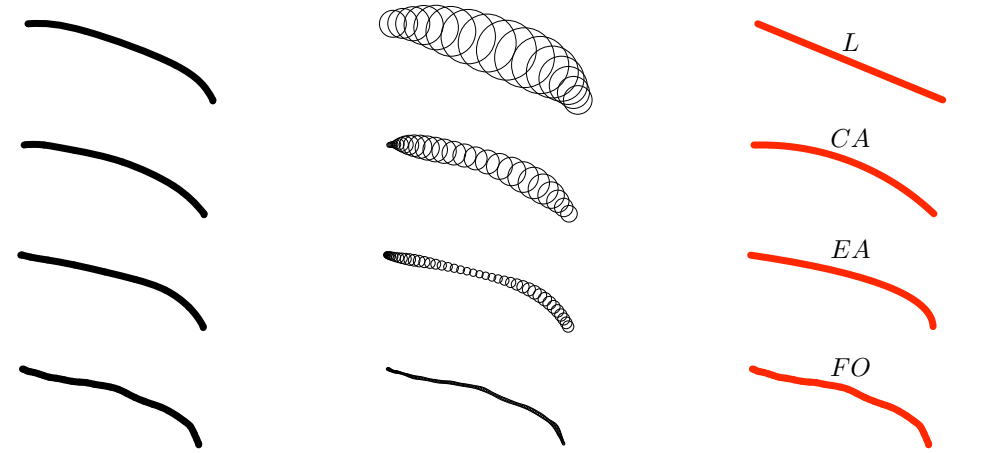
(a) 手書きストローク　　(b) FSC　　(c) 同定幾何プリミティブ

図 2.2: FSC 同定法による幾何作図の例

2.2.1.2　幾何プリミティブ同定

得られた $\tilde{\boldsymbol{s}}$ に対して，表 2.1 のファジィ推論規則にしたがって，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^{Linear}，$P^{Circular}$，$P^{Elliptic} \in [0,1]$ はそれぞれ $\tilde{\boldsymbol{s}}$ が線形，円形，楕円形である可能性値，$P^{Closed} \in [0,1]$ は $\tilde{\boldsymbol{s}}$ 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 [36] 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^{Linear}，$P^{Circular}$，$P^{Elliptic}$，P^{Closed} は，具体的には文献 [22] で提案されている FSC 同定法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．$\wedge$ は論理積であり，min 演算で計算される．

次に，L，C，CA，E，EA，FC，FO の中でもっと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プリミティブを同定幾何プリミティブとして出力する．ここ



(a) 手書きストローク　(b) FSC　(c) 同定幾何プリミティブ

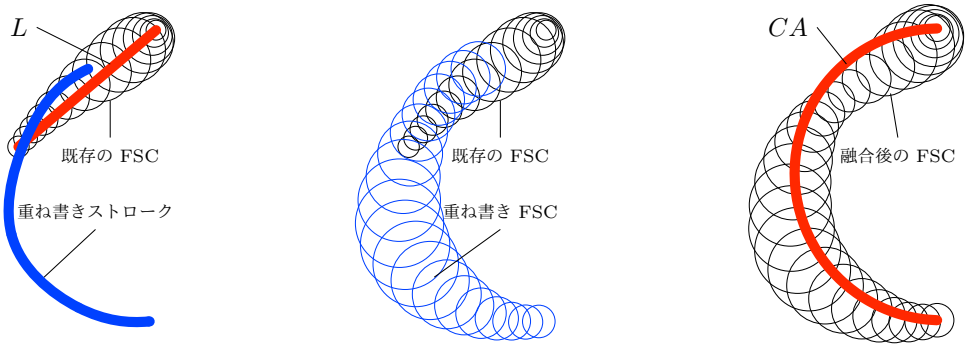
図 2.3: FSC のファジネスの変化による幾何プリミティブ同定結果の違い

表 2.1: 7 クラスの幾何プリミティブ同定のためのファジィ推論規則

$\mu(L)$	$=$	P^{Linear}						
$\mu(C)$	$=$	$\neg P^{Linear}$	$\wedge$	$P^{Circular}$			$\wedge$	P^{Closed}
$\mu(CA)$	$=$	$\neg P^{Linear}$	$\wedge$	$P^{Circular}$			$\wedge$	$\neg P^{Closed}$
$\mu(E)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$P^{Elliptic}$	$\wedge$	P^{Closed}
$\mu(EA)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$P^{Elliptic}$	$\wedge$	$\neg P^{Closed}$
$\mu(FC)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$\neg P^{Elliptic}$	$\wedge$	P^{Closed}
$\mu(FO)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$\neg P^{Elliptic}$	$\wedge$	$\neg P^{Closed}$

で，同定幾何プリミティブは，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．

以上の処理により，たとえば，図 2.2b のような 5 つの FSC に対して図 2.2c のように 5 つの幾何プリミティブが同定される．ここで，表 2.1 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もっと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 2.3 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プリミティブ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プリミティブ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．



(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) 重ね書き後の幾何プリミティブ同定

図 2.4: S-FSCG による幾何プリミティブの重ね書き修正の例

2.2.2　S-FSCG による幾何プリミティブの重ね書き修正

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 FSC 同定法と併用すれば，たとえば図 2.4 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プリミティブ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．

2.3　FSC 同定法を利用した自由曲線整形法の提案

図 2.2c 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 [22] 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 2.1 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，FSC 同定法で FC，FO と同定されたものを対象とする．ここでは，FSC 同定法で FC，FO と同定された手書きストロークに対して再度 FSC 同定法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 FSC 同定法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，FSC 同定法を用いているため，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，2.2.2 で述べたように FSC 同定法は 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）．

表 2.2: 4 クラスの幾何プリミティブ同定のためのファジィ推論規則

$\mu(L)$	$=$	P^{Linear}				
$\mu(CA)$	$=$	$\neg P^{Linear}$	$\wedge$	$P^{Circular}$		
$\mu(EA)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$P^{Elliptic}$
$\mu(FO)$	$=$	$\neg P^{Linear}$	$\wedge$	$\neg P^{Circular}$	$\wedge$	$\neg P^{Elliptic}$

以下では，まず FSC 同定法を利用して円錐曲線を同定する方法を 2.3.1 で示した上で，円錐曲線列化と平滑化のアルゴリズムを 2.3.2 および 2.3.3 で示す．

2.3.1 FSC 同定法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 つの開いた幾何プリミティブクラスをまとめて**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，FSC 同定法のファジィ推論規則を表 2.1 に示したオリジナルのものから表 2.2 に示すものに置き換える．これにより，FSC 同定法は L, CA, EA, FO の 4 クラスの幾何プリミティブ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 に分類することとすれば，FSC 同定法は分類結果として 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．

2.3.2 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上で，それぞれの部分 FSC を FSC 同定法で同定することで，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 FSC 同定法により CS と同定される．
2. 条件 (1) を満たした上で，分割数（分割後の部分 FSC の数）が最小となる．
3. 条件 (2) を満たした上で，円錐曲線列性（以下で定義）が最大となる．

ここで，(2) は 2.2.1 の FSC 同定法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もっと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．

[*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．

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 FSC 同定法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 FSC 同定法に入力したときに出力される円錐曲線性 $\mu(CS)$（2.3.1 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 $\tilde{\boldsymbol{s}}_{t_{p_{k-1}},t_{p_k}}$ $(1 \leq k \leq m)$ がすべて FSC 同定法によって 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 $\tilde{\boldsymbol{s}}_{t_i,t_j}$ 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 $\tilde{\boldsymbol{s}}_{t_{p_{k-1}},t_{p_k}}$ $(1 \leq k \leq m)$ 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}：

$\mu^{P} = \mu_{0,n-1}^{P}$ 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 $\tilde{\boldsymbol{s}}_{t_{p_{i-1}},t_{p_i}}$ $(1 \leq i \leq m)$ のそれぞれを FSC 同定法で同定した結果得られる幾何プリミティブ（円錐曲線）を，$\boldsymbol{q}_i^{P} : [0,1] \to \mathbb{E}^2$ $(0 \leq i \leq m-1)$ 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 $\boldsymbol{b}_{i\ j}^{P}$ $(0 \leq j \leq 2)$ を用いて，以下

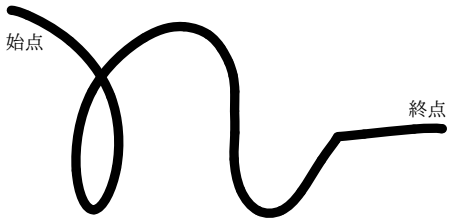


図 2.5: 手書きストローク

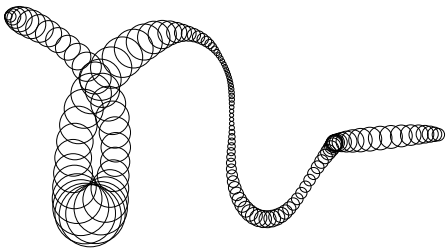


図 2.6: FSC

の 2 次有理 Bézier 曲線形式で表される[*4].

$$\boldsymbol{q}_i^P(t) = \frac{B_0^2(t)\boldsymbol{b}_i^P{}_0 + B_1^2(t)w_i^P\boldsymbol{b}_i^P{}_1 + B_2^2(t)\boldsymbol{b}_i^P{}_2}{B_0^2(t) + B_1^2(t)w_i^P + B_2^2(t)}$$

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上で，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 FSC 同定法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 $\mu_{i,j}(CS) \geq \mu_{i',j'}(CS)$ 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．

2.3.2.1 FSC の生成

一筆書きの手書きストロークから 2.2.1 と同様に FSC $\tilde{\boldsymbol{s}}$ を生成する．たとえば図 2.5 の手書きストロークから図 2.6 のような FSC が生成される．

[*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．

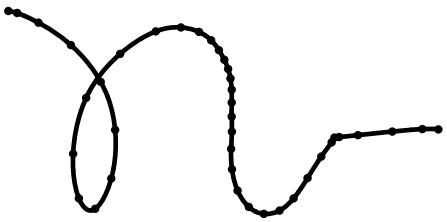


図 2.7: 探索点列

2.3.2.2 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 2.7 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．

2.3.2.3 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もっと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，終点自身が分割点となるまで順次

$$p_i^L = \max\{p \mid p \in I, p > p_{i-1}^L, r_{p_{i-1}^L, p} = CS\} \tag{2.1}$$

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もっと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

$$p_i^R = \min\{p \mid p \in I, p < p_{i+1}^R, r_{p, p_{i+1}^R} = CS\} \tag{2.2}$$

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 $A_i = \{p \mid p_i^R \leq p \leq p_i^L\}$ $(0 \leq i \leq m)$ と絞り込む[*7]．たとえば，図 2.7 の探索点列から分割点候補を絞り込むと図 2.8 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．

[*5] 本章では $n = 40$ 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 (2.1) 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，もっとも終点側にある 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 (2.2) では，式 (2.1) 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，もっとも始点側にある 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 $(0 \leq i \leq m)$ は互いに共通の要素を持たないことになる．

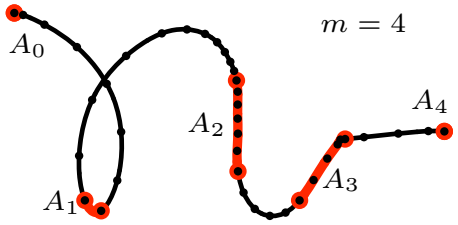


図 2.8: 分割点の候補

2.3.2.4 $\tilde{s}$ 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

2.3.2.3 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 $\tilde{\boldsymbol{s}}$ 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．ここで，P_j は部分 FSC $\tilde{\boldsymbol{s}}_{t_0,t_j}$ 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

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

を満たすインデックスを探索する．このとき，2.3.2.3 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，FSC 同定法の同定特性（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

[*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．

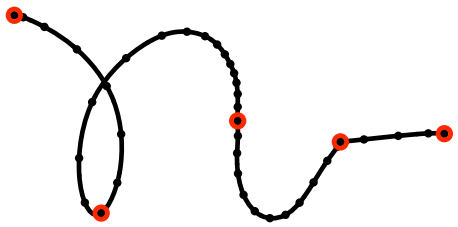


図 2.9: 分割点列

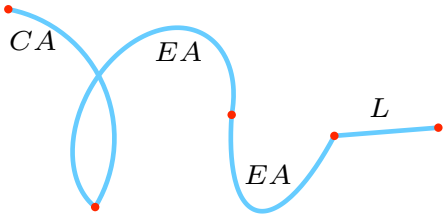


図 2.10: 円錐曲線列

$f : I \to [0, 1]$ 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 $k = \text{find}(p_{i-1}^R, p_{i-1}^L)$ と求める．

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

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 $\tilde{\boldsymbol{s}}$ 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 2.9 に示す．

2.3.2.5 円錐曲線列の生成

2.3.2.4 で得られた $\tilde{\boldsymbol{s}}$ 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 2.10 に示す．

[*9] $\tilde{\boldsymbol{s}}$ 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 はそのうちの一つとなる．

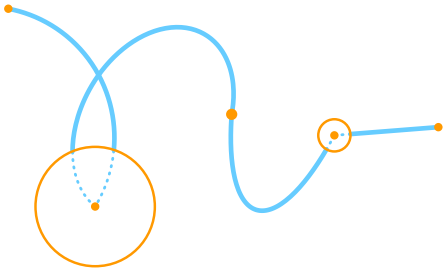


図 2.11: 刈り込み円

図 2.12: 整形結果

2.3.3 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 2.11 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 2.12 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．

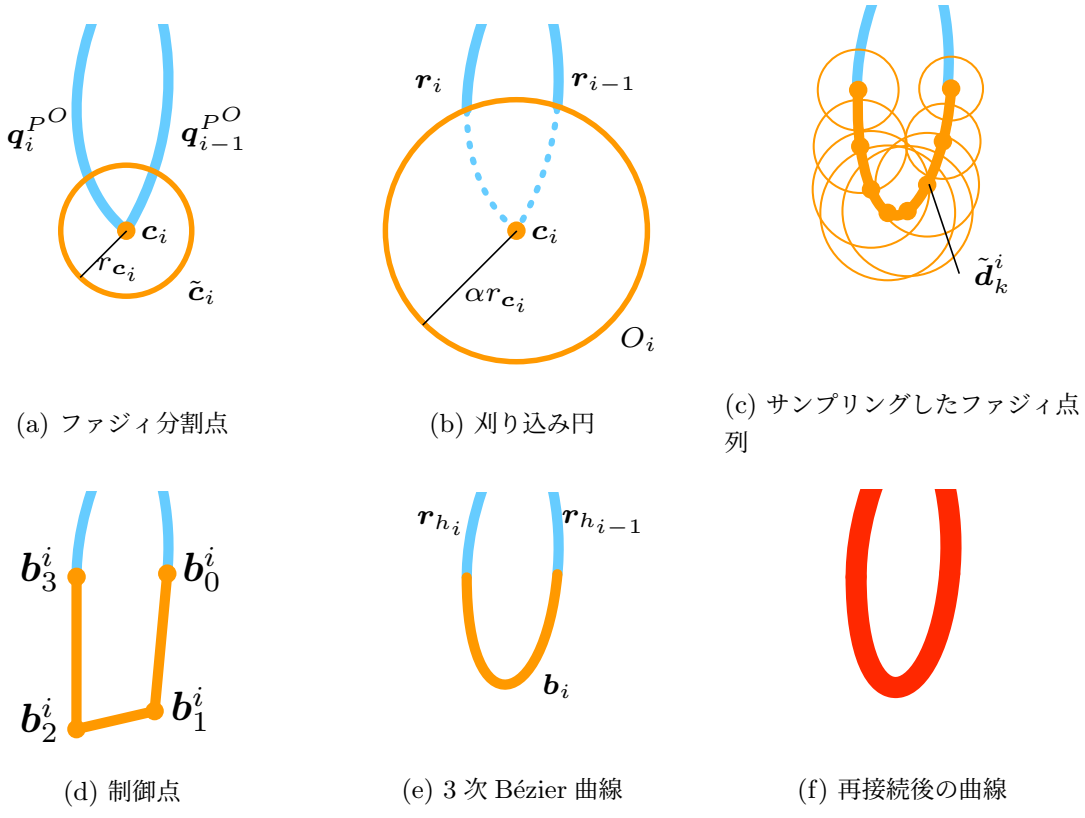
2.3.3.1 ファジィ分割点における刈り込み

2.3.2.4 で得られた $\tilde{\boldsymbol{s}}$ 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 $C = \{\tilde{\boldsymbol{c}}_i \mid \tilde{\boldsymbol{c}}_i = \tilde{\boldsymbol{s}}(t_{p_i^O})\}_{i=0}^m$ と求める．図 2.13a のファジィ分割点の例を示す．

次に，$\tilde{\boldsymbol{c}}_i$ $(0 \leq i \leq m)$ の中心 $\boldsymbol{c}_i$，ファジネス $r_{\boldsymbol{c}_i}$ を基にして，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 2.13b 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．

[*10] 本章では $\alpha = 2$ とした．



(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 2.13: ファジィ分割点における刈り込みと再接続

ただし，$q_i^{P^O}{}_{u_0^i,u_1^i}$ $(0 \leq i < m)$ は $q_i^{P^O}$ 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，

$$u_0^i = \min(\{u \mid u \in [0,1], \|q_i^{P^O}(u) - c_i\| = \alpha r_{c_i}\} \cup \{1\})$$
$$u_1^i = \max(\{u \mid u \in [0,1], \|q_i^{P^O}(u) - c_{i+1}\| = \alpha r_{c_{i+1}}\} \cup \{0\})$$

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．

2.3.3.2　3 次 Bézier 曲線による再接続

2.3.3.1 で求めた 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 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 $\tilde{s}_{a_{h_{i-1}+1},b_{h_i}}$ $(0 \leq i \leq m')$ を等時間間隔でサンプリングして，ファジィ点列 $D_i = \{\tilde{d}_k^i\}_{k=0}^{N-1}$ を求める[*11]．ここで，$h_{-1} = -1$，$h_{m'} = m$ であり，

[*11] 本章では $N = 33$ とした．

$\forall h \in \{0, \ldots, m\}$ 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 2.13c 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 $\boldsymbol{b}_i : [0,1] \to \mathbb{E}^2$ $(0 \leq i \leq m')$ 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 $\boldsymbol{b}_j^i$ $(0 \leq j \leq 3)$ 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

$$E_i = \sum_{k=0}^{N-1} w_k^i \left\| \boldsymbol{d}_k^i - \boldsymbol{b}_i(t_k^i) \right\|^2 \tag{2.3}$$

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

$$\boldsymbol{b}_0^i = \boldsymbol{r}_{h_{i-1}}(u_1^{h_{i-1}}) \ (0 < i \leq m') \tag{2.4}$$
$$\boldsymbol{b}_3^i = \boldsymbol{r}_{h_i}(u_0^{h_i}) \ (0 \leq i < m') \tag{2.5}$$

端点における接線方向一致条件

$$\boldsymbol{b}_1^i = \boldsymbol{b}_0^i + \beta_i \boldsymbol{r}'_{h_{i-1}}(u_1^{h_{i-1}}) \ (\beta_i \in \mathbb{R},\ 0 < i \leq m') \tag{2.6}$$
$$\boldsymbol{b}_2^i = \boldsymbol{b}_3^i + \gamma_i \boldsymbol{r}'_{h_i}(u_0^{h_i}) \ (\gamma_i \in \mathbb{R},\ 0 \leq i < m') \tag{2.7}$$

ここで，$\boldsymbol{r}'_l(t)$ $(0 \leq l < m)$ 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，$\boldsymbol{b}_0^i$ $(0 < i \leq m'), \boldsymbol{b}_3^i$ $(0 \leq i < m')$ は式 (2.4)，(2.5) より決定され，残りの制御点 $\boldsymbol{b}_0^0$，$\boldsymbol{b}_1^i$ $(0 \leq i \leq m')$，$\boldsymbol{b}_2^i$ $(0 \leq i \leq m')$，$\boldsymbol{b}_3^{m'}$ を決定する未知パラメータは

$i = 0$ **のとき** $x_0^0,\ y_0^0,\ x_1^0,\ y_1^0,\ \gamma_0,$
$0 < i < m'$ **のとき** $\beta_i,\ \gamma_i,$
$i = m'$ **のとき** $\beta_{m'},\ x_2^{m'},\ y_2^{m'},\ x_3^{m'},\ y_3^{m'}$

[*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．

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

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

とおき式 (2.3) を $\boldsymbol{\xi}_i$ の関数とみなせば，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 (2.6)，(2.7) 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 2.13d に示すような制御点が決定され，図 2.13e 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 2.13f に示すように全体として G^1 連続な曲線を生成する[*13]．

2.4 自由曲線整形法の動作実験

提案手法が 2.1 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 16GB，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．

2.4.1 自由曲線整形法の幾何作図への適用例

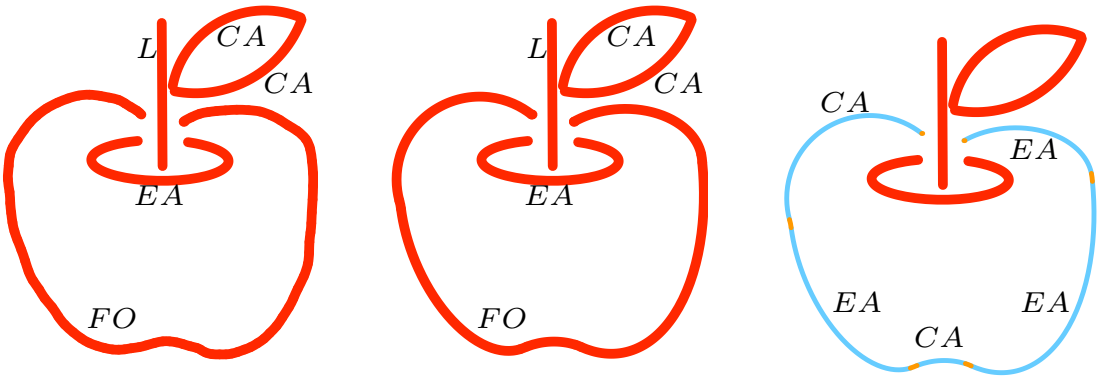
提案した自由曲線整形法を図 2.2 の幾何作図へ適用したことによる効果を図 2.14 で比較して示す．提案手法によって FO は図 2.14c のような構成要素からなる曲線として整形された．これにより，図 2.14b の FO では，図 2.14a に残っていた手書きストロークのうねりが解消されて，周囲の幾何プリミティブと調和することが分かる．

2.4.2 整形形状と処理時間

図 2.15，図 2.16 および図 2.17 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 2.3 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，整形処理時間は高々 1 秒程度となっており，インタラクティブなインターフェースに十分利用可能である．こ

[*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`

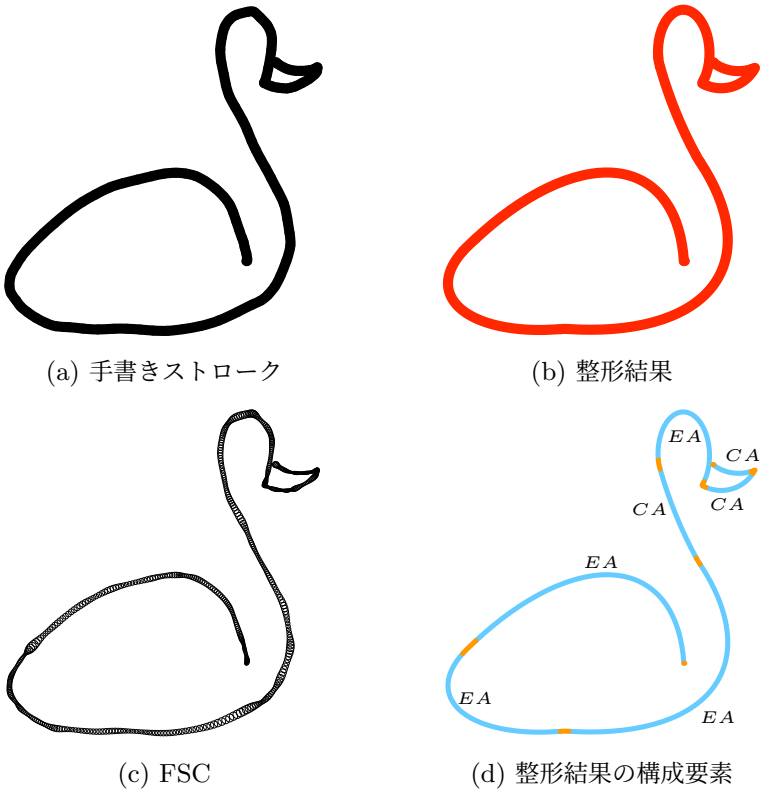


(a) FO への整形を行わない場合の
同定幾何プリミティブ

(b) FO への整形を行った場合の
同定幾何プリミティブ

(c) FO の整形結果の
構成要素

図 2.14: 図 2.2 の FO に対する自由曲線整形の効果



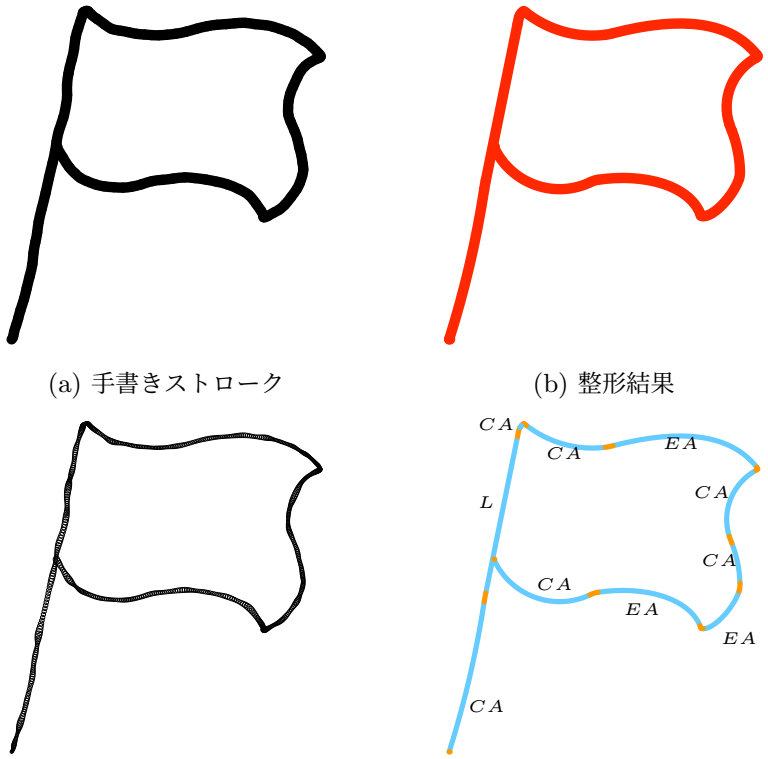
(a) 手書きストローク

(b) 整形結果

(c) FSC

(d) 整形結果の構成要素

図 2.15: 「スワン」の整形



(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 2.16: 「フラッグ」の整形

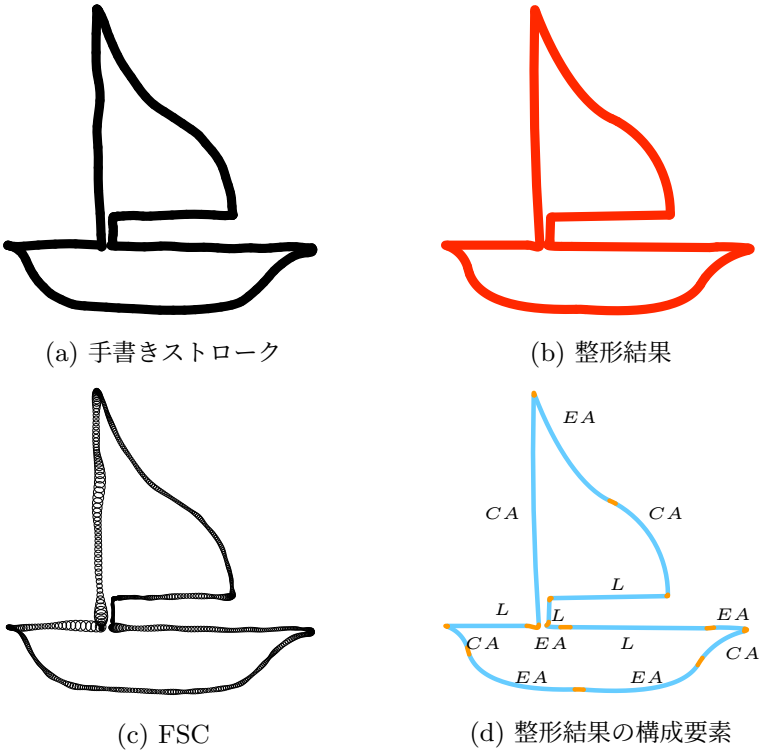
表 2.3: 「スワン」,「フラッグ」,「ヨット」の描画時間および整形処理時間

	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

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関わらず，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．

2.4.3 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 2.18 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 2.19, 図 2.20, 図 2.21 に示す．図 2.19 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プリミティブ数は減り，手書きストロークが大胆に単純化された整



(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 2.17: 「ヨット」の整形

形形状となっている．この結果は素早く雑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省略する効果を持つことを示す．一方で，図 2.21 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プリミティブ数は増え，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．

2.4.4　重ね書きによる整形形状の逐次的修正

手書き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，FSC 同定法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 [25] 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 2.22a のような既存の FSC に対して，ユーザが図 2.22b 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 2.22c 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 2.23 に示す．ここではまず，図

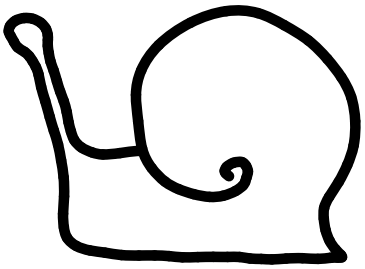


図 2.18: 手書きストローク「かたつむり」

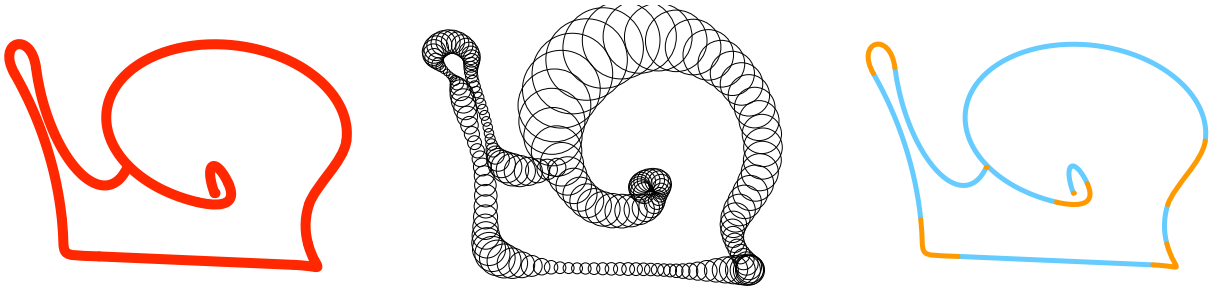


図 2.19: 「かたつむり」（描画速度 3 倍）の整形

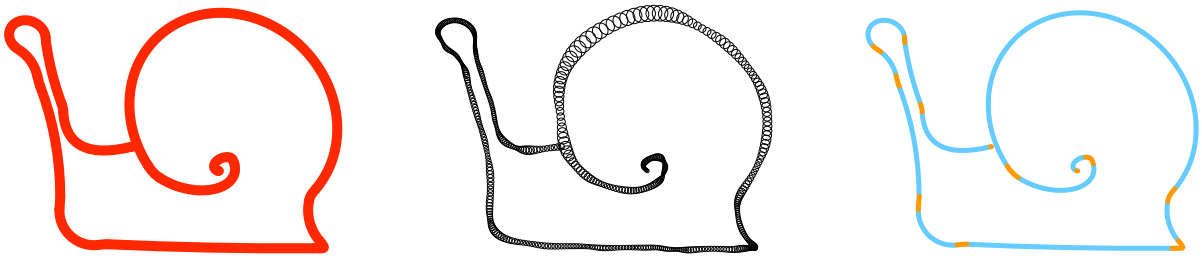


図 2.20: 「かたつむり」（描画速度 1 倍）の整形

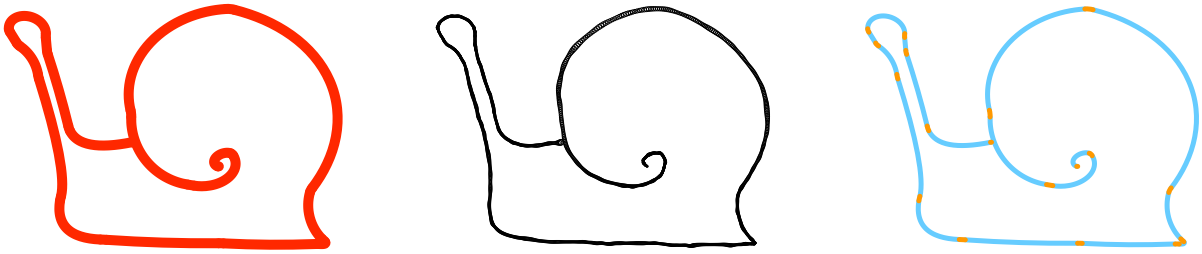
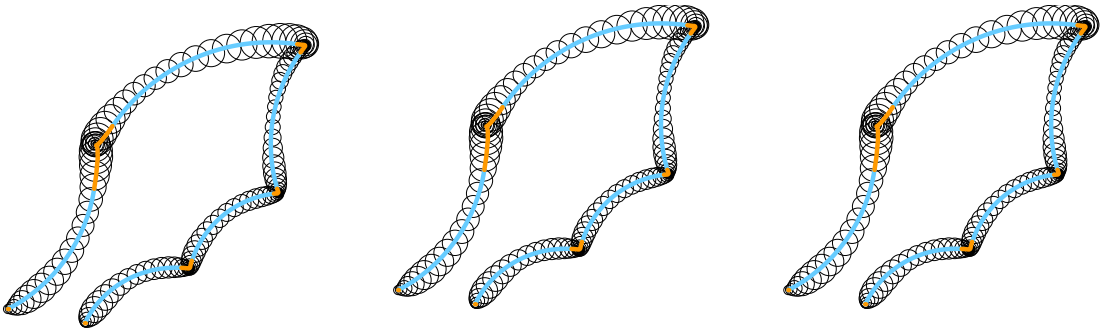
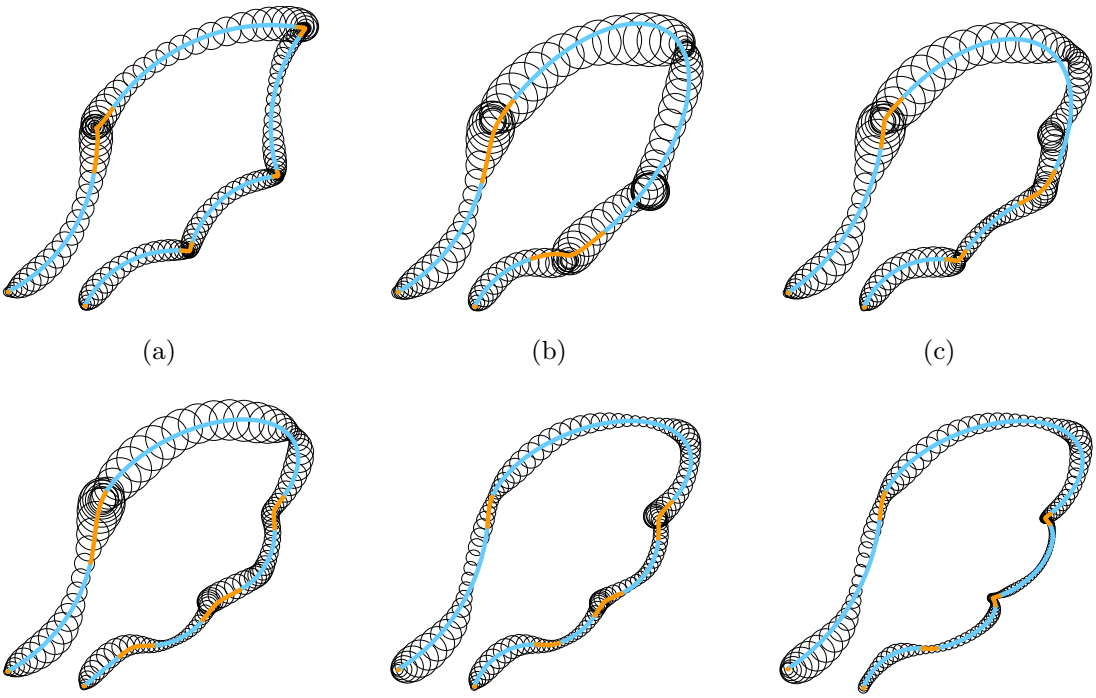


図 2.21: 「かたつむり」（描画速度 $\frac{1}{3}$ 倍）の整形



(a) 既存の FSC および整形結果　(b) 図 2.22a 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 2.22: 作図アプリケーションの動作例



(a)　(b)　(c)

(d)　(e)　(f)

図 2.23: 重ね書きによる整形結果の逐次的修正の例

2.23a に対して素早く雑に重ね書きすることで，図 2.23b のように FSC のファジネスを大きくし，整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 2.23c から図 2.23f 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．

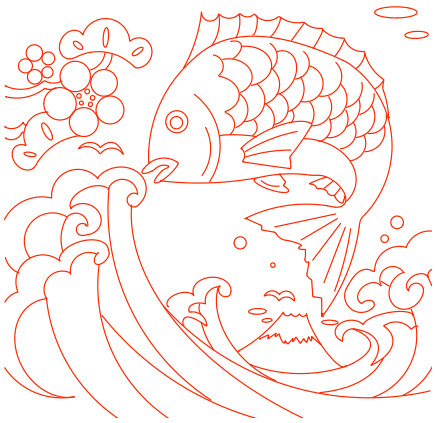
2.4.5　実際的な幾何作図における自由曲線整形法の適用例

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 2.4.4 の作図アプリケーションで重ね書きを繰り返しつつ，複数の幾何プリミティブから構成される幾何作図を完成した．入力された全ての手書きストロークを図 2.24a に，幾何作図の結果を図 2.24b 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 2.24c に示す．図 2.24c 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 2.24b ではこれらの FC, FO のうねりが解消され，周囲の幾何プリミティブ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．

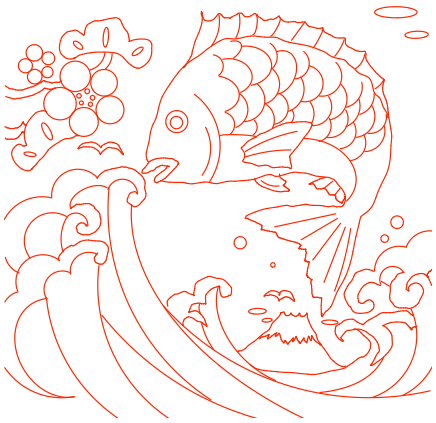
2.5　本章のまとめ

本章では，FSC 同定法の幾何プリミティブ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．

(a) 入力された全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プリミティブ



(c) FC，FO への整形を行わない場合の同定幾何プリミティブ

図 2.24: 実際的な幾何作図における自由曲線整形の効果

第 3 章

FSC 同定法に基づく $n/4$ 円弧および $n/4$ 楕円弧の同定

描画軌跡と描画動作のあいまいさをもとにして手書き曲線を 7 クラスの幾何プリミティブ (線分，円，円弧，楕円，楕円弧，閉自由曲線，開自由曲線) のいずれかに同定する手法として「ファジィスプライン曲線同定法（FSC 同定法)」が提案されている．また，FSC 同定法によって同定された幾何プリミティブを正方グリッドに合わせて整形する手法として「多重解像度ファジィグリッドスナッピング法 (MFGS)」が提案されており，この MFGS と FSC 同定法を組み合わせることによって，描画動作のみで 7 クラスの幾何プリミティブを配置でき様々な幾何作図を行うことができる CAD 用の手書きインターフェースが実現された．ここで，FSC 同定法が円弧または楕円弧を同定した場合，これらを正方グリッドに合わせて整形するためには，MFGS による整形に先立って，中心角や長径短径比といったパラメータを量子化して整形する必要があった．しかし，特に弧が浅い場合には，これらの形状パラメータ量子化による整形が適正に動作しがたくなり，MFGS による整形がうまく行われない場合が多発した．本論文では，初めに円弧および楕円弧の形状パラメータ量子化による整形は多くの場合において本質的に困難な問題となることを明らかにする．次にこの問題を回避するために，FSC 同定法の同定アルゴリズムを部分的に改変した「サブ曲線同定法」を提案し，これにより形状パラメータ量子化なしに MFGS による整形を施すことのできる「$n/4$ 円弧」および「$n/4$ 楕円弧」を新たに同定できるようにする．最後に，従来の 7 クラスの幾何プリミティブに加えて $n/4$ 円弧および $n/4$ 楕円弧をも配置できる新しい手書きインターフェースを実現し，評価実験によってこれが CAD インターフェースとして有効に機能することを示す．

3.1 本章の目的

ペンタブレットやタッチディスプレイの普及により，ユーザが手書き動作でデバイスを操作することが一般的となった．それに伴い手書きストロークを入力することによりインタラクティブに作図を行うシステムの開発が進められている．そこではまず，文献 [37] などのように作図部品の図形認識に基づく特定用途向けの手書き作図システムが提案された．次いで，文献 [18, 17, 32, 21] のような手書き曲線の整形技術の研究が進められた．これらは，手書き曲線を線分，円，円弧，楕円，自由曲線などの曲線セグメントの集合体として認識するもので，これによってより汎用的な手書き作図システムが実現した．しかし，これらの手書き作図システムは CAD 用の入力インターフェースとして開発されたものではないため，楕円弧の認識を実装しておらず，たとえば図 3.1 のような幾何作図を行うことはできない．これに対し，二次元 CAD の入力インタフェースとして汎用的に利用できる手書き幾何作図インターフェースの実現を目指し，手書き曲線を 7 クラスの幾何プリミティブ（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））のいずれかとして同定する手法「ファジィスプライン曲線同定法（FSC 同定法）」[22] が提案された．FSC 同定法では描画動作の素早さの程度を描画のあいまいさ (ファジネス) に関連付けるというコンセプトを提案することで楕円弧を含む 7 クラスの幾何プリミティブの同定を可能とした．さらに FSC 同定法で同定された幾何プリミティブの特徴点をそのファジネスに応じて適切な解像度の正方グリッドにスナッピングして整形する手法「多重解像度ファジィグリッドスナッピング法（MFGS）」[31, 38] が提案され，正方グリッドを利用した様々な幾何作図を描画動作のみで完了できる手書き CAD インターフェース「SKIT」[1] が実現された．

ところで SKIT において FSC 同定法が円弧または楕円弧を同定した場合，これらを図 3.1 に見るように正方グリッドに合わせて整形するためには，MFGS による整形に先立って中心角や長径短径比といったパラメータを量子化して整形する必要がある．このため文献 [39] では FSC 同定法の同定結果に対する後処理として，形状パラメータの量子化手法が提案された．しかし特に手書きストロークの弧が浅い場合には形状パラメータ量子化による整形が原理的に適正に機能しがたくなる問題が多発した．このため，比較的弧が深いときにはユーザがその意図通りに円弧および楕円弧を正方グリッドに合わせて作図できたが，弧が浅くなるにつれ円弧および楕円弧を意図通りに作図することは極端に困難となり，SKIT は手書き CAD インターフェースとしては不十分なものとなっていた．一方，FSC 同定法のコンセプトに基づいて別途提案された文献 [29] の FFDS でも描画のあいまいさを利用した楕円弧認識が行われ，文献 [40] で形状パラメータの調整によって楕円弧を等角投影軸に当てはめる整形が行われているが，弧が浅い場合の整形結果については示されていない．

本章では，初めに円弧および楕円弧の形状パラメータ量子化による整形は多くの場合において本質的に困難な問題となることを明らかにする．次にこの問題を回避するために，形状パラメータ量子化による整形を必要としない新たなクラスの幾何プリミティブとして「$n/4$ 円弧」および「$n/4$ 楕円弧」を導入する．そして，FSC 同定法の同定アルゴリズムを部分的に改変することにより，

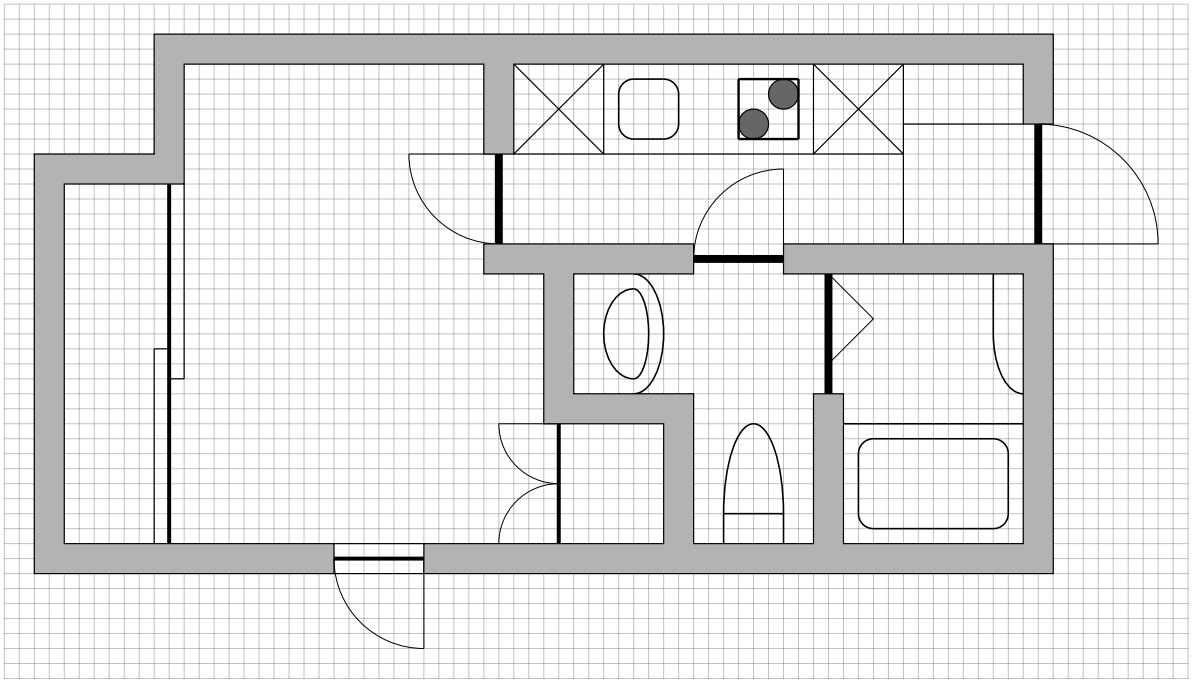


図 3.1: 幾何作図の例

$n/4$ 円弧および $n/4$ 楕円弧を同定する手法，すなわち「サブ曲線同定法」を新たに提案する．最後に，従来の形状パラメータ量子化による整形を廃止した上で新たにサブ曲線同定法を導入することで，7 クラスの幾何プリミティブに加えて $n/4$ 円弧および $n/4$ 楕円弧をも配置できる新たな SKIT を実現する．また，評価実験によってこれが手書き CAD インターフェースとして有効に機能することを示す．

3.2 手書き CAD インターフェースにおける円弧整形および楕円弧整形の困難性

FSC 同定法を基盤とする手書き CAD インターフェースについて概説したのち，手書きインターフェースにおいては浅い円弧／楕円弧 (それが部分として含まれる円／楕円全体に対してその占める割合が小さい円弧／楕円弧) を正方グリッドに合わせて整形することが本質的に困難となることを明らかにする．

3.2.1 FSC 同定法による手書き CAD インターフェースの概要

既に，手書きストロークを繰り返し描画し続けるだけで複雑な幾何作図を完了できるインラクティブな手書き CAD インターフェースとして SKIT が実現されている．SKIT は一本ごとのストローク入力に逐次反応し，図 3.2 に示す流れで，ファジィスプライン補間，FSC 同定法による曲線

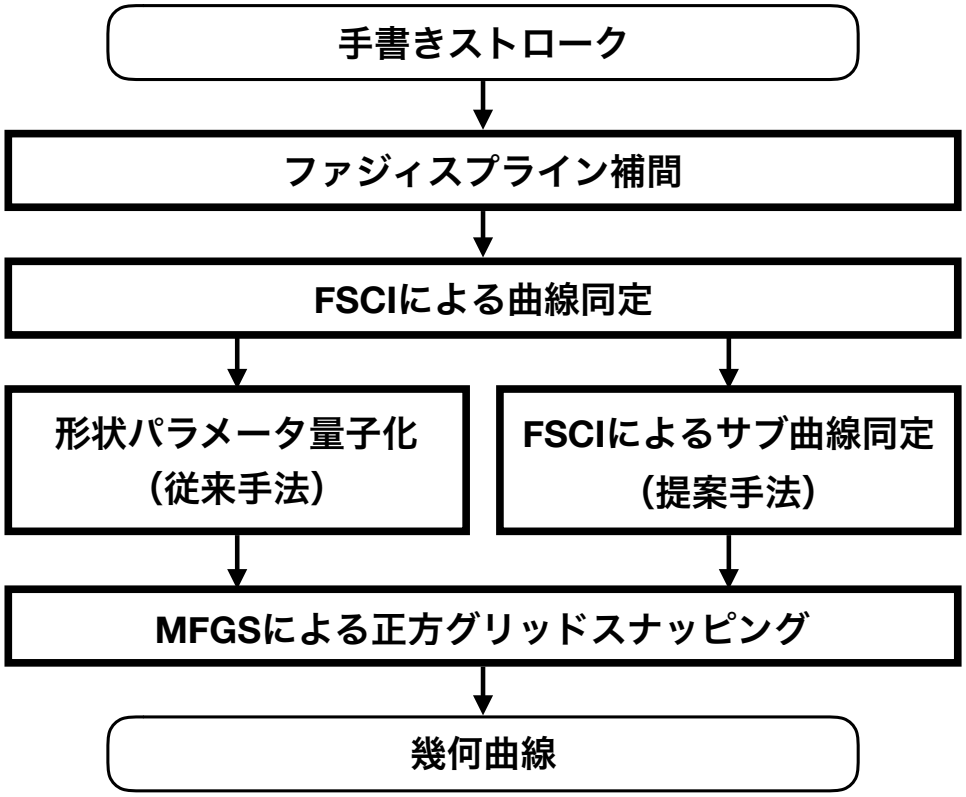


図 3.2: 手書き CAD インターフェースにおける曲線同定の流れ

同定，形状パラメータ量子化，MFGS による正方グリッドスナッピングをリアルタイムに実行する．以下にこの処理を概説する．

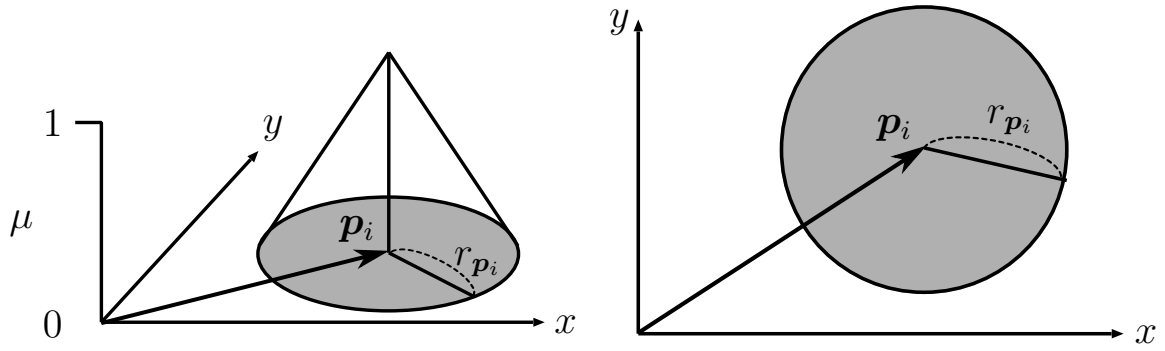
(1) **ファジィスプライン補間** 時系列点列 $\boldsymbol{p}_i$ $(i = 0, 1, \ldots, n_p)$ として入力された手書きストロークを文献 [35] の手法に基づいてファジィ化し，ファジィ時系列点列 $\tilde{\boldsymbol{p}}_i$ $(i = 0, 1, \ldots, n_p)$ を生成する．ここで，各ファジィ点 $\tilde{\boldsymbol{p}}_i$ は位置のあいまいな点のモデルである「円錐型ファジィ点」で表され，具体的には図 3.3 の円錐型メンバシップ関数

$$\mu_{\tilde{\boldsymbol{p}}_i}(\boldsymbol{v}) = \left(1 - \frac{\|\boldsymbol{v} - \boldsymbol{p}_i\|}{r_{\boldsymbol{p}_i}}\right) \vee 0 \tag{3.1}$$

で特徴づけられるファジィ集合で定義される．ただし $\boldsymbol{v}$ は変数位置ベクトル，$\boldsymbol{p}_i$ はファジィ点の頂点の位置ベクトル，$r_{\boldsymbol{p}_i}$ は位置のあいまいさの程度（ファジネス），$\vee$ は max 演算を表わす．このとき，$r_{\boldsymbol{p}_i}$ を各点における描画の加速度の大きさ a_i および速度の大きさ v_i に基づいて，

$$r_{\boldsymbol{p}_i} = C_a a_i + C_v v_i \tag{3.2}$$

と設定することにより，素早い描画部分では大きなファジネスを生成し，逆にゆっくりとした丁寧な描画部分では小さなファジネスを生成する．なお，本章では C_a =



(a) 射影図　　(b) 上面図

図 3.3: 円錐型ファジィ点 $\tilde{\boldsymbol{p}}_i$ のメンバシップ関数

$7.7[\times 10^{-3}\mathrm{s}^2], C_v = 8.6[\times 10^{-3}\mathrm{s}]$ とした．次に，ファジィ時系列点列 $\tilde{\boldsymbol{p}}_i$ を文献 [24] の手法によりファジィスプライン補間しファジィスプライン曲線 (Fuzzy Spline Curve, FSC) を生成する．FSC は手書きストロークの描画軌跡とともにその位置のあいまいさを表現するファジィな曲線で

$$\tilde{\boldsymbol{s}}(t) = \sum_{i=0}^{m} N_i(t)\tilde{\boldsymbol{d}}_i \tag{3.3}$$

と表現される．ただし，t は時刻，$N_i(t)$ は 3 次の B-スプライン基底関数，$\tilde{\boldsymbol{d}}_i$ は円錐型ファジィ点として得られるファジィ制御点である．ここで，$\sum_{i=0}^{m} N_i(t) = 1$ であることから，$\tilde{\boldsymbol{s}}(t)$ は $\tilde{\boldsymbol{d}}_i$ $(i = 0, 1, \ldots, m)$ の重心結合となっていることに注意する．文献 [24] が示すとおり，一般に，円錐型ファジィ点 $\tilde{\boldsymbol{d}}_i$ をその頂点位置ベクトル $\boldsymbol{d}_i$ とファジネス $r_{\boldsymbol{d}_i}$ を用いて $\langle \boldsymbol{d}_i, r_{\boldsymbol{d}_i} \rangle$ と表記することにすれば，その重心結合 $\sum_{i=0}^{m} k_i\tilde{\boldsymbol{d}}_i$ $(\sum_{i=0}^{m} k_i = 1)$ は円錐型ファジィ点となり $\sum_{i=0}^{m} k_i\tilde{\boldsymbol{d}}_i = \langle \sum_{i=0}^{m} k_i\boldsymbol{d}_i, \sum_{i=0}^{m} |k_i| r_{\boldsymbol{d}_i} \rangle$ と求められる．そのため，式 (3.3) の $\tilde{\boldsymbol{s}}(t)$ は t の変化とともに図 3.5 に見るように円錐型ファジィ点の移動軌跡を描く．なお，ファジィ時系列点列のファジネスを式 (3.2) で設定した結果，素早い描画部分ではファジネスが大きくなり，逆にゆっくりとした丁寧な描画部分ではファジネスが小さくなる FSC が生成されることになる．

(2)FSC **同定法による曲線同定** FSC を文献 [22] の FSC 同定法で 7 クラスの幾何プリミティブのいずれかとして同定し，曲線クラス，形状パラメータおよびファジィ特徴点を出力する (図 3.4 参照).

(2-1) **仮説ファジィモデルの構成** FSC が線形，円形，楕円形であると仮定したときの仮説モデルを，文献 [22, 41] の手法に従って，それぞれ線形仮説ファジィモデル $\tilde{\boldsymbol{r}}^L(t)$，円

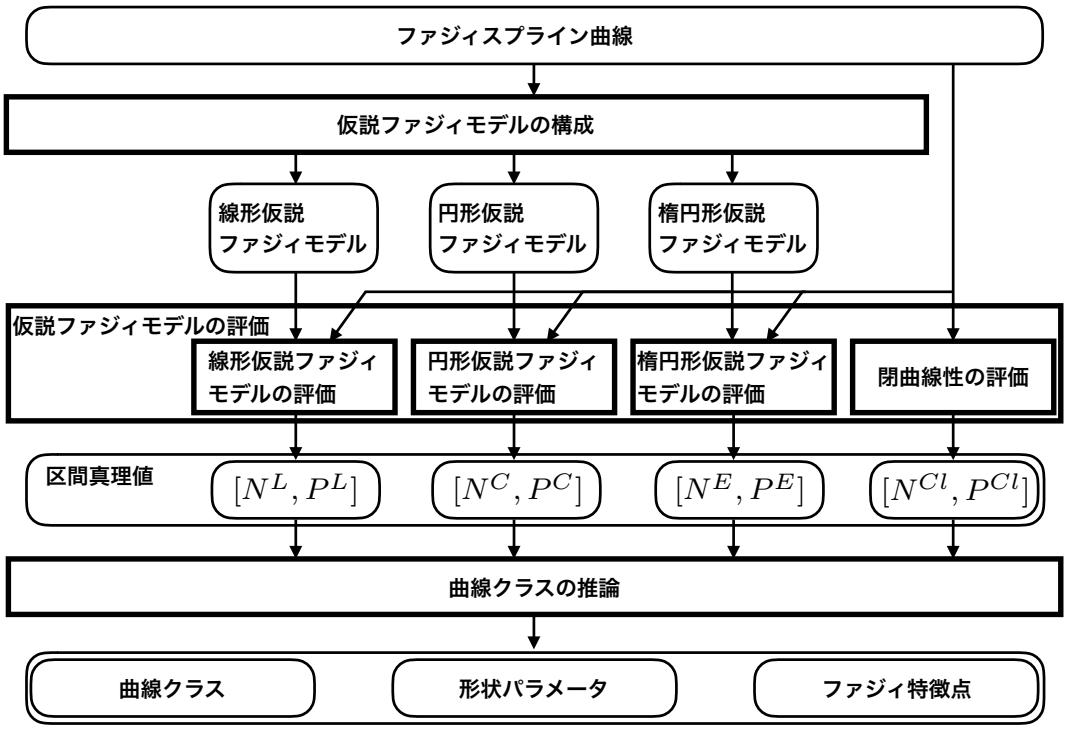


図 3.4: FSC 同定法による曲線同定の流れ

形仮説モデル $\tilde{\boldsymbol{r}}^C(t)$，楕円形仮説ファジィモデル $\tilde{\boldsymbol{r}}^E(t)$ として以下のように構成する．

$$\tilde{\boldsymbol{r}}^L(t) = B_0^1(t)\tilde{\boldsymbol{b}}_0^L + B_1^1(t)\tilde{\boldsymbol{b}}_1^L \tag{3.4}$$

$$\tilde{\boldsymbol{r}}^C(t) = \frac{P_0(t)\tilde{\boldsymbol{b}}_0^C + P_1(t)(1+w^C)\tilde{\boldsymbol{f}}^C + P_2(t)\tilde{\boldsymbol{b}}_2^C}{P_0(t) + P_1(t)(1+w^C) + P_2(t)} \tag{3.5}$$

$$\tilde{\boldsymbol{r}}^E(t) = \frac{P_0(t)\tilde{\boldsymbol{b}}_0^E + P_1(t)(1+w^E)\tilde{\boldsymbol{f}}^E + P_2(t)\tilde{\boldsymbol{b}}_2^E}{P_0(t) + P_1(t)(1+w^E) + P_2(t)} \tag{3.6}$$

ただし，$P_0(t) = B_0^2(t) - \frac{1}{2}B_1^2(t)$, $P_1(t) = B_1^2(t)$, $P_2(t) = B_2^2(t) - \frac{1}{2}B_1^2(t)$ であり，$B_i^n(t)$ は n 次 Bernstein 多項式を表す．ここで，$\tilde{\boldsymbol{b}}_0^L$, $\tilde{\boldsymbol{b}}_1^L$, $\tilde{\boldsymbol{b}}_0^C$, $\tilde{\boldsymbol{f}}^C$,$\tilde{\boldsymbol{b}}_2^C$, $\tilde{\boldsymbol{b}}_0^E$, $\tilde{\boldsymbol{f}}^E$, $\tilde{\boldsymbol{b}}_2^E$ は FSC 上から選択されるファジィ代表点であり，式 (3.4) は 1 次 Bézier 曲線，式 (3.5) および式 (3.6) は 2 次有理 Bézier 曲線で表され，いずれもファジィ代表点の重心結合となっている．文献 [22, 41] の手法で，これらの代表点を FSC の概形を代表するように適切に選択し，さらにパラメータ w^C および w^E を適切に設定することで，それぞれの仮説ファジィモデルは図 3.5 に見るようにそれぞれのモデルの制約条件のもとでできるだけ FSC に近い仮説を構成する．

(2-2) **仮説ファジィモデルの評価** 三つの仮説ファジィモデル $\tilde{\boldsymbol{r}}^L(t)$，$\tilde{\boldsymbol{r}}^C(t)$，および $\tilde{\boldsymbol{r}}^E(t)$ がもとの FSC $\tilde{\boldsymbol{s}}(t)$ と合致する度合いをそれぞれ区間真理値 $[N^L, P^L]$, $[N^C, P^C]$, $[N^E, P^E]$ で評価する．さらに FSC $\tilde{\boldsymbol{s}}(t)$ の閉曲線性を FSC のファジィ始点とファジィ終点の合致の度合いとして区間真理値 $[N^{Cl}, P^{Cl}]$ で評価する．これらの N および P

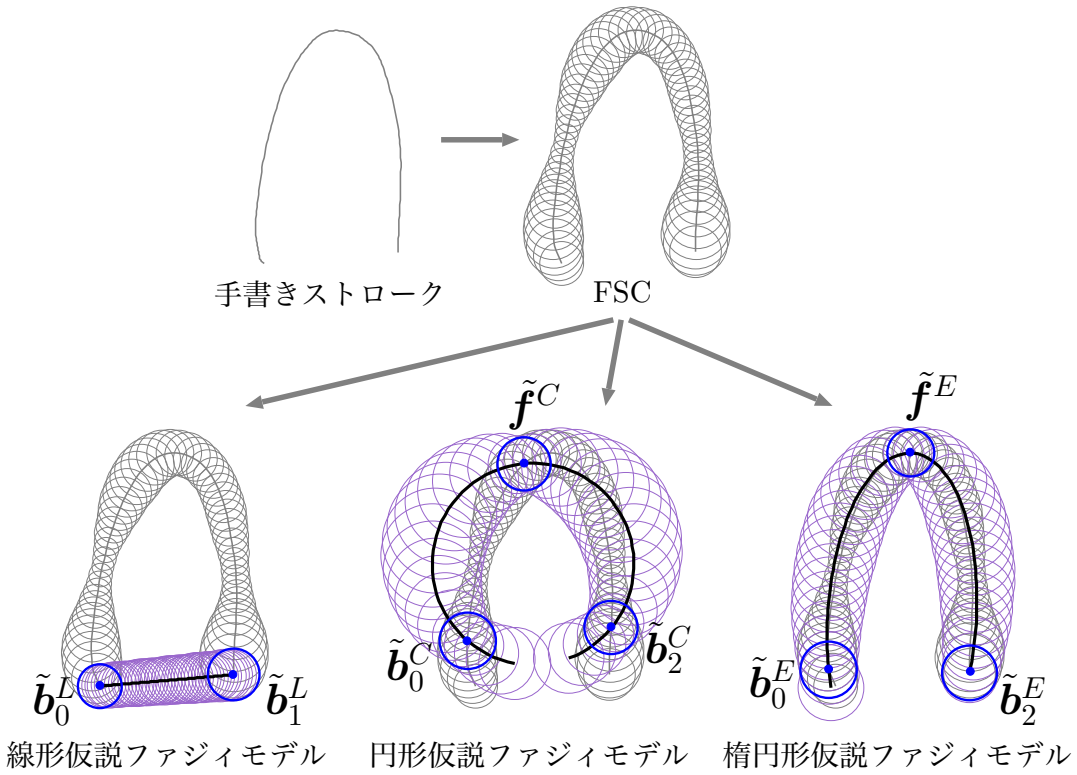


図 3.5: 仮説ファジィモデル生成の流れ

表 3.1: 曲線同定のファジィ推論規則

$\mu(L)$	$= (P^L \text{ is } T)$		
$\mu(C)$	$= (N^L \text{ is } F) \wedge (P^C \text{ is } T)$		$\wedge (P^{Cl} \text{ is } T)$
$\mu(CA)$	$= (N^L \text{ is } F) \wedge (P^C \text{ is } T)$		$\wedge (N^{Cl} \text{ is } F)$
$\mu(E)$	$= (N^L \text{ is } F) \wedge (N^C \text{ is } F)$	$\wedge (P^E \text{ is } T)$	$\wedge (P^{Cl} \text{ is } T)$
$\mu(EA)$	$= (N^L \text{ is } F) \wedge (N^C \text{ is } F)$	$\wedge (P^E \text{ is } T)$	$\wedge (N^{Cl} \text{ is } F)$
$\mu(FC)$	$= (N^L \text{ is } F) \wedge (N^C \text{ is } F)$	$\wedge (N^E \text{ is } F)$	$\wedge (P^{Cl} \text{ is } T)$
$\mu(FO)$	$= (N^L \text{ is } F) \wedge (N^C \text{ is } F)$	$\wedge (N^E \text{ is } F)$	$\wedge (N^{Cl} \text{ is } F)$

はそれぞれファジィ理論における必然性値および可能性値 [36] であり，具体的には文献 [41] の手法で求められる．

(2-3) **曲線クラスの推論** 文献 [41] の手法に従って表 3.1 のファジィ推論規則で 7 クラスの曲線クラスのグレード値 $\mu(L)$, $\mu(C)$, $\mu(CA)$, $\mu(E)$, $\mu(EA)$, $\mu(FC)$, $\mu(FO)$ を算出する．ただし T と F はそれぞれ「真」と「偽」を表す言語的真理値 [42]，$\wedge$ は論理積を表し，それぞれのグレード値は文献 [41, 43] により区間 $[0, 1]$ 内の実数値として計算される．このときもっとも高いグレード値を得た曲線クラスを曲線クラスの同定結果と

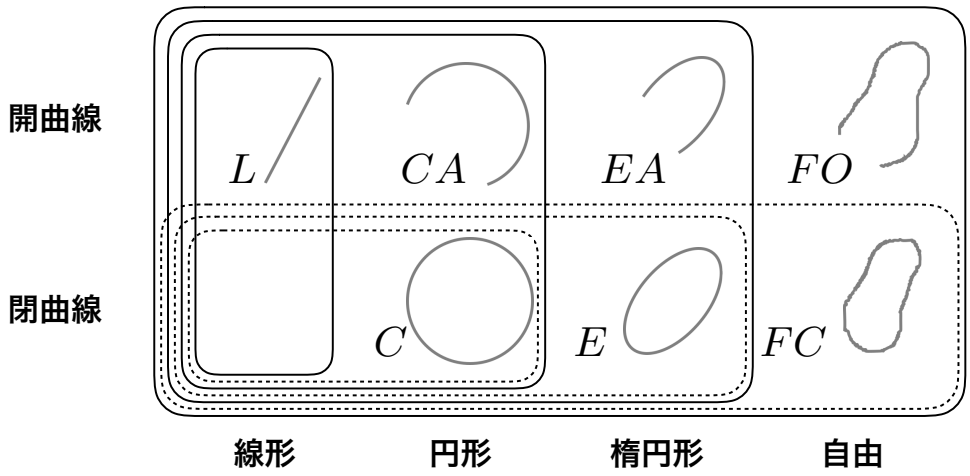


図 3.6: 曲線クラス間の自由度に関する包含関係

して出力する．また，同定された曲線クラスに対応した仮説ファジィモデルから同定曲線の形状パラメータを算出し出力する．さらに，正方グリッドスナッピングのために，同定された曲線クラスに応じて，文献 [38, 44] に基づいて仮説ファジィモデルから，2 点ないしは 3 点のファジィ特徴点を抽出し出力する．

ここで 7 クラスの曲線クラスの間には図 3.6 の自由度に関する包含関係があることに注意する．たとえば EA より自由度の低い CA は EA にその特殊な場合として包含される．また，閉曲線である E は開曲線である EA にその特殊な場合として包含される．したがって，描画形状のみからこれらを判別することは原理的に困難である．表 3.1 のファジィ推論規則は描画形状だけでなく FSC のファジネスをも勘案することでこの困難を解決している．こ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かぎりもっとも自由度の低い曲線クラスを推論するものとなっている．ユーザはこの推論規則を利用して 7 クラスの曲線クラスを意図的に区別して FSC 同定法に同定させることができる．すなわち，ユーザは素早い描画でファジネスの大きな FSC を生成させることで自由度の低い曲線クラスを同定させることができる．逆にゆっくりとした丁寧な描画でファジネスの小さな FSC を生成させることで自由度の高い曲線クラスを同定させることができる．

(3) **形状パラメータ量子化** 同定曲線クラスが CA の場合，開始角および終了角を切りのよい値に量子化する．また，同定曲線クラスが EA の場合には，開始角，終了角に加え，長径短径比の量子化も行う．具体的には文献 [39] の手法により，ファジネスによる可能性の広がりが許すかぎり粗い量子化を行うファジィ推論にしたがった量子化を行う．また，量子化結果に応じて (2-3) で得られたファジィ特徴点を再抽出する．

(4)MFGS **による正方グリッドスナッピング** ファジィ特徴点をそれぞれ文献 [31] の MFGS で正方

グリッドにスナッピングし，その結果に応じて曲線全体を相似変換（ファジィ特徴点が 2 点の場合）またはアフィン変換（ファジィ特徴点が 3 点の場合）して正方グリッドに合わせて整形された幾何プリミティブを得る．ここで，MFGS は解像度の異なる正方グリッドを複数有する．MFGS は特徴点のファジネスによる可能性の広がりが許すかぎりできるだけ低い解像度の正方グリッドを選択するファジィ推論を行った上でスナッピング先を決定する．

3.2.2 円弧整形および楕円弧整形の困難性

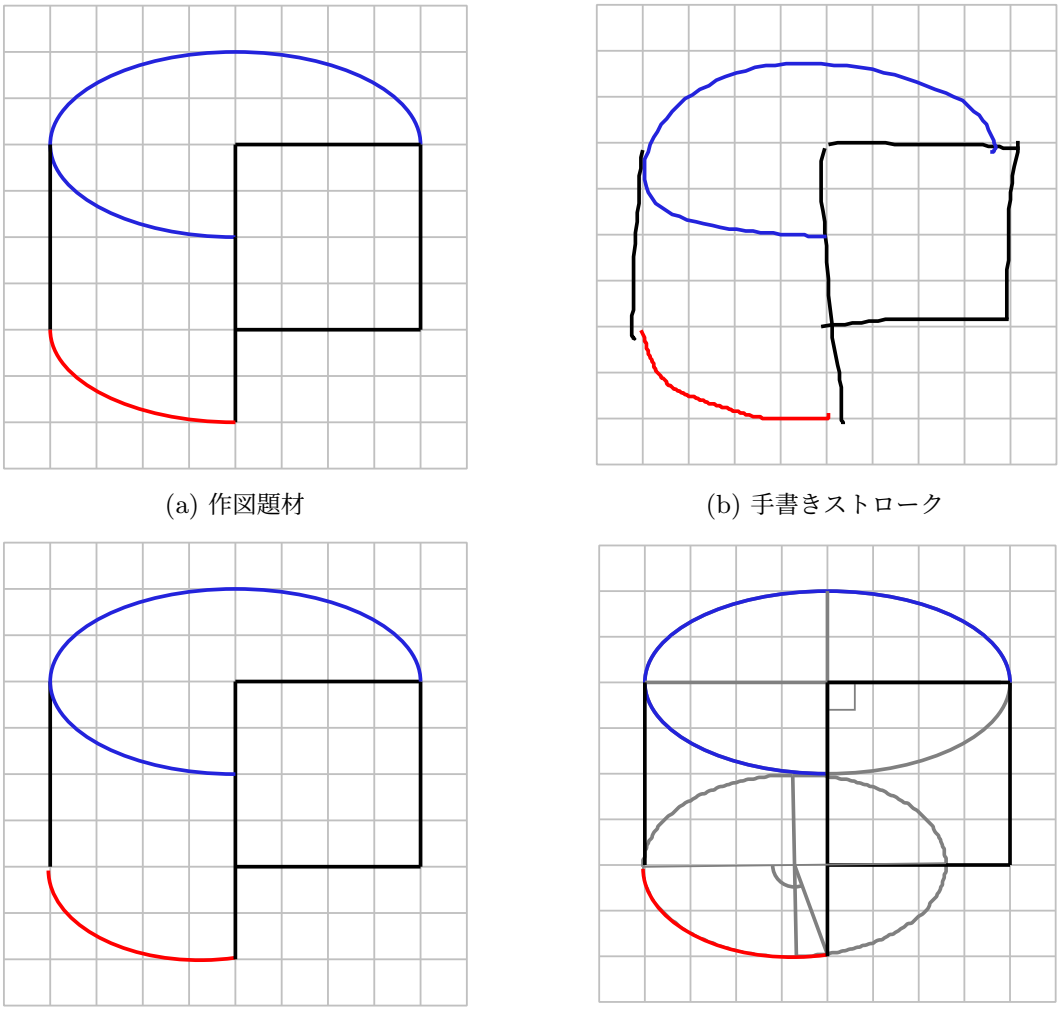
一般的な二次元 CAD は正方グリッドを有しており，このような CAD では多様な形状の円弧や楕円弧の作図だけでなく，図 3.1 に見るような正方グリッドに合わせて整形された円弧および楕円弧も作図できる必要がある．したがって，汎用的手書き CAD インターフェースでは多様な形状の円弧や楕円弧を自由に同定させることができると同時に，正方グリッドに合わせて整形された円弧および楕円弧を所望するときにはこれらを意図的に同定させることもできる必要がある．

しかし，FSC 同定法を基盤とする SKIT ではこのような整形が困難となる場合がある．たとえば図 3.7(a) の二次元作図題材を SKIT で作図しようと，図 3.7(b) に示す手書きストロークを描画したとき図 3.7(c) の作図結果が得られた．一見これで作図が完了したように見えるが，2 つの楕円弧の整形状態を詳しく見るために，図 3.7(d) にこれらの楕円弧を含む楕円全体を表示してみると，赤の楕円弧については長径短径が正方グリッドに重なっておらず中心角も $\frac{\pi}{2}$ でないことがわかる．つまりこの場合には，形状パラメータ量子化が赤の楕円弧を正方グリッドに合った 1/4 楕円弧に整形できなかったことを意味する．

ここで，図 3.8a のほぼ 1/4 楕円弧と思われる四つの楕円弧を考えたとき，それらを楕円全体とともに表示すると図 3.8b となり，それらの開始角と終了角は大きくばらつく．このことは FSC 同定法に楕円弧を同定させた後にその形状パラメータを量子化してもユーザが意図する所望の 1/4 楕円弧が整形結果として得られにくいことを示唆する．実際 SKIT では，孤の浅い楕円弧を同定させようとした場合に手書きストロークのわずかな変化が形状パラメータに大きな変化を招く．そのため，特に孤の浅い楕円弧の場合には形状パラメータ量子化で所望の整形結果を得ることが極端に困難となる．これは描画部分のみからそれを含む幾何プリミティブ全体を同定しようとする手書き幾何プリミティブ認識が直面する本質的な問題である．

3.3 $n/4$ 円弧同定および $n/4$ 楕円弧同定のためのサブ曲線同定法の提案

ユーザが正方グリッドに合わせて整形された円弧および楕円弧の入力を意図して描画をしたときに FSC 同定法がその意図をより正確に同定するよう改善する．そのために，従来の形状パラメータ量子化を必要としない曲線クラスとして新たなサブ曲線クラス「$n/4$ 円弧」および「$n/4$ 楕円弧」を導入した上で，これらを同定する手法，すなわち「サブ曲線同定法」を提案する．



(a) 作図題材　(b) 手書きストローク

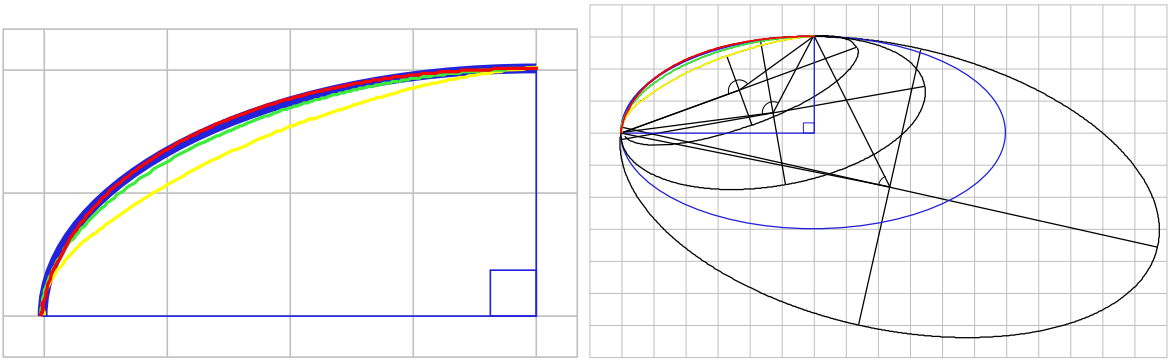
(c) 同定結果　(d) 楕円弧を含む楕円の表示

図 3.7: 形状パラメータ量子化による楕円弧同定の困難性

3.3.1 円弧のサブ曲線クラスおよび楕円弧のサブ曲線クラスの定義

FSC 同定法が同定する曲線クラスの一つである「円弧（CA)」について，その特殊な場合として三つのサブ曲線クラス，すなわち「1/4 円弧 （$\frac{1}{4}CA$)」,「2/4 円弧 （$\frac{2}{4}CA$)」,「3/4 円弧 （$\frac{3}{4}CA$)」を定義し，これらを総称して「$n/4$ 円弧 （$\frac{n}{4}CA$)」と呼ぶ．同様に「楕円弧 （EA)」についても，三つのサブ曲線クラス，すなわち，「1/4 楕円弧 （$\frac{1}{4}EA$)」,「2/4 楕円弧（$\frac{2}{4}EA$)」,「3/4 楕円弧（$\frac{3}{4}EA$)」を定義し，これらを総称して「$n/4$ 楕円弧（$\frac{n}{4}EA$)」と呼ぶ．具体的には以下のとおり定義する．

$n/4$ **円弧** （$\frac{n}{4}CA$)： 中心角が $\frac{n}{2}\pi$ の円弧
$n/4$ **楕円弧** （$\frac{n}{4}EA$)： 始点および終点が楕円の長径上または短径上に位置しかつ中心角が $\frac{n}{2}\pi$ の



(a) 四つの楕円弧　　(b) 楕円弧を含む楕円の表示

図 3.8: 1/4 楕円弧に近い楕円弧の形状パラメータのばらつき

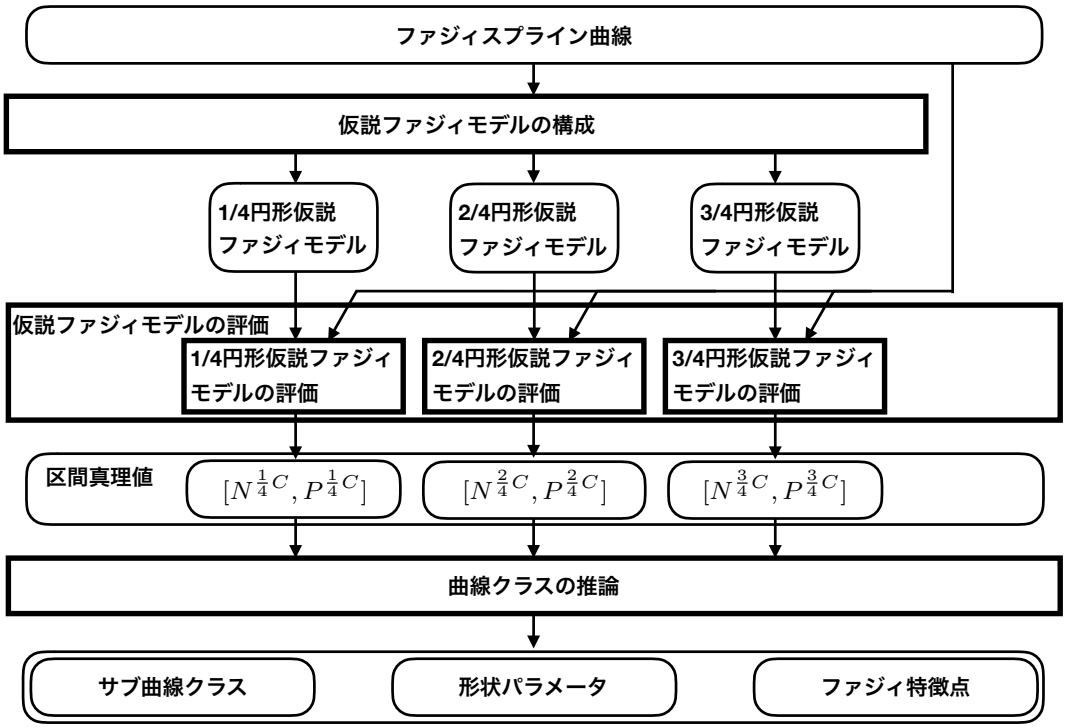


図 3.9: FSC 同定法による円弧サブ曲線同定の流れ

楕円弧，およびこれをせん断変形して得られる楕円弧

3.3.2　FSC 同定法におけるサブ曲線同定の位置づけと機能

ここで提案するサブ曲線同定は従来の FSC 同定法による曲線同定の後処理として実装するもので，従来の形状パラメータ量子化を置換する (図 3.2 参照)．サブ曲線同定は，前段の曲線同定にお

ける曲線クラスの同定結果に応じて，それをさらに以下のとおりサブ曲線クラスに細分化して同定する．

CA の場合： 円弧サブ曲線同定を行い，$\frac{1}{4}CA$，$\frac{2}{4}CA$，$\frac{3}{4}CA$，CA のいずれかのサブ曲線クラスの幾何プリミティブとして再同定し出力する．

EA の場合： 楕円弧サブ曲線同定を行い，$\frac{1}{4}EA$，$\frac{2}{4}EA$，$\frac{3}{4}EA$，EA のいずれかのサブ曲線クラスの幾何プリミティブとして再同定し出力する．

L, C, E, FC, FO の場合： サブ曲線同定による細分化を行わず，もとの同定結果をそのまま出力する．

ここで，円弧サブ曲線同定が出力するサブ曲線クラスが $\frac{n}{4}CA$ のみならず CA ともなりうる点に注意する．サブ曲線クラスとしての CA は（特別な円弧である $\frac{n}{4}CA$ ではない）一般の CA を意味する．楕円弧サブ曲線同定も同様である．サブ曲線同定の具体的な処理を 3.3.3 および 3.3.4 に示す．

3.3.3 円弧サブ曲線同定

CA が同定された場合に，その特殊な場合である $\frac{n}{4}CA$ であるかどうかを評価し，$\frac{1}{4}CA$, $\frac{2}{4}CA$, $\frac{3}{4}CA$, CA のいずれかに細分化して再同定する．円弧サブ曲線同定の流れは図 3.9 に示すように従来の FSC 同定法の曲線同定の流れを踏襲したものとなる．以下に具体的な処理を示す．

(1) **仮説ファジィモデルの構成** 以下の三つの $n/4$ 円形仮説ファジィモデルを式 (3.5) と同じく 2 次有理 Bézier 曲線形式で構成する (図 3.10 参照)．

$$\tilde{\boldsymbol{r}}^{\frac{n}{4}C}(t)=\frac{P_0(t)\tilde{\boldsymbol{b}}_0^{\frac{n}{4}C}+P_1(t)(1+w^{\frac{n}{4}C})\tilde{\boldsymbol{f}}^{\frac{n}{4}C}+P_2(t)\tilde{\boldsymbol{b}}_2^{\frac{n}{4}C}}{P_0(t)+P_1(t)(1+w^{\frac{n}{4}C})+P_2(t)} \qquad (n=1,2,3) \tag{3.7}$$

ここで，2 次有理 Bézier 曲線の性質に注意して $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}C}$,$\tilde{\boldsymbol{b}}_2^{\frac{n}{4}C}$, $w^{\frac{n}{4}C}$,$\tilde{\boldsymbol{f}}^{\frac{n}{4}C}$ を以下の通り定めて，$\tilde{\boldsymbol{r}}^{\frac{n}{4}C}(t)$ の稜線が $n/4$ 円弧となるよう制約する．

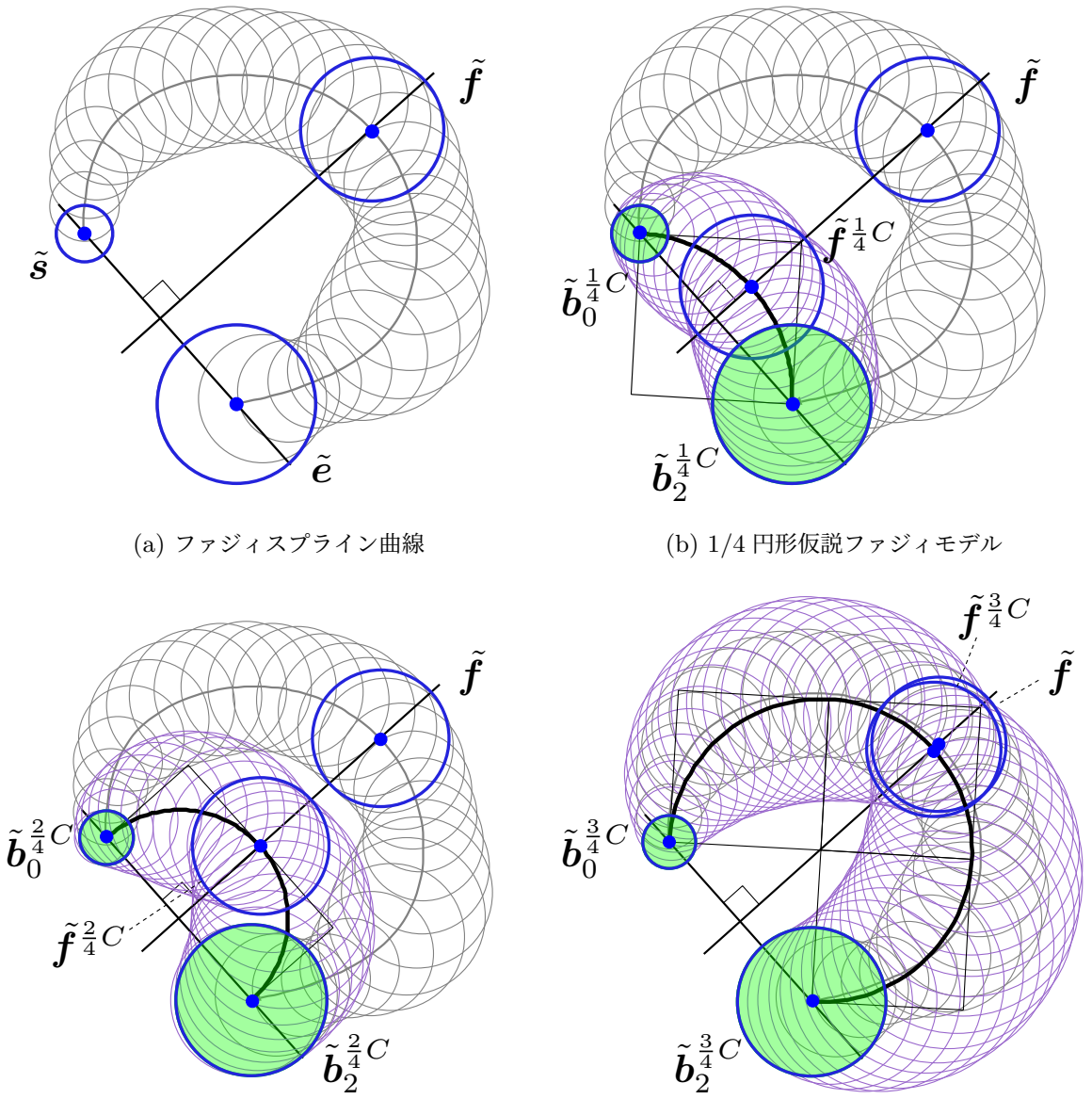
(1-1) **ファジィ代表点 $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}C}$ と $\tilde{\boldsymbol{b}}_2^{\frac{n}{4}C}$ の選択** FSC $\tilde{\boldsymbol{s}}(t)$ のファジィ始点 $\tilde{\boldsymbol{s}}$ とファジィ終点 $\tilde{\boldsymbol{e}}$ を求めて，これらをファジィ代表点として選択する．すなわち，$\tilde{\boldsymbol{b}}_0^{\frac{n}{4}C}=\tilde{\boldsymbol{s}}$ $(n=1,2,3)$ および $\tilde{\boldsymbol{b}}_2^{\frac{n}{4}C}=\tilde{\boldsymbol{e}}$ $(n=1,2,3)$ とする．

(1-2) **$w^{\frac{n}{4}C}$ の設定** 中心角が $\frac{n}{2}\pi$ となるように $w^{\frac{n}{4}C}=\cos\frac{n}{4}\pi$ $(n=1,2,3)$ と設定する．

(1-3) **ファジィ代表点 $\tilde{\boldsymbol{f}}^{\frac{n}{4}C}$ の設定**　以下の手順で $\tilde{\boldsymbol{f}}^{\frac{n}{4}C}$ $(n=1,2,3)$ を定める．まず線分 $\boldsymbol{b}_0^{\frac{n}{4}C}\boldsymbol{b}_2^{\frac{n}{4}C}$ の垂直二等分線と FSC $\tilde{\boldsymbol{s}}(t)$ の稜線との交点 $\boldsymbol{f}$ を求め，$\boldsymbol{f}$ を頂点とする $\tilde{\boldsymbol{s}}(t)$ 上のファジィ点を抽出しこれをファジィ点 $\tilde{\boldsymbol{f}}$ とする．次に $\tilde{\boldsymbol{f}}$ の頂点位置を

$$\boldsymbol{f}^{\frac{n}{4}C}=\frac{\left\|\boldsymbol{b}_0^{\frac{n}{4}C}-\boldsymbol{m}\right\|}{\|\boldsymbol{f}-\boldsymbol{m}\|}\sqrt{\frac{1-w^{\frac{n}{4}C}}{1+w^{\frac{n}{4}C}}}\,(\boldsymbol{f}-\boldsymbol{m})+\boldsymbol{m} \tag{3.8}$$

に移動させたものをファジィ代表点 $\tilde{\boldsymbol{f}}^{\frac{n}{4}C}$ とする．ただし，$\boldsymbol{m}=(\boldsymbol{b}_0^{\frac{n}{4}C}+\boldsymbol{b}_2^{\frac{n}{4}C})/2$ とする．



(a) ファジィスプライン曲線　(b) 1/4 円形仮説ファジィモデル

(c) 2/4 円形仮説ファジィモデル　(d) 3/4 円形仮説ファジィモデル

図 3.10: FSC と三つの $n/4$ 円形仮説ファジィモデル

(2) **仮説ファジィモデルの評価** 3.2.1 (2-2) と同様に円形仮説ファジィモデル $\tilde{\boldsymbol{r}}^{\frac{n}{4}C}(t)$ $(n=1,2,3)$ がもとの FSC $\tilde{\boldsymbol{s}}(t)$ と合致する度合いを区間真理値 $[N^{\frac{n}{4}C}, P^{\frac{n}{4}C}]$ $(n=1,2,3)$ で評価する．

(3) **曲線クラスの推論** 表 3.2 のファジィ推論規則にしたがって 4 クラスのサブ曲線クラスのグレード値 $\mu(\frac{1}{4}CA)$, $\mu(\frac{2}{4}CA)$, $\mu(\frac{3}{4}CA)$, $\mu(CA)$ を算出する．それぞれのグレード値は 3.2.1 (2-3) と同様に文献 [41, 43] に従って区間 $[0,1]$ 内の実数値として計算される．ここでもっとも高いグレード値を得たサブ曲線クラスをサブ曲線クラスの同定結果として出力する．さら

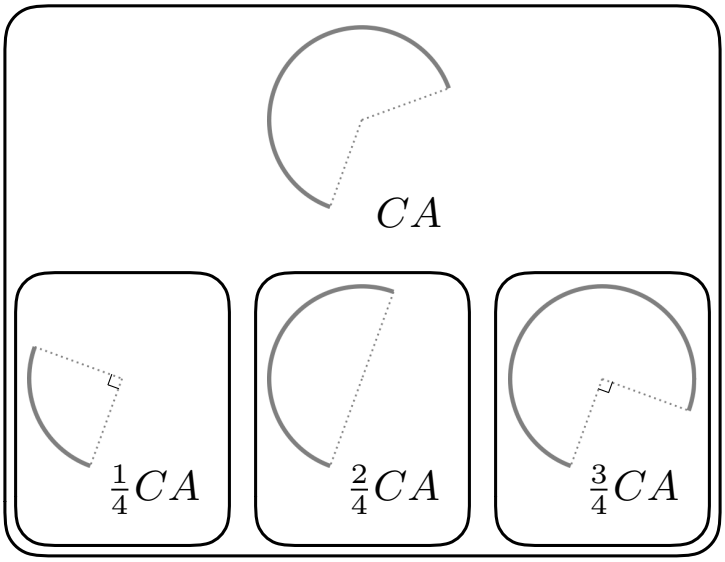


図 3.11: 曲線クラス CA およびサブ曲線クラス $\frac{1}{4}CA$, $\frac{2}{4}CA$, $\frac{3}{4}CA$ 間の自由度に関する包含関係

表 3.2: 円弧サブ曲線同定のファジィ推論規則

$\mu(\frac{1}{4}CA) = (P^{\frac{1}{4}C} \text{ is } T)$
$\mu(\frac{2}{4}CA) = (P^{\frac{2}{4}C} \text{ is } T)$
$\mu(\frac{3}{4}CA) = (P^{\frac{3}{4}C} \text{ is } T)$
$\mu(CA) \quad = (N^{\frac{1}{4}C} \text{ is } F) \wedge (N^{\frac{2}{4}C} \text{ is } F) \wedge (N^{\frac{3}{4}C} \text{ is } F)$

にここで $\frac{n}{4}CA$ が同定された場合には，$\tilde{\boldsymbol{r}}^{\frac{n}{4}C}(t)$ から形状パラメータを再計算して出力するとともに，ファジィ特徴点を $n/4$ 円形仮説ファジィモデルの始終点である $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}C}$ および $\tilde{\boldsymbol{b}}_2^{\frac{n}{4}C}$ の 2 点に置換する．

ここで図 3.11 に示すように三つのサブ曲線クラス $\frac{1}{4}CA$, $\frac{2}{4}CA$, $\frac{3}{4}CA$ は曲線クラス CA にその特殊な場合として包含されることに注意する．表 3.2 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かぎり低い自由度のサブ曲線クラス (すなわち，$\frac{1}{4}CA$, $\frac{2}{4}CA$, $\frac{3}{4}CA$) を推論するものとなっている．ユーザはこの推論規則を利用して一般の CA と $\frac{n}{4}CA$ を意図的に区別して FSC 同定法に同定させることができる．すなわち，ユーザは比較的素早い描画でファジネスの大きな FSC を生成させることで自由度の低いサブ曲線クラス $\frac{n}{4}CA$ を同定させることができる．逆にゆっくりとした丁寧な描画でファジネスの小さな FSC を生成させることで自由度の高いサブ曲線クラス CA を同定させることができる．

3.3.4 楕円弧サブ曲線同定

EA が同定された場合に，その特殊な場合である $\frac{n}{4}EA$ であるかどうかを評価し，$\frac{1}{4}EA$, $\frac{2}{4}EA$, $\frac{3}{4}EA$, EA のいずれかに細分化して再同定する．楕円弧サブ曲線同定の流れは 3.3.3 の円弧サブ曲線同定と基本的に同様である．以下に具体的な処理を示す．

(1) **仮説ファジィモデルの構成** 以下の三つの $n/4$ 楕円形仮説ファジィモデルを式 (3.6) と同じく 2 次有理 Bézier 曲線形式で構成する (図 3.12 参照)．

$$\tilde{\boldsymbol{r}}^{\frac{n}{4}E}(t)=\frac{P_0(t)\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}+P_1(t)(1+w^{\frac{n}{4}E})\tilde{\boldsymbol{f}}^{\frac{n}{4}E}+P_2(t)\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}}{P_0(t)+P_1(t)(1+w^{\frac{n}{4}E})+P_2(t)} \quad (n=1,2,3) \tag{3.9}$$

ここで，2 次有理 Bézier 曲線の性質に注意して $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}$,$\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}$, $w^{\frac{n}{4}E}$,$\tilde{\boldsymbol{f}}^{\frac{n}{4}E}$ を以下の通り定めて，$\tilde{\boldsymbol{r}}^{\frac{n}{4}E}(t)$ の稜線が $n/4$ 楕円弧となるよう制約する．

(1-1) **ファジィ代表点 $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}$ と $\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}$ の選択** $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}=\tilde{\boldsymbol{s}}$ $(n=1,2,3)$，$\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}=\tilde{\boldsymbol{e}}$ $(n=1,2,3)$ とする．

(1-2) **$w^{\frac{n}{4}E}$ の設定** $\frac{n}{4}EA$ となるように $w^{\frac{n}{4}E}=\cos\frac{n}{4}\pi$ $(n=1,2,3)$ と設定する．

(1-3) **ファジィ代表点 $\tilde{\boldsymbol{f}}^{\frac{n}{4}E}$ の設定** 以下の手順で $\tilde{\boldsymbol{f}}^{\frac{n}{4}E}$ $(n=1,2,3)$ を定める．まず，FSC $\tilde{\boldsymbol{s}}(t)$ の稜線上の点のうち直線 $\boldsymbol{b}_0^{\frac{n}{4}E}\boldsymbol{b}_2^{\frac{n}{4}E}$ から最遠となる点 $\boldsymbol{f}$ を求める．次に，$\boldsymbol{f}$ を頂点とする $\tilde{\boldsymbol{s}}(t)$ 上のファジィ点 $\tilde{\boldsymbol{f}}$ を抽出し，これをファジィ代表点 $\tilde{\boldsymbol{f}}^{\frac{n}{4}E}$ とする．

(2) **仮説ファジィモデルの評価** 3.2.1 (2-2) と同様に楕円形仮説ファジィモデル $\tilde{\boldsymbol{r}}^{\frac{n}{4}E}(t)$ $(n=1,2,3)$ がもとの FSC $\tilde{\boldsymbol{s}}(t)$ と合致する度合いを区間真理値 $[N^{\frac{n}{4}E},P^{\frac{n}{4}E}]$ $(n=1,2,3)$ で評価する．

(3) **曲線クラスの推論** 表 3.3 のファジィ推論規則にしたがって，円弧サブ曲線同定法と同様の方法 (3.3.3 (3) 参照) で，サブ曲線クラスを同定し出力する．ここで $\frac{n}{4}EA$ が同定された場合には，$\tilde{\boldsymbol{r}}^{\frac{n}{4}E}(t)$ から形状パラメータを再計算して出力するとともに，ファジィ特徴点を $\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}$,$\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}$,$\tilde{\boldsymbol{c}}^{\frac{n}{4}E}$ の 3 点に置換する．ここで，$\tilde{\boldsymbol{b}}_0^{\frac{n}{4}E}$,$\tilde{\boldsymbol{b}}_2^{\frac{n}{4}E}$ は $n/4$ 楕円形仮説ファジィモデルの始終点である．一方，$\tilde{\boldsymbol{c}}^{\frac{n}{4}E}$ については，2 次有理 Bézier 曲線の性質に注意して，n に応じて以下のとおり設定する．

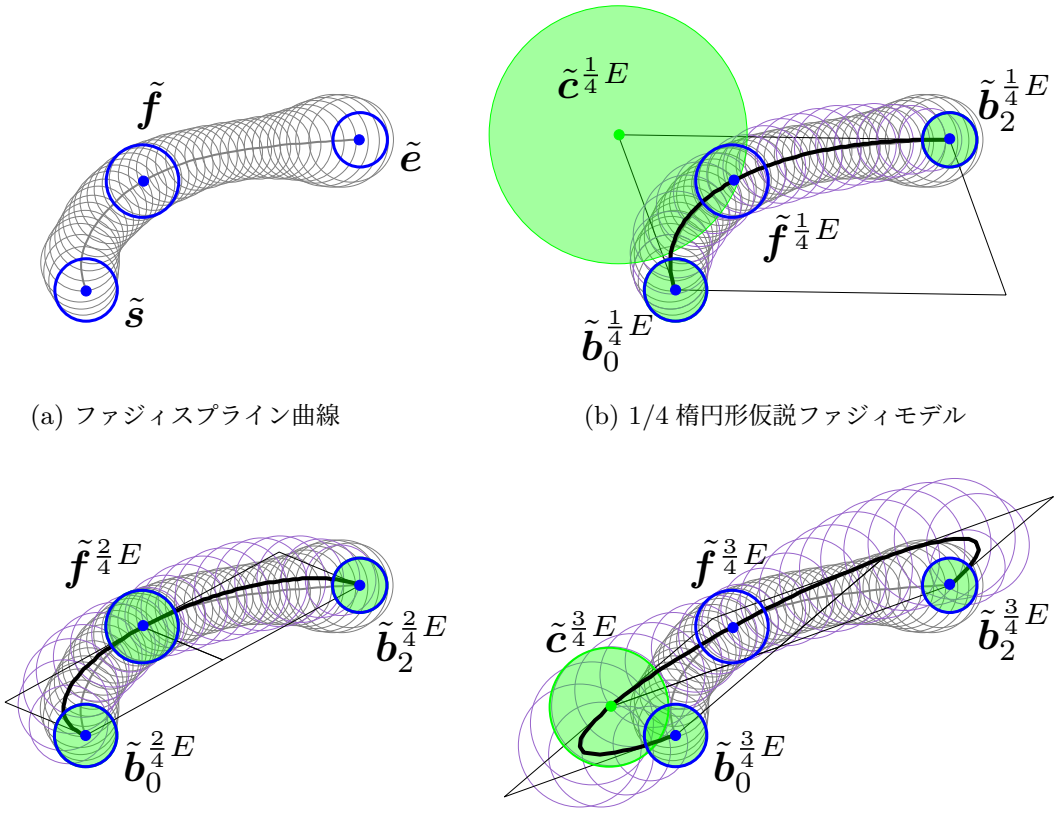
$$\tilde{\boldsymbol{c}}^{\frac{1}{4}E}=(\sqrt{2}+1)\tilde{\boldsymbol{f}}^{\frac{1}{4}E}-(\sqrt{2}/2)(\tilde{\boldsymbol{b}}_0^{\frac{1}{4}E}+\tilde{\boldsymbol{b}}_2^{\frac{1}{4}E}) \tag{3.10}$$

$$\tilde{\boldsymbol{c}}^{\frac{2}{4}E}=\tilde{\boldsymbol{f}}^{\frac{2}{4}E} \tag{3.11}$$

$$\tilde{\boldsymbol{c}}^{\frac{3}{4}E}=(2-\sqrt{2})\tilde{\boldsymbol{b}}_0^{\frac{3}{4}E}+2(\sqrt{2}-1)\tilde{\boldsymbol{f}}^{\frac{3}{4}E}+(1-\sqrt{2})\tilde{\boldsymbol{b}}_2^{\frac{3}{4}E} \tag{3.12}$$

これにより，$\tilde{\boldsymbol{c}}^{\frac{n}{4}E}$ は図 3.13 に見るように $n/4$ 楕円形仮説ファジィモデルの外接平行四辺形上のファジィ特徴点となる．

なお，表 3.3 のファジィ推論規則は円弧サブ曲線同定のファジィ推論規則 (表 3.2) と同様となっている．これにより，ユーザは比較的素早い描画でファジネスの大きな FSC を生成させることで自由度の低いサブ曲線クラス $\frac{n}{4}EA$ を同定させることができる．逆にゆっくりと



(a) ファジィスプライン曲線　(b) 1/4 楕円形仮説ファジィモデル

(c) 2/4 楕円形仮説ファジィモデル　(d) 3/4 楕円形仮説ファジィモデル

図 3.12: FSC と三つの $n/4$ 楕円形仮説ファジィモデル

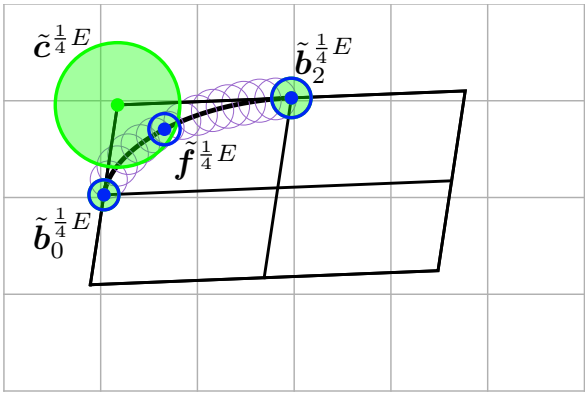
表 3.3: 楕円弧サブ曲線同定のファジィ推論規則

$\mu(\frac{1}{4}EA) = (P^{\frac{1}{4}E} \text{ is } T)$
$\mu(\frac{2}{4}EA) = (P^{\frac{2}{4}E} \text{ is } T)$
$\mu(\frac{3}{4}EA) = (P^{\frac{3}{4}E} \text{ is } T)$
$\mu(EA) \quad = (N^{\frac{1}{4}E} \text{ is } F) \wedge (N^{\frac{2}{4}E} \text{ is } F) \wedge (N^{\frac{3}{4}E} \text{ is } F)$

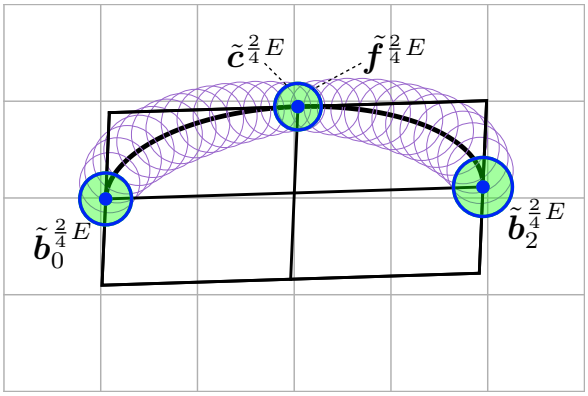
した丁寧な描画でファジネスの小さな FSC を生成させることで自由度の高いサブ曲線クラス EA を同定させることができる．

3.3.5　サブ曲線同定法の動作例

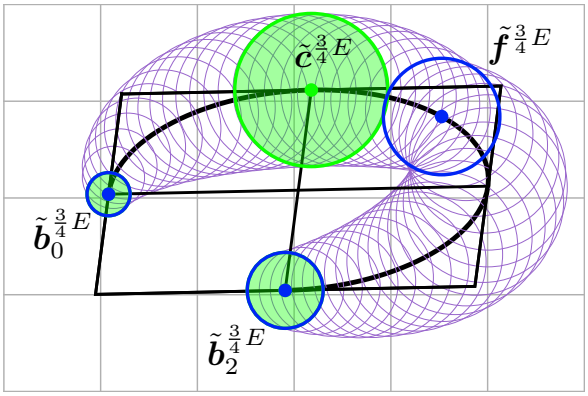
楕円弧を例にしてサブ曲線同定法の典型的な動作の様子を以下に示す．浅い楕円弧状の素早い描画から生成された図 3.14a のファジネスの大きい FSC は自由度の低いサブ曲線クラスである $\frac{1}{4}EA$ として同定される．この段階ではせん断変形しているが，MFGS がファジネスの大きなファ



(a) 1/4 楕円形仮説ファジィモデル



(b) 2/4 楕円形仮説ファジィモデル



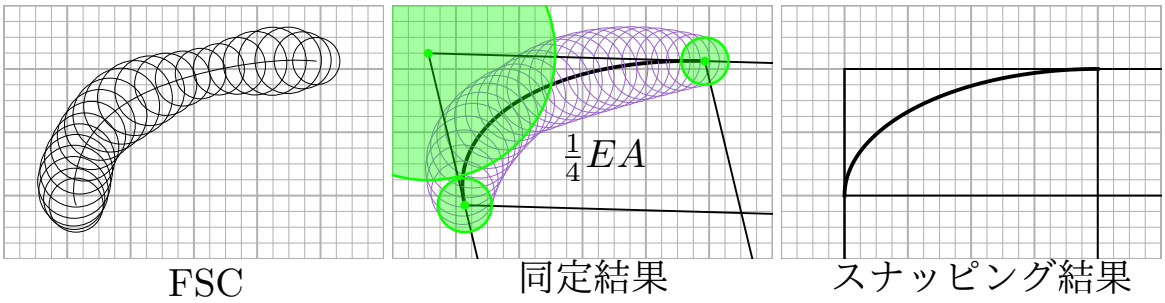
(c) 3/4 楕円形仮説ファジィモデル

図 3.13: $n/4$ 楕円形仮説ファジィモデルの外接平行四辺形上のファジィ特徴点

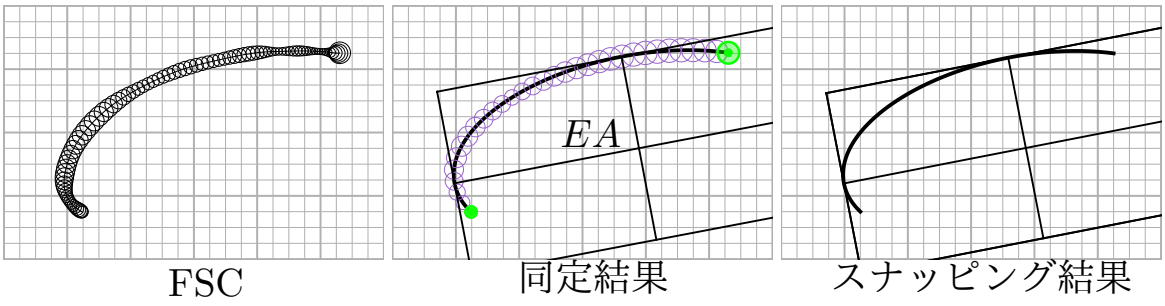
ジィ特徴点を低解像度のグリッドにスナッピングするため，最終的に外接平行四辺形が正方グリッドに重なった $\frac{1}{4}EA$ が出力される．これに対して，図 3.14b のゆっくりとした丁寧な描画から生成されたファジネスの小さな FSC は自由度の高いサブ曲線クラスである一般の EA として同定される．このとき，MFGS はファジネスの小さなファジィ特徴点を高解像度のグリッドにスナッピングするため，最終的にもとの描画形状に忠実な浅い EA が出力される．深い楕円弧状の描画についても図 3.15a と図 3.15b に示す同様の結果が得られる．このことから，ユーザは描画の素早さや丁寧さを変化させることで正方グリッドに合わせて整形された $\frac{n}{4}EA$ と描画に忠実な多様な EA を区別して入力できることがわかる．

3.4 サブ曲線同定法の有効性評価実験

提案したサブ曲線同定法の有効性を確認する評価実験を行った．実験にあたって，文献 [1] で提案された手書き CAD インターフェース SKIT にサブ曲線同定法を追加実装した．SKIT には文献 [25] の重ね書き修正機能も実装されており，たとえ最初のストローク描画で所望の同定結果が得ら



(a) 素早い描画の場合



(b) ゆっくりとした丁寧な描画の場合

図 3.14: 浅い楕円弧状描画の同定例

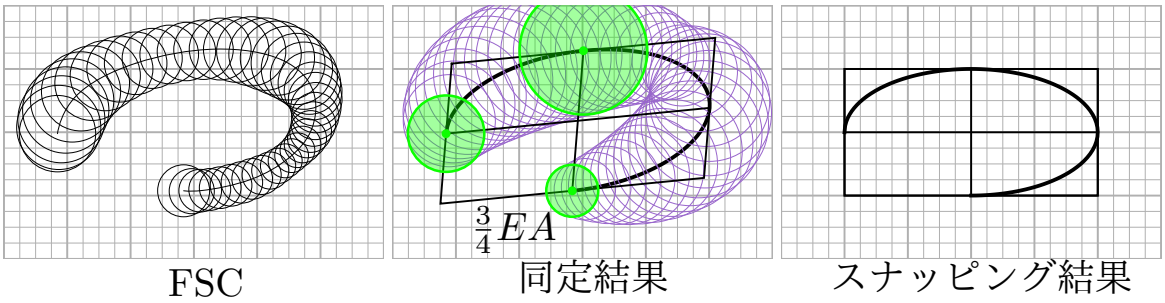
れなくても追加のストロークを重ね描くことで同定結果を逐次更新し徐々に所望の同定結果に近づけることができる．

3.4.1 幾何作図における効果の確認

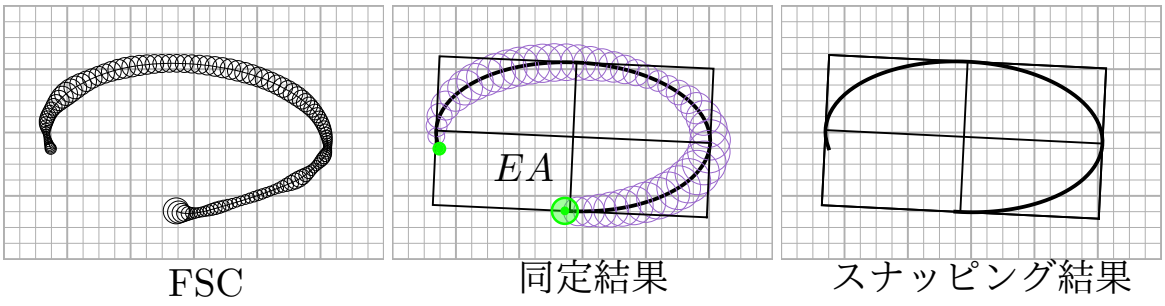
幾何作図を実際に行うことで提案したサブ曲線同定法の効果を確認した．

図 3.1 を意図した「間取り図」の幾何作図例を図 3.16，図 3.17，図 3.18 に示す．SKIT の操作に精通したユーザが SKIT を使ってストローク描画を繰り返しながらインタラクティブに作図を進めた．その結果，最終的に図 3.16 に示す 333 本の手書きストロークを描画することで図 3.17 に示す幾何作図を完成した．これに要した時間は約 15 分であった．比較のために，手書きストロークを従来の形状パラメータ量子化を用いた SKIT に入力して得た結果を図 3.18 に示す．図 3.17 の提案手法では $\frac{1}{4}CA$，$\frac{2}{4}EA$，$\frac{1}{4}EA$ が意図通りに整形されておりサブ曲線同定法が効果的に動作したことがわかる．これに対し，従来手法の図 3.18 ではこれらの多くが一般の CA や EA となり図 3.1 の作図としては不十分なものとなっている．

一方，$\frac{n}{4}CA$ と $\frac{n}{4}EA$ および一般の CA と EA を意図的に混在させた例として「橋のイラスト」の幾何作図例を図 3.19，図 3.20，図 3.21 に示す．SKIT の操作に精通したユーザ (「間取り図」を作図した者と別の一名) が作図を行った．その結果，最終的に図 3.19 に示す 242 本の手書きストロークを描画することで図 3.20 に示す幾何作図を完成した．これに要した時間は約 15 分であっ



(a) 素早い描画の場合



(b) ゆっくりとした丁寧な描画の場合

図 3.15: 深い楕円弧状描画の同定例

た．比較のために，同じ手書きストロークを従来の形状パラメータ量子化を用いた SKIT に入力して得た結果を図 3.21 に示す．提案手法の図 3.20 の結果では，$\frac{n}{4}CA$ や $\frac{n}{4}EA$ を正方グリッドに合わせて適切に配置できているとともに，「木」や「崖」の部分などを意図的に一般の CA や EA で作図することもできておりサブ曲線同定法が効果的に動作したことがわかる．これに対して，従来手法の図 3.21 では，「橋」のアーチ部分や「街灯」の部分がきれいに $\frac{n}{4}EA$ に整形されていなかったり，浅い楕円弧の形状が破綻したりしているのがわかる．これは，3.2.2 で述べたとおり特に浅い楕円弧では形状パラメータ量子化による整形が原理的に困難なことによる破綻である．

3.4.2　弧の深さと縦横比の違いによる同定性能の変化

一般に手書きで楕円弧形状を描画しようとしたとき，その扁平率が 1 に近いほど手書きストロークの形状は折れ線に近づき楕円弧として判別することが困難となる．一方その扁平率が 0 に近いほど手書きストロークの形状は円弧と判別がつきにくくなる．したがって，原理的に扁平率が 0 または 1 に近い楕円弧ほど手書きで同定させることは困難となることが予想される．ここでは，提案手法の同定性能が扁平率によってどのように変化するか確認するため，図 3.22 に示す 43 種類の目標楕円弧を用意し，これらの生成を目標としてユーザがどの程度意図通りに同定させることができるか評価した．ここで表記 a:b $[n/4]$ は図 3.23 に示すような縦横比 a:b の $\frac{n}{4}EA$ を意味するものとし，1:1 $[n/4]$ は $\frac{n}{4}CA$ を意味するものとする．同定性能を示す具体的な指標として「目標達成時

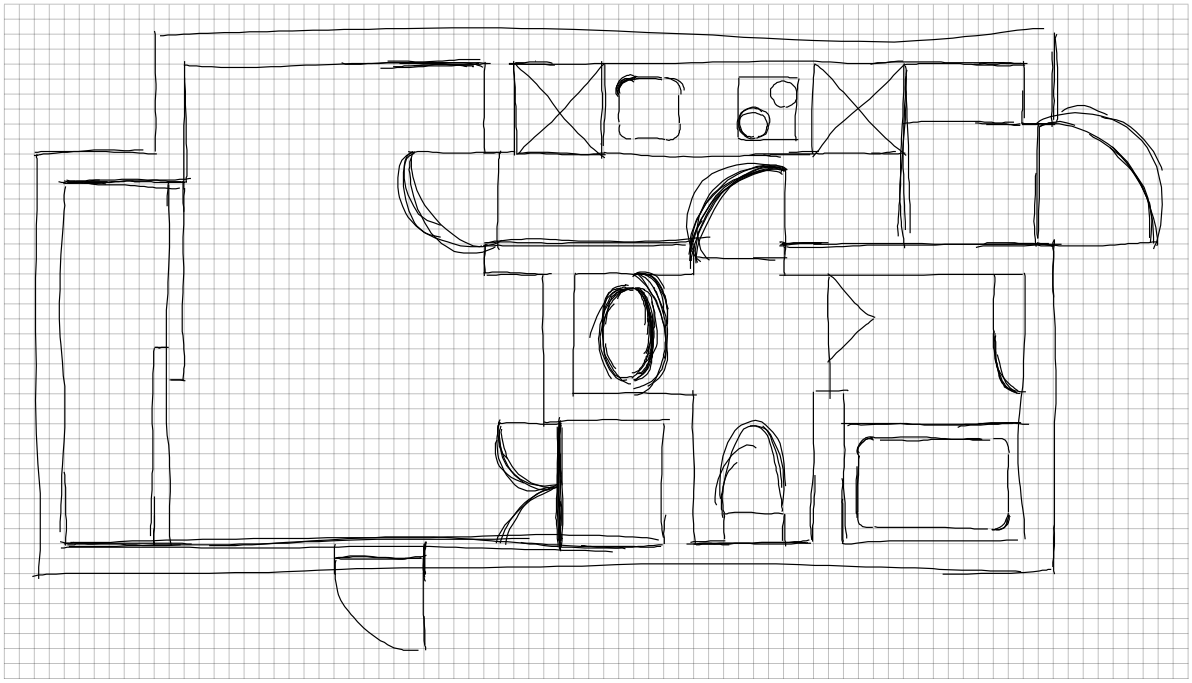


図 3.16: 「間取り図」の幾何作図のために SKIT に入力された全ての手書きストローク

間」を用いることとした．ただしここで目標達成時間とは「目標楕円弧がペンタブレット上に提示された瞬間からユーザが重ね書き修正機能を必要に応じて駆使しつつ目標楕円弧と全く同じ楕円弧を同定させることに成功する瞬間までの所要時間」を意味する．なお，一本のストロークから幾何プリミティブが同定されるまでの応答時間は 0.2 秒以下であり，リアルタイムシステムとして動作することに注意する． SKIT に精通した一名のユーザ (3.4.1 で「橋のイラスト」を作図したユーザ) にそれぞれの目標楕円弧を 10 回ずつ計 430 回ランダムに提示して測定した目標達成時間を平均した結果を図 3.22 に示す．ここで，縦横比 3:4, 4:3, 2:3, 3:2, 1:2, 2:1, 1:4, 4:1 といった扁平率が $\frac{1}{4}$ から $\frac{3}{4}$ の範囲内にある比較的簡単な縦横比の目標楕円弧については，そのほとんどが 5 秒以内の目標達成時間となっていることが分かる．一方，扁平率が 0 もしくは 1 に近づくにつれて，目標達成時間が長くなっていることが分かる．特に扁平率が $\frac{1}{6}$ と極端に 0 に近い 5:6 [1/4] や，扁平率が $\frac{11}{12}$ と極端に 1 に近い 1:12 [1/4], 1:12 [2/4] および 1:12 [3/4] では 10 秒以上の目標達成時間を要した．これは，5:6 [1/4] は $\frac{1}{4}CA$ と，1:12 [1/4] および 1:12 [2/4] は CA，FO と誤同定されやすくなり，また 1:12 [3/4] は描画の折り返し点で分割されて 2 本の幾何プリミティブとして誤同定されやすくなったためである．一方，$\frac{n}{4}CA$ (すなわち，1:1 $[n/4]$) については 2 秒程度の目標達成時間となっており，これらを $\frac{n}{4}EA$ と区別して生成できていることがわかる．

以上の結果から，提案手法は比較的簡単な縦横比の $\frac{n}{4}EA$ や $\frac{n}{4}CA$ の場合についてはそのほとんどが 5 秒以内，それ以外の場合でも扁平率が極端に 0 または 1 に近いものを除き 10 秒以内に所望の結果が得られる実用的な同定性能を持つことがわかる．

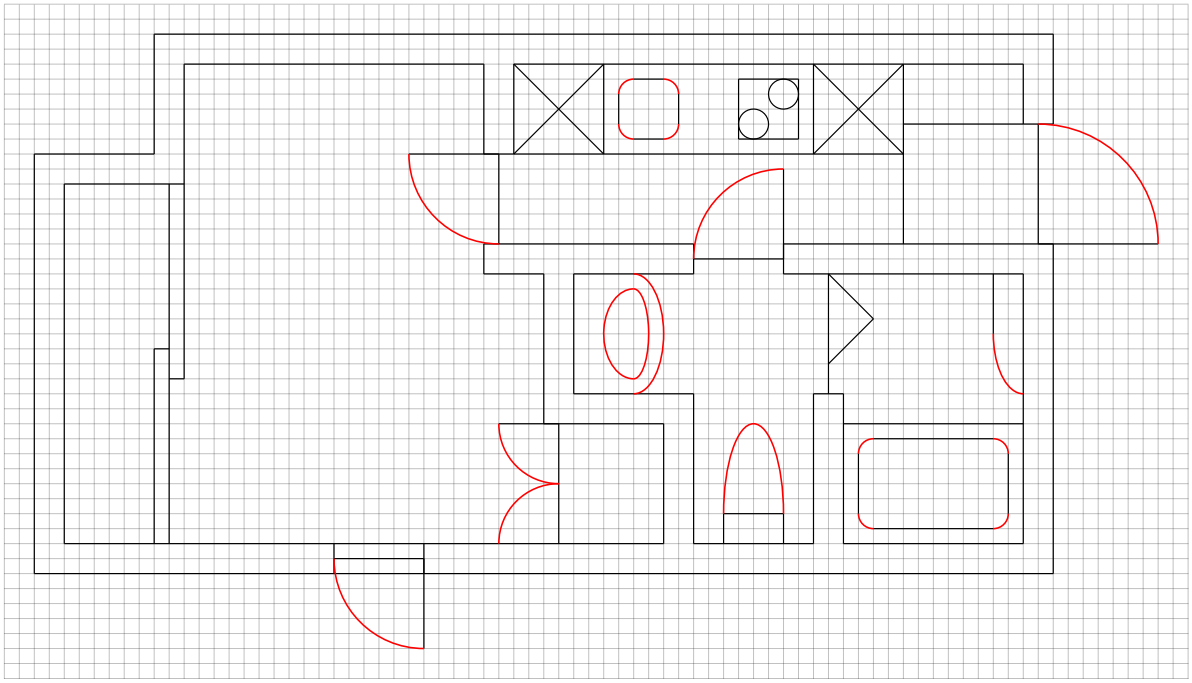


図 3.17: 図 3.16 の入力に対する提案したサブ曲線同定法を用いた SKIT の同定結果 ($\frac{n}{4}CA$ および $\frac{n}{4}EA$ は赤で表示)

3.4.3　ユーザの違いによる同定性能の変化

同定性能がユーザによってどの程度変化するか評価する実験を行った．ここでは目標楕円弧を図 3.24 に示す比較的簡単な縦横比の $\frac{n}{4}EA$ および $\frac{n}{4}CA$ 15 種類に絞って，UserA から UserF までの 6 名に対して 3.4.2 と同様の実験を 2 回ずつ行った．実験には 1 回あたり 15 分から 20 分程度かかった．このうち 1 回目の実験は練習にあてることとし，描画の素早さの程度に応じて出力結果が変化する SKIT の動作特性 (3.2.1，3.3.3，3.3.4 参照) を説明した上でこれを積極的に利用するように慣れてもらった．そののち 2 回目の実験で目標達成時間を計測し図 3.24 の結果を得た．ユーザにより得意不得意はあるものの，何れのユーザの場合でも 10 秒程度以内に所望の結果が得られる実用的な同定性能を持つことがわかる．

3.5　本章のまとめ

本章ではまず，FSC 同定法による曲線同定に基づく手書き CAD インターフェース SKIT において，弧の浅い円弧および楕円弧を形状パラメータ量子化によって正方グリッドに合わせて配置することが原理的に困難になることを，1/4 楕円弧を例にとり具体的に示した．次にこの問題を回避するために，従来の FSC 同定法の同定アルゴリズムの仮説ファジィモデルとファジィ推論規則を部分的に改変することで，形状パラメータ量子化を用いずに直接 $n/4$ 円弧および $n/4$ 楕円弧を同

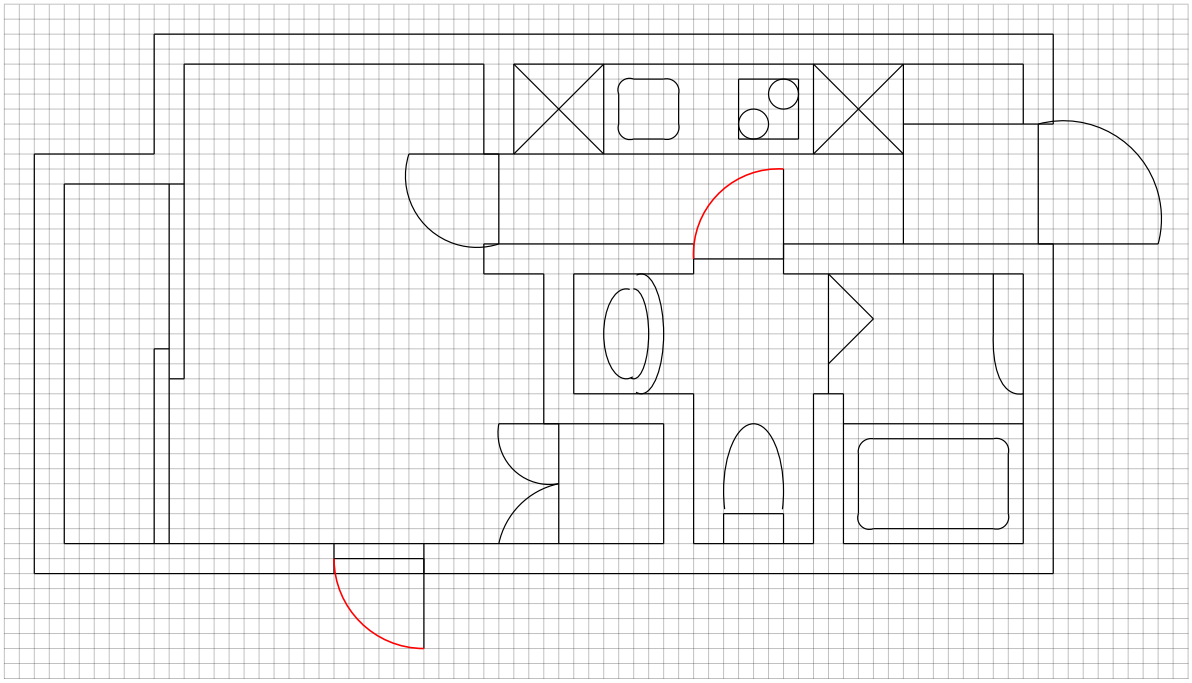


図 3.18: 図 3.16 の入力に対する従来の形状パラメータ量子化を用いた SKIT の同定結果 ($\frac{n}{4}CA$ および $\frac{n}{4}EA$ は赤で表示)

定する手法，すなわちサブ曲線同定法を提案した．さらに，形状パラメータ量子化を廃止した上で新たにサブ曲線同定法を導入することで，従来の 7 クラスの幾何プリミティブに加えて $n/4$ 円弧および $n/4$ 楕円弧をも配置できる新たな SKIT を実現した．また，新たな SKIT を用いて 6 名のユーザによる評価実験を行い，比較的簡単な縦横比の $n/4$ 円弧および $n/4$ 楕円弧の作図に関してはたとえ弧が浅くとも 10 秒程度以内で所望の結果を得られることを明らかにし，サブ曲線同定法が実用的な同定性能を有することを示し，これが手書き CAD インターフェースとして有効に機能することを示した．

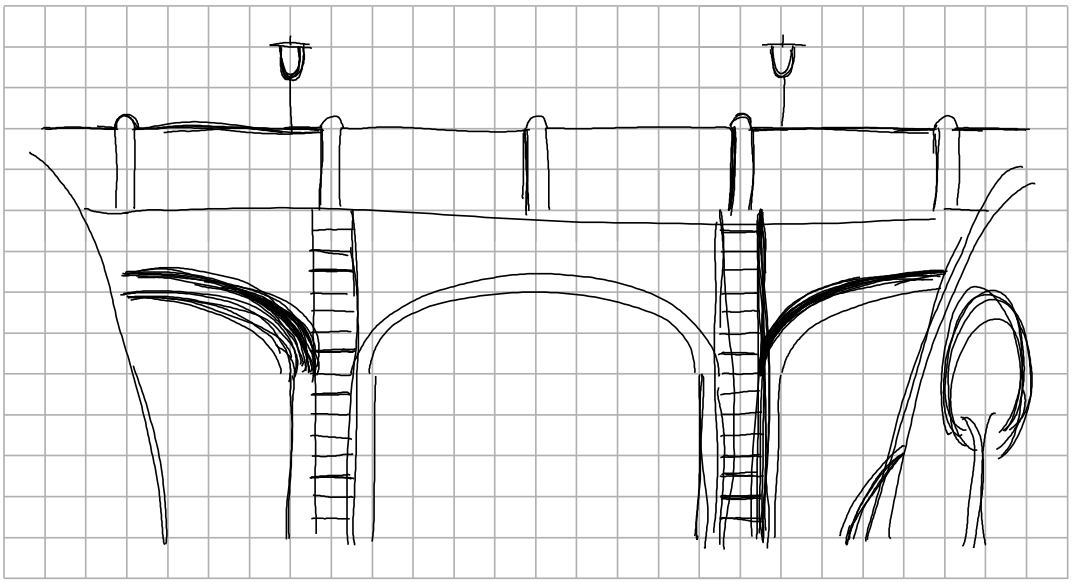


図 3.19: 「橋のイラスト」の幾何作図のために SKIT に入力された全ての手書きストローク

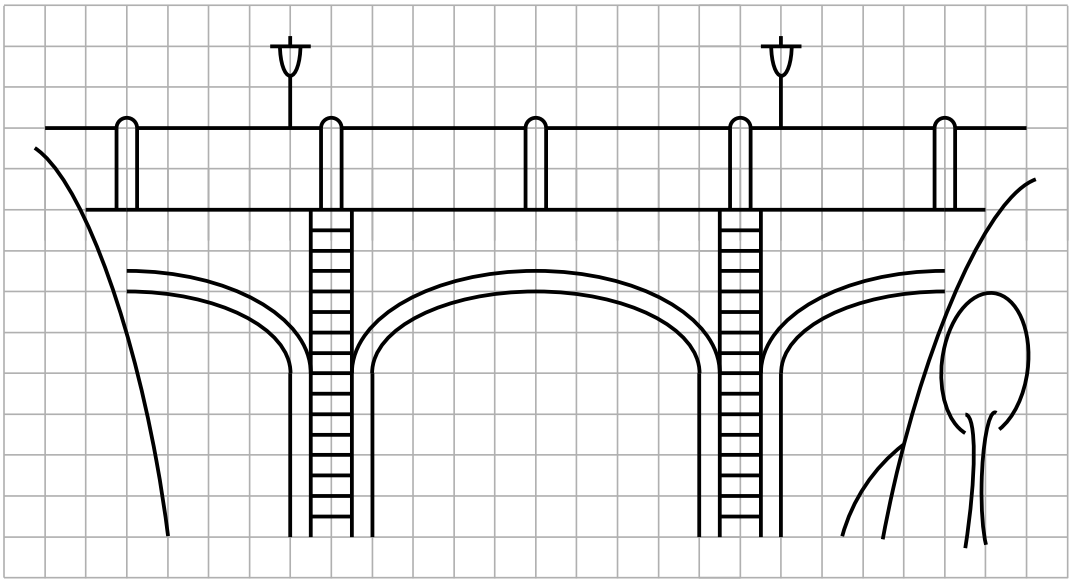


図 3.20: 図 3.19 の入力に対する提案したサブ曲線同定法を用いた SKIT の同定結果

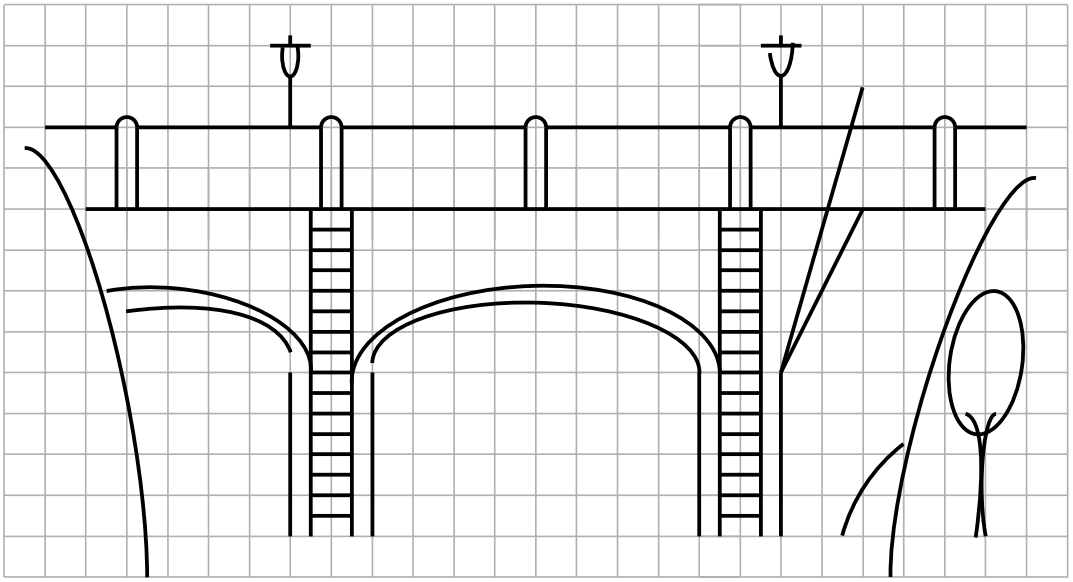


図 3.21: 図 3.19 の入力に対する従来の形状パラメータ量子化を用いた SKIT の同定結果

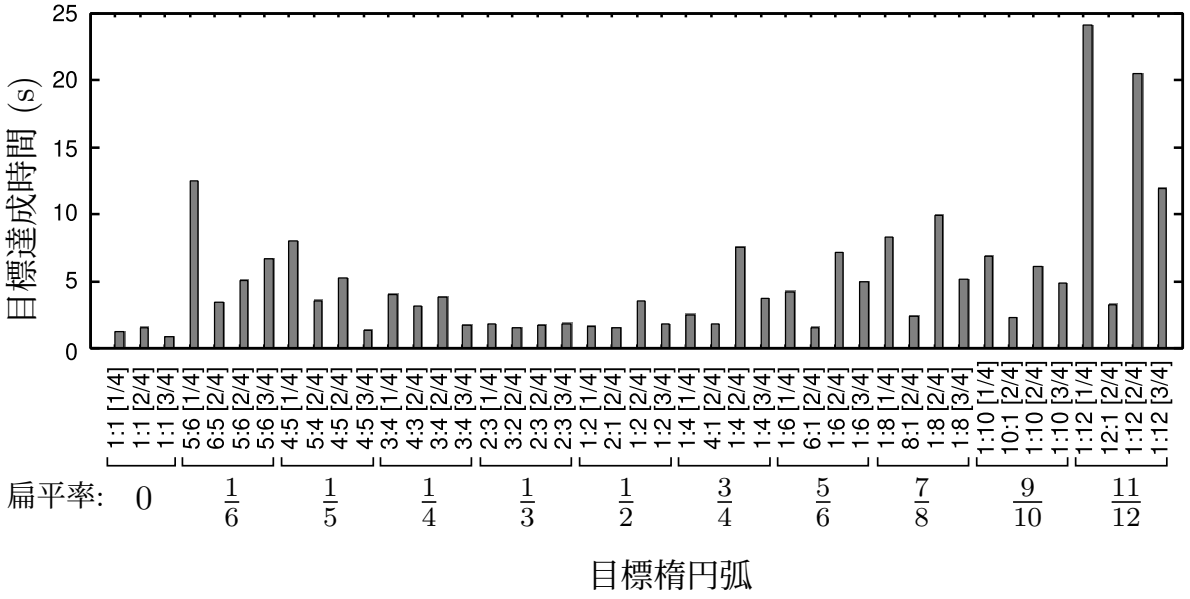
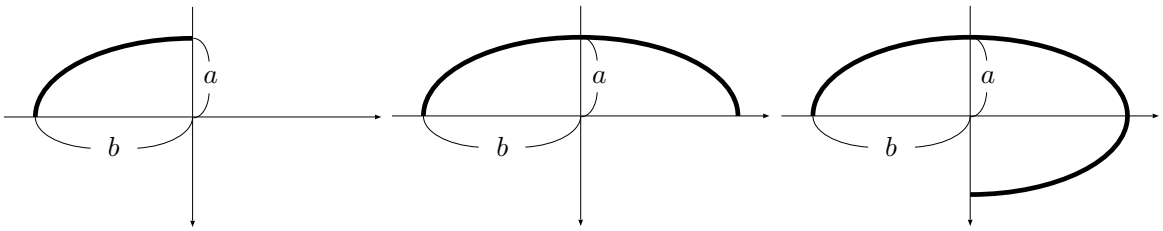


図 3.22: 弧の深さと縦横比の違いによる目標達成時間の違い



(a) a:b [1/4] (b) a:b [2/4] (c) a:b [3/4]

図 3.23: 目標楕円弧

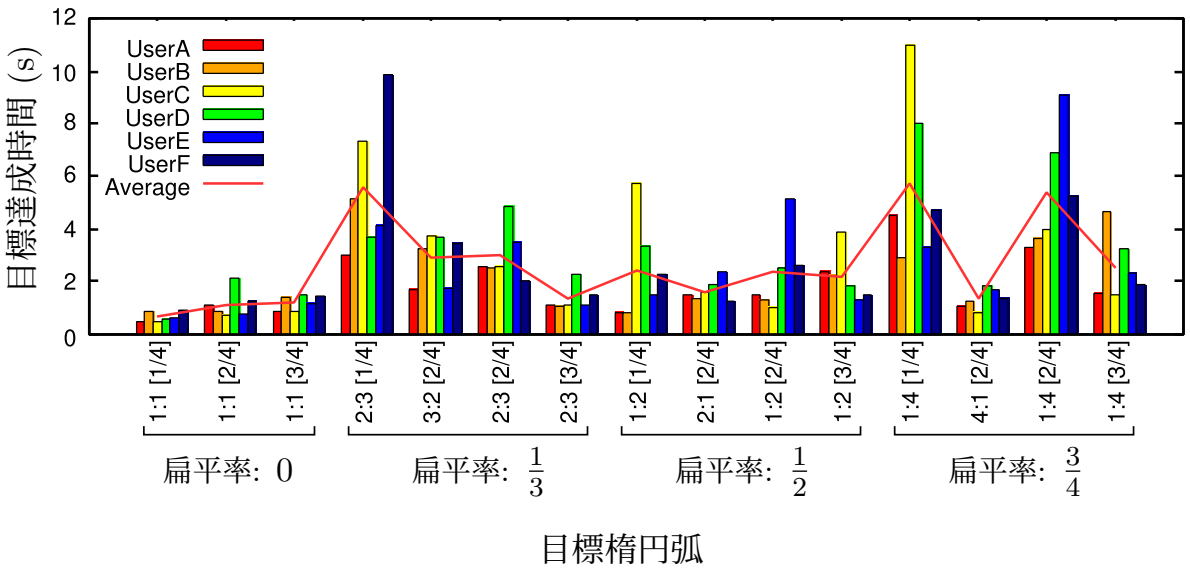


図 3.24: ユーザの違いによる目標達成時間の違い

第 4 章

ファジィ理論に基づく無限解像度グリッドスナッピング技術

ポインティングデバイスを備えた通常の CAD システムでは，ユーザはポインティング操作やドラッギング操作を通じて特徴点を 1 つずつ入力することにより幾何オブジェクトの入力を行う．そのようにして，それぞれの特徴点が所定の位置にインタラクティブにスナッピングされ，結果として幾何オブジェクトは指定されたグリッド上に配置される．ペン入力デバイスを備えた手書き CAD システムでは，ユーザが幾何オブジェクトとして同定される手書きストロークを描画すると，システムはバッチプロセスによってすべての特徴点をグリッドに自動的にスナッピングすることが必要となる．この場合，描画によって適切なグリッド解像度が変化するため，ユーザが事前にグリッド解像度を設定することは容易ではない．そこで，多重解像度ファジィグリッドスナッピング（MFGS）が提案された． MFGS では，多重解像度グリッドシステムにおける適切なスナッピング解像度は描画の雑さに応じて動的に選択される．しかしながら，MFGS のグリッド解像度の多重度は有限の数に制限されており，この解像度の範囲外のグリッドスナッピングは適切に処理されない．本章では，グリッド解像度が無限である無限解像度ファジィグリッドスナッピング（IFGS）を提案し，IFGS が MFGS の問題を効果的に解決することを実験的に示す．

4.1 本章の目的

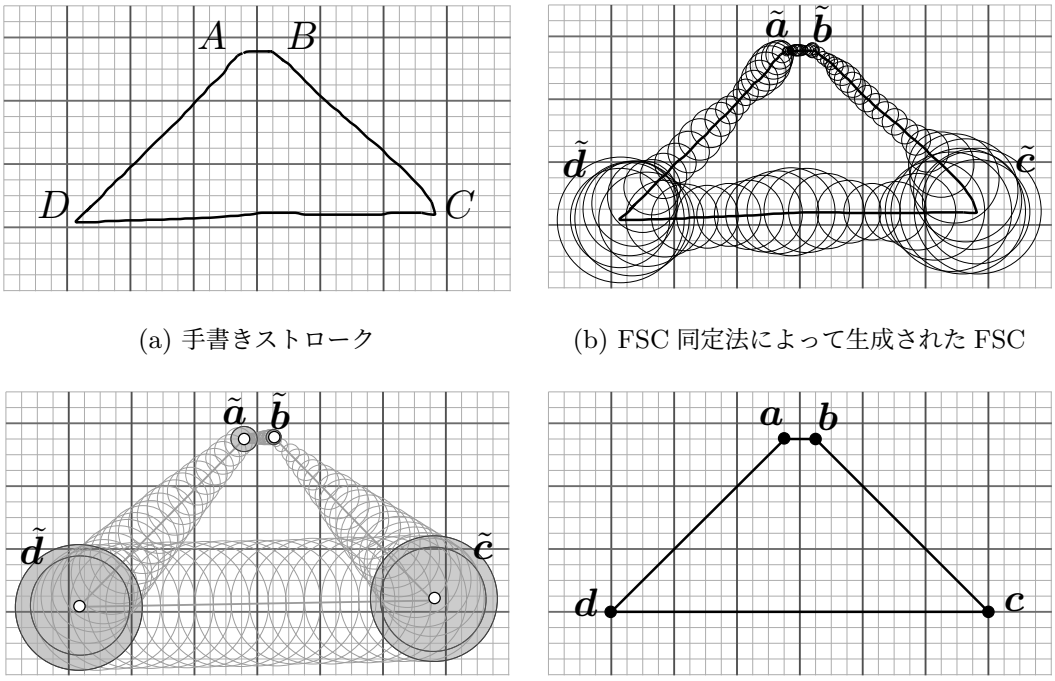
近年，タッチスクリーンや LCD タブレットの注目され，Interactive Beautification[18] や Fluid Sketches[14] などの多くの手書き作図システムが開発されている．しかし，これらのシステムで同定される幾何プリミティブの曲線クラスは，線分や円などの単純なものに限定されているため，汎用 CAD システムには適していない．汎用手書き CAD システムを実現するために，手書きス

トロークを幾何プリミティブの列として構成された幾何オブジェクトに同定できる FSC 同定法（FSC Identifier，FSC 同定法）[23, 22, 22] が提案された．それぞれの幾何プリミティブは FSC 同定法によって，手書きストロークの形状とあいまいさ（または粗さ）に基づき 7 クラスの幾何プリミティブ（線分，円，円弧，楕円，楕円弧，閉自由曲線，開自由曲線）のいずれかとして同定される．また，[29] では FSC 同定法の本質的な考え方に基づいて Fuzzy Freehand Drawing System（FFDS）が開発された．FSC 同定法が提案された後に，FSC 同定法といくつかのグリッドスナッピング手法を組み合わせて，CAD システム用の手書きインターフェースが開発された [25, 1].

CAD システムでは，幾何オブジェクトを位置合わせするためのスナッピングプロセスは幾何図形を描画する上で重要な部分である．位置合わせ機能を強化するために，Sketchpad[45], Briar[46]，Snap-Dragging[47] など，さまざまな制約を加える制約ベースの作図システムが多くの研究で検討されている．しかし，これらの研究では，制約に対する自動的な解像度の設定に焦点を当てていない．細かい幾何オブジェクトと粗い幾何オブジェクトを交互に描画するためには，ユーザは所望の位置合わせを得るために適切なスナッピング解像度に切り替える必要があり，解像度の頻繁な切り替えは本質的な描画操作の妨げとなる可能性がある．この問題を解決するために，HyperSnapping[48] と Snap-and-go[49] は，多重解像度グリッドシステムからスナッピング解像度を動的に選択することが効果的であることを示した．これらのシステムを使用すると，ユーザはそれぞれの特徴点に対して順番にポインティングとドラッギングを行うことで，グリッド上に幾何オブジェクトを配置することができる．ペンストロークごとにフィードバックが提供される手書き CAD システムでは，幾何オブジェクトの同定とそれに続く幾何オブジェクトのすべての特徴点の自動スナッピングは，ペンを持ち上げた直後にバッチプロセスにより実行される必要がある．この目的のために，[25] と [1] の手書き CAD インターフェースに複数のスナッピング解像度を提供する多重解像度ファジィグリッドスナッピング（MFGS）[31, 38] が提案された．MFGS は，FSC 同定法によって同定された幾何オブジェクトの特徴点を多重解像度グリッドにスナッピングする．この手法は，[23, 22, 22, 29, 25, 1] で利用される描画のあいまいさを用いてそれぞれ特徴点に対する適切な解像度の動的な選択を実現した．

MFGS を使用すると，ユーザは幅広い作図状況においてグリッド解像度を切り替えるという面倒な操作を回避できる．しかし，利用可能な解像度の数は有限であるため，MFGS は定義された範囲外の解像度へのスナッピングを適切に処理できない．したがって，ユーザは，作図状況がその解像度設定と相容れなくなるたびに，多重解像度グリッドを再設定しなければならない．この問題に対処するもっとも簡単な方法は，MFGS の解像度の数を増やして，考えられるすべての描画が十分な余裕を持ってカバーされるようにするである．しかしながら，解像度の数が増えると，ブルートフォースアプローチに基づく MFGS の計算効率が大幅に低下してしまう．

この問題を解決するために，本章ではまず解像度の数を無限となるように増やして MFGS を拡張した無限解像度ファジィグリッドピング（IFGS）の概念を提案する．次に，ファジィグリッドの特性を解明することにより無限の数の可能な解像度から適切な解像度を選択するための効率的な探索アルゴリズムを提案する．さらに，作図状況が大きく変化した場合でも，IFGS では多重解像度グリッドの再設定の必要性を回避できることを実験的に示す．



(a) 手書きストローク

(b) FSC 同定法によって生成された FSC

(c) FSC 同定法によって同定されたファジィ幾何プリミティブ

(d) MFGS によって多重解像度グリッドに配置された幾何オブジェクト

図 4.1: FSC 同定法による幾何オブジェクト同定と MFGS による多重解像度グリッドスナッピング

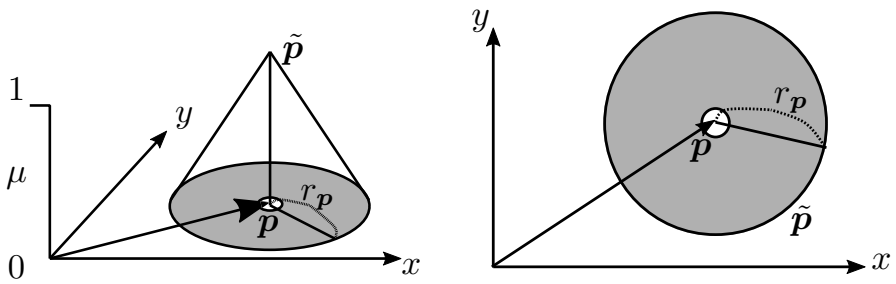
4.2 多重解像度ファジィグリッドスナッピングのオーバービュー

提案手法を説明するために，まず，IFGS の基礎となる MFGS の強み，グリッド設定，アルゴリズム，問題についてまとめる．

4.2.1 MFGS の強み

MFGS[31, 38] は，FSC 同定法 [22] によって同定された幾何オブジェクトのファジィ特徴点を多重解像度グリッドにスナッピングする．以下では，FSC 同定法による幾何オブジェクトの同定と MFGS による幾何オブジェクトの配置のプロセスについて，図 4.1 に示す多重度が 2 である多重解像度グリッドの例を用いながら説明する．

(1) 手書きストロークが，ユーザによりペンを用いて入力される．図 4.1(a) に示す例では，台形を描画することを意図しているユーザが A，B，C，D，および A を通過する手書きストロークを描画している．



(a) 投影図　　(b) 上面図

図 4.2: 円錐型ファジィ点 $\tilde{\boldsymbol{p}}$

(2) ペンが持ち上げられた後すぐに，図 4.1(b) に示すように，FSC 同定法は描画動作の雑さの程度に応じて手書きストロークに位置情報のあいまいさ（ファジネス）を追加することにより，ファジィスプライン曲線（FSC）を生成する．ここで FSC は，図 4.2 に示すような，移動する円錐型ファジィ点 $\tilde{\boldsymbol{p}} = \langle \boldsymbol{p}, r_{\boldsymbol{p}} \rangle$ の軌跡として表現される．この $\tilde{\boldsymbol{p}}$ は円錐型のメンバシップ関数

$$\mu_{\tilde{\boldsymbol{p}}}(\boldsymbol{v}) = \left\{ 1 - \frac{||\boldsymbol{v} - \boldsymbol{p}||}{r_{\boldsymbol{p}}}, 0 \right\} \tag{4.1}$$

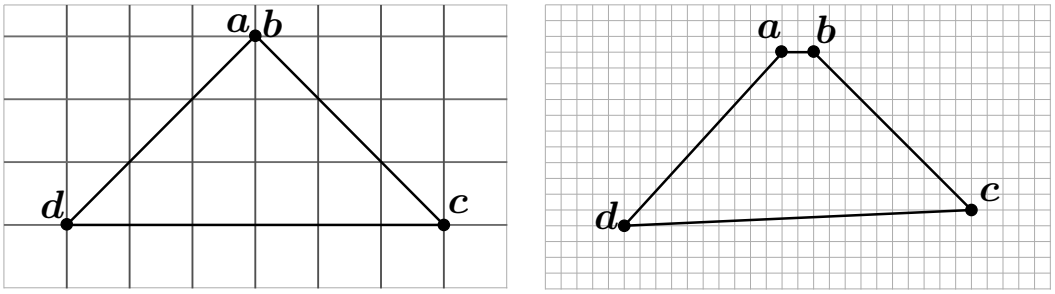
によって特徴付けられるファジィ集合である．ただし，$\boldsymbol{v}$ は変数位置ベクトル，$\boldsymbol{p}$ は $\tilde{\boldsymbol{p}}$ の位置ベクトル，$r_{\boldsymbol{p}}$ は $\tilde{\boldsymbol{p}}$ のファジネス[*1]である．

(3) FSC 同定法は FSC を同定し，1 つ以上のファジィ幾何プリミティブを生成する．図 4.1(c) に示す例では 4 本のファジィ線分 $\tilde{\boldsymbol{a}}\tilde{\boldsymbol{b}}$，$\tilde{\boldsymbol{b}}\tilde{\boldsymbol{c}}$，$\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}$，$\tilde{\boldsymbol{d}}\tilde{\boldsymbol{a}}$ が生成されている．

(4) MFGS は，ファジィ幾何プリミティブの全てのファジィ特徴点をバッチプロセスにより同時に多重解像度グリッドにスナッピングすることで，幾何プリミティブの列として構成される幾何オブジェクトを配置する．図 4.1(d) に示す例では，4 つのファジィ特徴点 $\tilde{\boldsymbol{a}}$，$\tilde{\boldsymbol{b}}$，$\tilde{\boldsymbol{c}}$，$\tilde{\boldsymbol{d}}$ がそれぞれ 4 つのグリッド点 $\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{c}$，$\boldsymbol{d}$ に同時にスナッピングされている．ここで MFGS は，ファジネスのより大きいファジィ点（$\tilde{\boldsymbol{c}}$ と $\tilde{\boldsymbol{d}}$）ほどより低い解像度のグリッドにスナッピングし，ファジネスのより小さいファジィ点（$\tilde{\boldsymbol{a}}$ と $\tilde{\boldsymbol{b}}$）ほどより高い解像度のグリッドにスナッピングするという戦略に従って，それぞれの特徴点のスナッピンググリッドの解像度を動的に選択する．

MFGS を使用せずに幾何オブジェクトを単に単一解像度のグリッドにスナッピングするだけの場合，図 4.3 に示すように，配置結果がユーザの意図と一致しないことがしばしばある．つまり，図 4.3(a) に示す低解像度のグリッドを使用すると，線分 $\boldsymbol{ab}$ などの描画の細かい部分が点に縮退する

[*1] $r_{\boldsymbol{p}}$ はファジネス生成モデル [35] によって生成される．ファジネス生成モデルは描画動作の速度と加速度に基づいて，小さなファジネス（細かいポジショニング）を曲線の注意深く描画された部分に関連付け，大きなファジネス（粗いポジショニング）を大まかに描画された部分に関連付ける．



(a) 低解像度グリッドを用いた場合の縮退　(b) 高解像度グリッドを用いた場合の微妙なずれ

図 4.3: 単一解像度グリッドスナッピングの問題

ことがある．しかしながら，図 4.3(b) に示す高解像度グリッドを使用すると，線分 $\boldsymbol{cd}$ などの描画の粗い部分が微妙にずれることがある．対照的に，MFGS は，それぞれのファジィ特徴点のファジネスに応じてスナッピング解像度を動的に選択するため，それぞれの部分の描画の雑さの程度に応じて適切なスナッピングが得られる．その結果，図 4.1(a) に示す手書きストロークから図 4.1(d) に示す台形を得ることに成功する．

4.2.2 MFGS のグリッド設定

MFGS を用いる場合，ユーザは n 層のファジィグリッド G_i $(i = 1, 2, \ldots, n)$ を組み合わせることにより，多重度 n の多重解像度ファジィグリッド（MFG）を設定する．ここでそれぞれの G_i は，図 4.4 に示すパターンでファジィ点が配置されたファジィグリッドである．任意の原点を決定すれば，これらのファジィグリッドはプロパティのペア (S_{G_i}, r_{G_i}) で表される．ただし，S_{G_i} はグリッド間隔，r_{G_i} はグリッド上に配置されたファジィ点のファジネスである．すなわち，MFG は次のパラメータで設定されることとなる．

- 多重度：n
- G_1 のプロパティ：$(S_{G_1}, r_{\boldsymbol{G}_1})$
- G_2 のプロパティ：$(S_{G_2}, r_{\boldsymbol{G}_2})$
  $\vdots$
- G_n のプロパティ：$(S_{G_n}, r_{\boldsymbol{G}_n})$

ここで，G_i $(i = 1, 2, \ldots, n)$ が解像度に関して降順に並ぶようにグリッドのインデックス i を割り当てるものとするとき，常に $S_{G_i} < S_{G_{i+1}}$ $(i = 1, 2, \ldots, n-1)$ および $r_{G_i} < r_{G_{i+1}}$ $(i = 1, 2, \ldots, n-1)$ が成り立つようにパラメータを設定することとする．

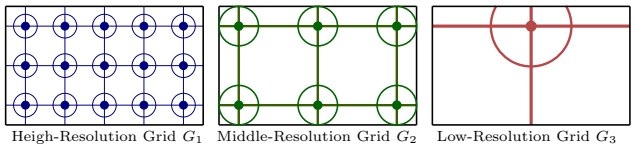


図 4.4: 多重度 3 の多重解像度ファジィグリッド（MFG）を構成するファジィグリッド

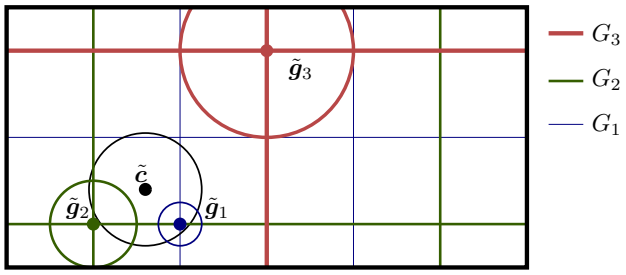


図 4.5: 多重度 3 の MFG 上のファジィ特徴点 $\tilde{\boldsymbol{c}}$ とスナッピング候補 $\tilde{\boldsymbol{g}}_i$ $(i = 1, 2, 3)$

4.2.3 MFGS のアルゴリズム

以下では，図 4.5 に示す例を使用して MFGS のアルゴリズムを説明する．この例では，ファジィ特徴点 $\tilde{\boldsymbol{c}}$ が 3 つのファジィグリッド G_i $(i = 1, 2, 3)$ で構成される多重度 3 の MFG にスナッピングされることとなる．

(1) **スナッピング候補の生成：** それぞれの i $(i = 1, 2, \ldots, n)$ に対して，特徴点の位置 $\boldsymbol{c}$ にもっとも近い G_i のグリッド点から 1 つのグリッド点を選択し，$\boldsymbol{g}_i$ $(i = 1, 2, \ldots, n)$ という名前を付ける．次に，スナッピング候補 $\tilde{\boldsymbol{g}}_i = \langle \boldsymbol{g}_i, r_{\boldsymbol{g}_i} \rangle = \langle \boldsymbol{g}_i, r_{G_i} \rangle$ $(i = 1, 2, \ldots, n)$ を生成する．

(2) **必然性値の評価：** それぞれのスナッピング候補 $\tilde{\boldsymbol{g}}_i$ に対して，ファジィ理論 [36, 50] の必然性測度に基づいて必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ を評価する．ただし，$N^{\tilde{\boldsymbol{g}}_i}$ はファジィ命題「$\tilde{\boldsymbol{g}}_i$ が $\tilde{\boldsymbol{c}}$ に含まれる」を表しており，次の式で計算される．

$$N^{\tilde{\boldsymbol{g}}_i} = \max\left\{\frac{r_{\boldsymbol{c}} - \|\boldsymbol{g}_i - \boldsymbol{c}\|}{r_{\boldsymbol{c}} + r_{\boldsymbol{g}_i}}, 0\right\} \tag{4.2}$$

(3) **グレードの計算：** 表 4.1 に示されるファジィ論理演算を適用することにより，それぞれの $\tilde{\boldsymbol{g}}_i$ のグレード $\mu(\tilde{\boldsymbol{g}}_i)$ と $\tilde{\boldsymbol{c}}$ のグレード $\mu(\tilde{\boldsymbol{c}})$ を計算する．ただしこの表において，記号 $\wedge$ は min 演算で計算される論理積を表す．また，$(1 - N^{\tilde{\boldsymbol{g}}_i})$ は $N^{\tilde{\boldsymbol{g}}_i}$ の論理否定を表す．

(4) **スナッピング：** $\tilde{\boldsymbol{g}}_i$ $(i = 1, 2, \ldots, n)$ の中でもっとも高いグレードを持つスナッピング候補を決定し，それを $\tilde{\boldsymbol{g}}_k$ として同定する．次に，ファジィ特徴点 $\tilde{\boldsymbol{c}}$ がグリッド点 $\boldsymbol{g}_k$ にスナッピングされる．ここで，$\mu(\tilde{\boldsymbol{c}}) > \mu(\tilde{\boldsymbol{g}}_k)$ の場合は，$\tilde{\boldsymbol{c}}$ はそれ自体の位置 $\boldsymbol{c}$ にスナッピングされる．これは，いわゆる「ノースナッピング」が選択されることを意味する．図 4.5 に示す例では，$\tilde{\boldsymbol{g}}_1$ のグレードがもっとも高くなるため，$\tilde{\boldsymbol{c}}$ は $\boldsymbol{g}_1$ にスナッピングされる．

表 4.1: MFGS のファジィ推論規則 [31]

$\mu(\tilde{\boldsymbol{g}}_3)$	=	$N^{\tilde{\boldsymbol{g}}_3}$				
$\mu(\tilde{\boldsymbol{g}}_2)$	=	$(1-N^{\tilde{\boldsymbol{g}}_3})$	$\wedge$	$N^{\tilde{\boldsymbol{g}}_2}$		
$\mu(\tilde{\boldsymbol{g}}_1)$	=	$(1-N^{\tilde{\boldsymbol{g}}_3})$	$\wedge$	$(1-N^{\tilde{\boldsymbol{g}}_2})$	$\wedge$	$N^{\tilde{\boldsymbol{g}}_1}$
$\mu(\tilde{\boldsymbol{c}})$	=	$(1-N^{\tilde{\boldsymbol{g}}_3})$	$\wedge$	$(1-N^{\tilde{\boldsymbol{g}}_2})$	$\wedge$	$(1-N^{\tilde{\boldsymbol{g}}_1})$

全体として，表 4.1 で与えられたファジィ論理演算は，ファジィ特徴点をその必然性値が十分に高いものの中でもっとも低い解像度のグリッドにスナッピングするように機能する．そのようにして，より雑な描画から得られたファジィ特徴点はより低い解像度のグリッドにスナッピングされ，より丁寧な描画から得られた特徴点はより高い解像度のグリッドにスナッピングされる，という MFGS の動的なスナッピングが実現される．表 4.1 は，MFGS が，それぞれの特徴点をスナッピングするための必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ への参照が $\frac{n(n+1)}{2}+n$ 回必要となることを示している．

4.2.4 MFGS の問題

描画される幾何オブジェクトのサイズの範囲に適した適切な MFG の設定を使用すると，ユーザは作図中にグリッド解像度を切り替えるという面倒な操作を回避できる．しかしながら，適切ではない MFG の設定では，ユーザは多重度とそれぞれのファジィグリッドのプロパティを制御するパラメータをリセットする必要が生じてしまう．MFGS では，これらの操作は通常の単一解像度のグリッドスナッピングよりも面倒である．

実際，手書きインターフェースでは，グリッドの解像度と描画するオブジェクトのサイズ分布との相対的な関係を事前に判断するのは必ずしも簡単ではない．この困難性の原因の一つは，描画された幾何オブジェクトのサイズが大きく異なるかもしれないという事実である．手書き入力によって幾何オブジェクトをクリエイティブに描画する時，ユーザは多くの場合，小さな幾何オブジェクトと大きな幾何オブジェクトをカバーするために，広いダイナミックレンジで手書きストロークのサイズを変更する．加えて，ユーザは図 4.1 で示した 4 本の線のように一つの手書きストロークの中でいくつかの異なるサイズの幾何オブジェクトを入力することもできる．それゆえに，手書きストロークの描画に先立って幾何オブジェクトのサイズの分布を事前に決定することは困難である．二つ目の原因は，個別のペン入力デバイスによって画面サイズや解像度が大きく異なることである．この原因のため，もしある特定のデバイスで適切なグリッド解像度が設定されていても，別のデバイスでは物理的なグリッド解像度が大幅に変わる可能性がある[*2]．そのため，ペン入力デバイスを指定することなく，描画される幾何オブジェクトの物理的なサイズに対して，グリッド解像度のプロパティをピクセル単位で決定することはできない[*3]．例としてユーザが 50 ピクセルのグ

[*2] 今日では，CAD システムはスマートフォンやタブレットから電気黒板までさまざまなデバイスで利用可能であり，ユーザは用途に応じてこれらのデバイスを自由に切り替えることができる．

[*3] ユーザは，（アプリケーションプログラムとして）インターフェースがオペレーティングシステムから標準的な方法

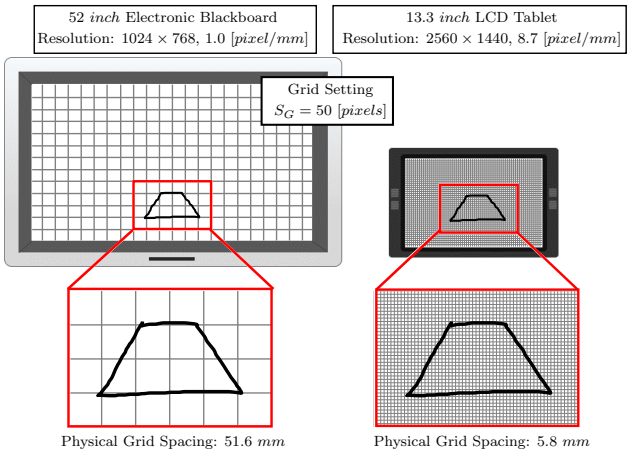


図 4.6: Difference in physical grid spacing between two different pen-input devices.

リッド間隔を設定し，図 4.6 に示すように幅が約 200 mm の台形を描画することを想定する．52 inch の電子黒板を使用する場合，このグリッド解像度の設定は物理的なグリッド間隔が 51.6 mm になるため適切である．対照的に，13.3 inch の LCD タブレットを使用する場合，物理的なグリッド間隔が 5.8 mm になるため，同じグリッド解像度の設定では高すぎる．

以上の説明は，ユーザがさまざまなペン入力デバイスを使用して創造的にスケッチする現実的な状況において，適切な MFG パラメーターを事前に設定することが容易ではないことを明確にしている．MFGS は，描画される幾何オブジェクトのサイズの分布や使用するペン入力デバイスが事前に定義されている限られた範囲の状況で有効である．しかし，MFG をリセットする必要が頻繁にあると，これにより実用における本質的な作図の流れが妨げられてしまう．

この問題に対処するために，ユーザは多重度 n を増やすことですべての可能な作図状況をカバーすることができる．しかしながら 4.2.3 で述べたように，n が増加すると必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ への参照回数が $O(n^2)$ の時間計算量で増加し，計算効率が大幅に低下する．

4.3 無限解像度ファジィグリッドスナッピングの提案

MFGS の問題を解決するために，可能なグリッド解像度の多重度を無限の範囲に拡張する IFGS について説明する．

4.3.1 IFGS のグリッド設定

任意の原点を決定すれば，IFGS では無限解像度ファジィグリッド（IFG）を，次の 2 つのパラメータのセットを用いて設定する．

でペン入力デバイスの物理的な解像度を取得できる場合，物理的な距離として適切なグリッド解像度を設定できる．しかしながら，これは一般的なオペレーティングシステムで常に当てはまるとは限らない．

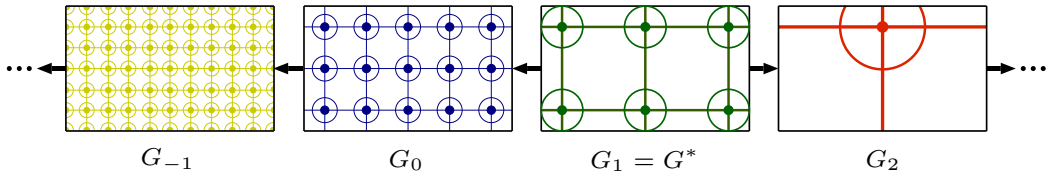


図 4.7: グリッド解像度の拡大率が 2 である IFG を構成するファジィグリッド

表 4.2: IFGS のファジィ推論規則

⋮	⋮
$\mu(\tilde{\boldsymbol{g}}_{i+1})$	$= (1 - N^{\tilde{\boldsymbol{g}}_\infty}) \wedge \cdots \wedge (1 - N^{\tilde{\boldsymbol{g}}_{i+2}}) \wedge N^{\tilde{\boldsymbol{g}}_{i+1}}$
$\mu(\tilde{\boldsymbol{g}}_i)$	$= (1 - N^{\tilde{\boldsymbol{g}}_\infty}) \wedge \cdots \wedge (1 - N^{\tilde{\boldsymbol{g}}_{i+1}}) \wedge N^{\tilde{\boldsymbol{g}}_i}$
$\mu(\tilde{\boldsymbol{g}}_{i-1})$	$= (1 - N^{\tilde{\boldsymbol{g}}_\infty}) \wedge \cdots \wedge (1 - N^{\tilde{\boldsymbol{g}}_i}) \wedge N^{\tilde{\boldsymbol{g}}_{i-1}}$
$\mu(\tilde{\boldsymbol{g}}_{i-2})$	$= (1 - N^{\tilde{\boldsymbol{g}}_\infty}) \wedge \cdots \wedge (1 - N^{\tilde{\boldsymbol{g}}_{i-1}}) \wedge N^{\tilde{\boldsymbol{g}}_{i-2}}$
⋮	⋮

- 基準グリッド G^* のプロパティ: (S_{G^*}, r_{G^*})
- グリッド解像度の拡大率: f

これらのパラメータを使用して，IFGS は無限の個数のファジィグリッド G_i（$i \in \mathbb{Z}$，ここで $\mathbb{Z}$ は整数の集合を表す.）のプロパティを以下の式で生成できる.

$$(S_{G_i}, r_{G_i}) = (S_{G^*} \times f^{i-1}, r_{G^*} \times f^{i-1}) \tag{4.3}$$

ここで，G_i $(i \in \mathbb{Z})$ が解像度に関して降順に並ぶようにグリッドのインデックス i を割り当てるものとするとき，常に $f > 1$ が成り立つように f を設定することとする．図 4.7 は，$f = 2$ である IFG の例を示している.

4.3.2 IFGS のアルゴリズム

表 4.2 に示すように IFGS では無数のファジィルールが適用する[*4]．これは表 4.1 で与えられた MFGS の有限の個数のファジィルールの拡張である．MFGS と同様に，IFGS は表 4.2 に従って $\tilde{\boldsymbol{g}}_i$ $(i \in \mathbb{Z})$ の中でもっとも高いグレードを持つスナッピング候補を決定し，これを $\tilde{\boldsymbol{g}}_k$ として同定する．次にファジィ特徴点 $\tilde{\boldsymbol{c}}$ がグリッド点 $\boldsymbol{g}_k$ にスナッピングされる.

ここで問題となるのは，それぞれが無限の数の項を持つ無限の数のルールから，どのようにグレードがもっとも高くなるスナッピング候補を見つけるかということである．$i < j$ の場合に

[*4] MFGS におけるノースナッピングのグレード $\mu(\tilde{\boldsymbol{c}})$ は，無限に高解像度のグリッドを利用できる IFGS では無意味であることに注意する.

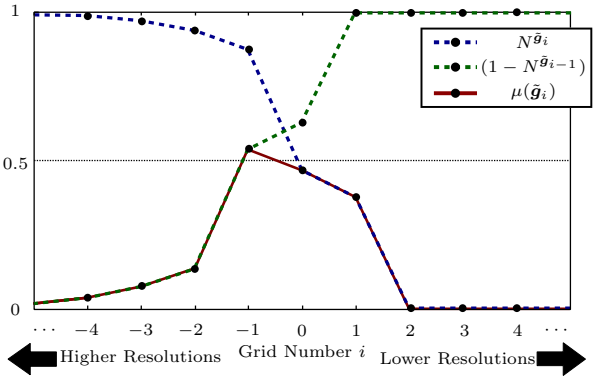


図 4.8: IFGS における $N^{\tilde{g}_i}$，$(1 - N^{\tilde{g}_{i+1}})$，$\mu(\tilde{g}_i)$ の遷移

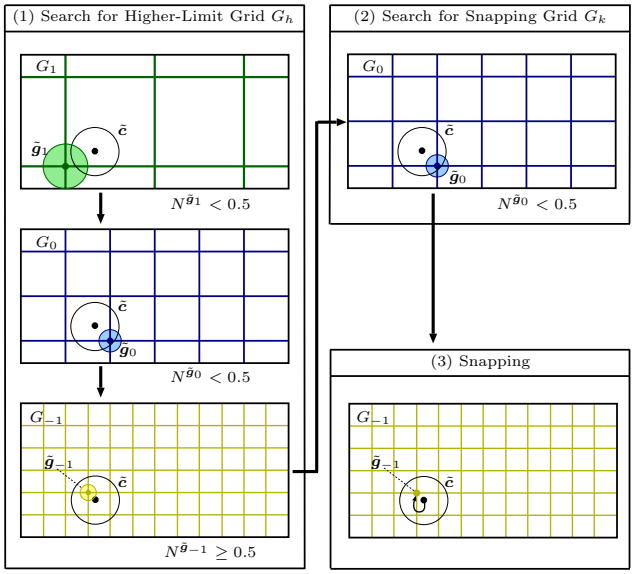


図 4.9: IFGS のプロセス

$\|\boldsymbol{g}_i - \boldsymbol{c}\| \leq \|\boldsymbol{g}_j - \boldsymbol{c}\|$ および $r_{\boldsymbol{g}_i} < r_{\boldsymbol{g}_j}$ であることを考慮すると，式 (4.2) の必然性値 $N^{\tilde{g}_i}$ は i に関して単調非増加であることがわかる．したがって，その否定 $1 - N^{\tilde{g}_i}$ は単調非減少である．結果として，表 4.2 のそれぞれのグレード $\mu(\tilde{g}_i)$ は，無限の個数の項で表現されているが単に以下の式で計算することができる．

$$\mu(\tilde{\boldsymbol{g}}_i) = (1 - N^{\tilde{\boldsymbol{g}}_{i+1}}) \wedge N^{\tilde{\boldsymbol{g}}_i} \tag{4.4}$$

さらに，表 4.2 のルールの個数は無限であるが，式 (4.4) で与えられる $\mu(\tilde{g}_i)$ は図 4.8 に示すように単峰性を持ってシフトするため，有限の回数の手順でスナッピング先 $\boldsymbol{g}_k$ を決定できることがわかる．以下では図 4.9 に示す例を使用しながら，IFGS の処理がスナッピング先 $\boldsymbol{g}_k$ を探索する方法を示す．

(1) **上限グリッド G_h の探索：** ベースグリッド G^* $(= G_1)$ から開始して，次の手順に従って解像

度を上げながら，$\mu(\tilde{\boldsymbol{g}}_h) \geq \mu(\tilde{\boldsymbol{g}}_{h-1})$ を満たすようなグリッド G_h を探索する．図 4.9 の場合，G_h は G_{-1} として得られる．

(1-1) **初期化：** $i := 1$ と初期化する．

(1-2) **必然性値の評価：** ファジィ特徴点の位置 $\boldsymbol{c}$ にもっとも近い G_i のグリッド上の点を選択し，$\boldsymbol{g}_i$ という名前を付ける．次に，スナッピング候補 $\tilde{\boldsymbol{g}}_i = \langle \boldsymbol{g}_i, r_{\boldsymbol{g}_i} \rangle = \langle \boldsymbol{g}_i, r_{G_i} \rangle$ を生成し，必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ を式 (4.2) で評価する．

(1-3) **上限の検出：** $N^{\tilde{\boldsymbol{g}}_i} \geq 0.5$ の場合，A で導出される式 (A.1) の性質に基づいて $\mu(\tilde{\boldsymbol{g}}_i) \geq \mu(\tilde{\boldsymbol{g}}_{i-1})$ が満たされていると判断し，上限グリッドのインデックス h を $h := i$ と設定してステップ（2）に進む．それ以外の場合，$i := i - 1$ と設定した後，ステップ（1-2）に戻る．

(2) **スナッピンググリッド G_k の探索：** 上限グリッド G_h から開始して，以下の手順に従って解像度を下げながら，$\mu(\tilde{\boldsymbol{g}}_k)$ がもっとも高くなるグリッド G_k を探索する．図 4.9 の場合，G_k は G_{-1} として得られる．

(2-1) **初期化：** $i := h + 1$ と初期化する．

(2-2) **必然性値の評価：** ステップ (1-2) と同様に，G_i 上に $\tilde{\boldsymbol{g}}_i$ を生成し，$N^{\tilde{\boldsymbol{g}}_i}$ を評価する．

(2-3) **最大グレードの検出：** $N^{\tilde{\boldsymbol{g}}_i} < 0.5$ の場合，A の式 (A.2) の性質に基づいて $\mu(\tilde{\boldsymbol{g}}_i) \leq \mu(\tilde{\boldsymbol{g}}_{i-1})$ が満たされていると判断し，スナッピンググリッドのインデックス k を $k := i - 1$ と設定してステップ (3) に進む．それ以外の場合，$i := i + 1$ と設定した後，ステップ (2-2) に戻る．

(3) **スナッピング：** ファジィ特徴点 $\tilde{\boldsymbol{c}}$ をグリッド点 $\boldsymbol{g}_k$ にスナッピングする．グリッド点 $\boldsymbol{g}_k$ はスナッピンググリッド G_k から生成されたスナッピング候補 $\tilde{\boldsymbol{g}}_k$ の位置ベクトルである．

上記のアルゴリズムにおいて，ファジィ特徴点 $\tilde{\boldsymbol{c}}$ をグリッド G_k にスナッピングする場合，必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ の参照回数 C_{G_k} は $C^{(1)}_{G_k} + C^{(2)}_{G_k}$ となる．ただし，$C^{(1)}_{G_k}$ と $C^{(2)}_{G_k}$ はそれぞれステップ (1)，ステップ (2) での参照回数であり，以下で与えられる．

$$C^{(1)}_{G_k} = \begin{cases} 2 - k & (k < 1) \\ 1 & (k \geq 1) \end{cases} \tag{4.5}$$

$$C^{(2)}_{G_k} = \begin{cases} 1 & (k < 1) \\ k & (k \geq 1) \end{cases} \tag{4.6}$$

ここで，$C^{(1)}_{G_k}$ は $C^{(2)}_{G_k}$ は Gk ステップ (2) による参照回数を表している．結果として C_{G_k} は以下の式で与えられることとなる．

$$C_{G_k} = |k - 1| + 2. \tag{4.7}$$

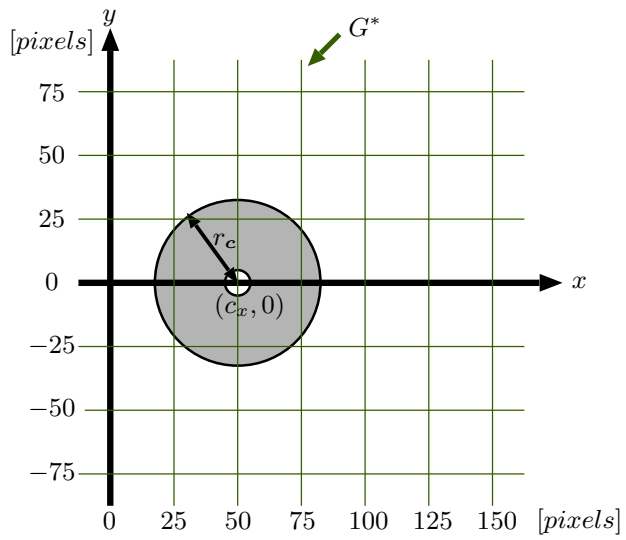


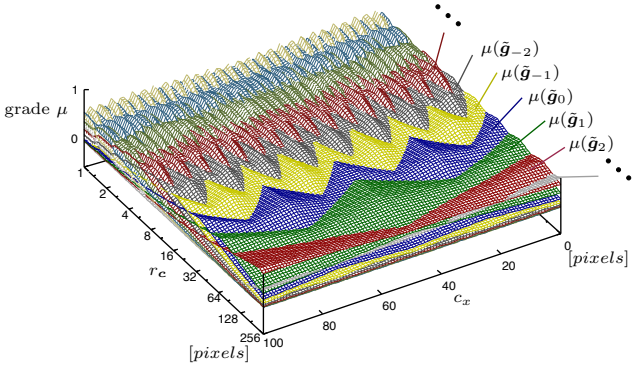
図 4.10: IFG 上のファジィ点 $\tilde{\boldsymbol{c}}$

4.3.3 IFGS のスナッピング特性

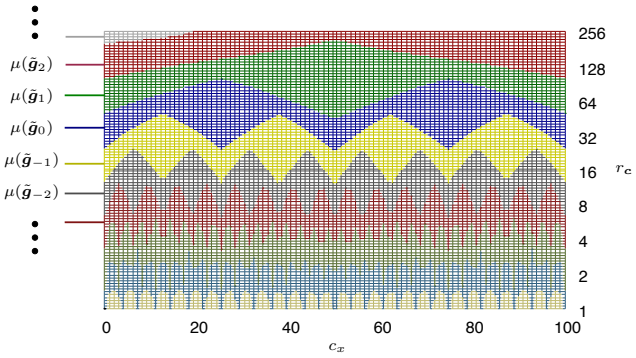
IFGS のスナッピング特性を明らかにするために，図 4.10 に示すような $(S_{G^*}, r_{G^*}) = (25.00, 6.25)$ ピクセルおよび $f = 2$ である IFG 上のファジィ特徴点 $\tilde{\boldsymbol{c}} = \langle (c_x, 0), r_{\boldsymbol{c}} \rangle$ について考えてみる．$c_x \in [0, 100]$，$r_{\boldsymbol{c}} \in [1, 256]$ の範囲で $\tilde{\boldsymbol{c}}$ を変えながら，IFGS により評価したグレード $\mu(\tilde{\boldsymbol{g}}_i)$ をプロットする．その結果を図 4.11 に示す．IFGS が $\tilde{\boldsymbol{c}}$ をグレードがもっとも高くなるグリッドへスナッピングすることに注意すると図 4.11 は IFGS がより大きいファジネスを持つファジィ点をより低い解像度のグリッドにスナッピングし，逆もまた然りであることを示す．このような IFGS の特徴は，[31, Fig. 6] で示されている MFGS の特徴の拡張として考えることができる．

4.4 MFGS と IFGS の比較実験

IFGS がどのように MFGS の問題点を克服したかを示すために，手書き CAD システムで幾何オブジェクトを作図する実験を実施し，MFGS と IFGS を使った結果を比較した．この目的のために，MFGS を用いた手書き CAD システム [38] に IFGS に基づくオブジェクトスナッピング機能を追加した．これにより，MFGS と IFGS の切り替えが可能な手書き CAD システムを構築した．本実験では，線分，円，円弧の組み合わせで構成される幾何図形に焦点を当てた．これらの特徴点は，図 4.12 に示すように，ファジィ線分とファジィ円弧の場合は始点 $\tilde{\boldsymbol{s}}$ と終点 $\tilde{\boldsymbol{e}}$，ファジィ円の場合は始点 $\tilde{\boldsymbol{s}}$ と中心 $\tilde{\boldsymbol{o}}$ として定義した．ペン入力デバイスは，解像度 3.6 ピクセル/mm の液晶タブレットで，3.2 GHz の Intel Core i5-3470 プロセッサと 4 GB のメインメモリを搭載したコンピュータに接続されていた．



(a) 投影図



(b) 上面図

図 4.11: 図 4.10 に示されるファジィ点 $\tilde{\boldsymbol{c}}$ に対して IFGS によって評価されたグレード

4.4.1 パフォーマンスの比較：作図時間

MFGS と IFGS の性能を比較するために，ある目標図形を作図するのに必要な時間を測定する実験を行った[*5]．目標図形として，12 個の幾何オブジェクト，すなわち，大きさの異なる線分，円，円弧で構成された図 4.13(a) に示すコンパスの形状を用意した．図 4.14 に示すように，この作図では，上部のループを形成する一対の小さな円弧を 20 ピクセルのオフセット間隔で揃えるために，20 ピクセル以下の高解像度のグリッドが必要である．したがって，MFG と IFG の設定を以下のように用意した．ただし，これらの原点はディスプレイの中央に設定した．

MFG-A 低解像度 MFG 設定

- $n = 3$
- $(S_{G_1}, r_{G_1}) = (20.0, 5.0)$ ピクセル

[*5] なお，1 ストロークの応答時間（一つの手書きストロークが描画されてからスナッピングされた幾何オブジェクトが表示されるまでの時間）が実験期間中常に 0.5 s 以下であったことから，手書き CAD システムはリアルタイムインターフェースとみなすことができることに注意する．

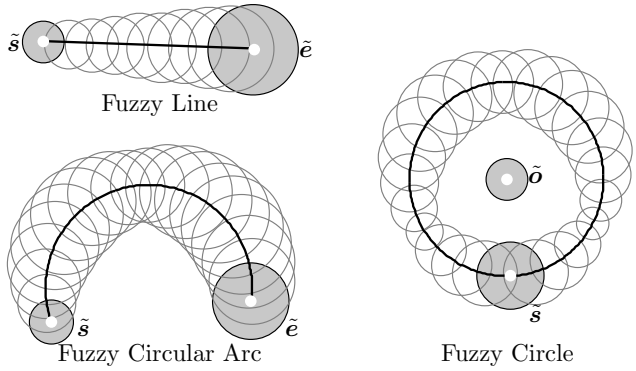


図 4.12: ファジィ幾何プリミティブのファジィ特徴点

- $(S_{G_2}, r_{G_2}) = (40.0, 10.0)$ ピクセル
- $(S_{G_3}, r_{G_3}) = (80.0, 20.0)$ ピクセル

MFG-B 高解像度 MFG 設定

- $n = 3$
- $(S_{G_1}, r_{G_1}) = (5.0, 1.25)$ ピクセル
- $(S_{G_2}, r_{G_2}) = (10.0, 2.5)$ ピクセル
- $(S_{G_3}, r_{G_3}) = (20.0, 5.0)$ ピクセル

IFG-A 低解像度 IFG 設定

- $(S_{G^*}, r_{G^*}) = (20.0, 5.0)$ ピクセル
- $f = 2$

IFG-B 高解像度 IFG 設定

- $(S_{G^*}, r_{G^*}) = (5.0, 1.25)$ ピクセル
- $f = 2$

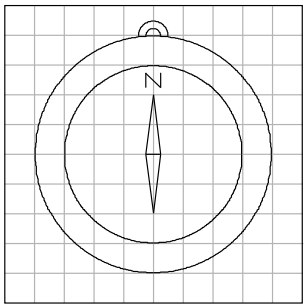
作図時間は次のように測定した.

(1) 手書き CAD システムでは, 液晶タブレット上に目標図形と表示グリッドを表示する．MFG-A または IFG-A の場合は 2 層の表示グリッドが図 4.14(a) のように表示され，MFG-B または IFG-B の場合は図 4.14(b) のように表示される.

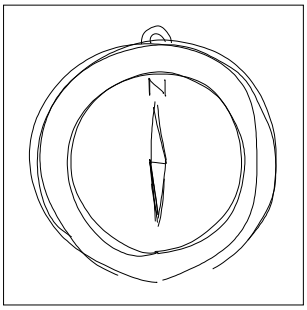
(2) ユーザが作図を開始すると，このシステムは最初に描画された手書きストロークの開始時刻を記録する.

(3) ユーザは（図 4.13(b) のように）手書きストロークを繰り返し描画することで[*6]，このシステムに幾何オブジェクトを（図 4.13(c) のように）同定させ，これらをグリッドにスナッピングさせながら，目標図形を完成させていく.

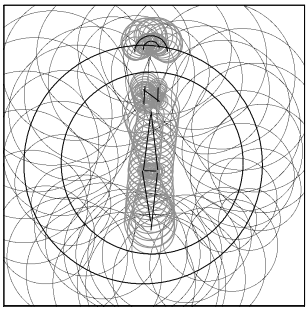
[*6] いくつかの手書きストロークが互いに重なっている．これは，[38] の手書き CAD システムでは [25] で提案されたアルゴリズムが実装されており，ユーザが同定された結果を修正するために手書きストロークを重ねて描画することができるためである.



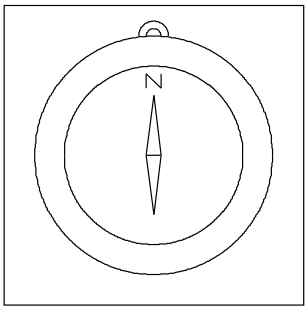
(a) グリッド間隔が 80 ピクセルであるグリッド上の目標図形



(b) ユーザによって描画された手書きストロークの例



(c) 手書きストロークから同定されたファジィ幾何プリミティブ



(d) ファジィ幾何プリミティブからスナッピングされた幾何オブジェクト

図 4.13: 目標図形「コンパス」とこれを完成させるための手書きストロークの例

(4) 作図を監視しているスーパーバイザは，図 4.13(d) のように目標図形が完成した時点でユーザの作図を終了させる．

(5) 最初の手書きストロークの開始時刻から最後の手書きストロークの終了時刻までの時間を計算することで，作図時間を求める．

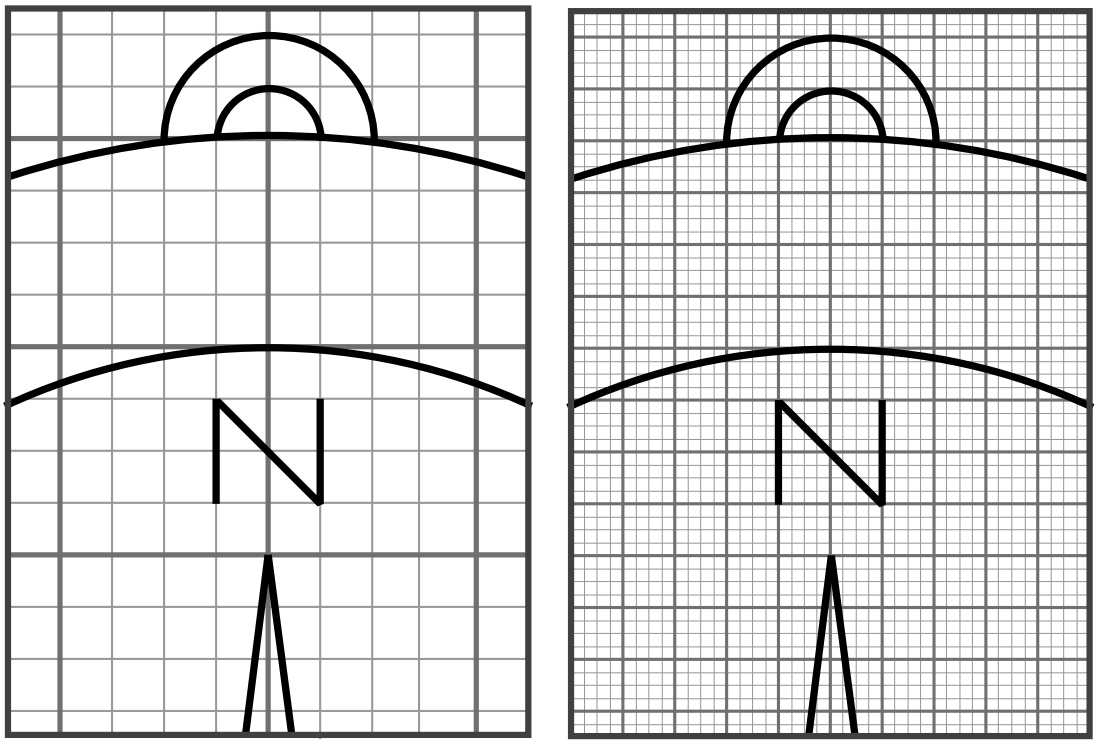
それぞれのユーザに対して，グリッドの設定を変えながら，繰り返し作図時間を測定した．その全体的な手順は以下の通りであった．

(1) ユーザは，4 つのグリッド設定のそれぞれで目標図形を作図するタスクを 2 回完了させる練習をした．

(2) IFG-A，MFG-A，IFG-B，MFG-B のそれぞれに対して，作図時間を 10 回測定し，平均作図時間を算出した．

それぞれのユーザが上記の作図タスクを完了するのには約 40 分を要した．

図 4.15 は，手書き CAD システム [38] に精通した 17 人のユーザから得られた平均作図時間を示している．個人差はあるものの，全ユーザの結果を総合すると MFG-B の平均作図時間が他の 3



(a) グリッド間隔が 20.0 ピクセルと 80.0 ピクセルの 2 層の表示グリッド

(b) グリッド間隔が 5.0 ピクセルと 20.0 ピクセルの 2 層の表示グリッド

図 4.14: ディスプレイに表示グリッドとともに表示される目標図形「コンパス」

つのグリッド設定に比べて著しく長いことが明らかとなった．したがって，MFG-B は目標図形には不適切な設定であると言える．実際，MFG-B を使った描画タスクでは，コンパスの本体となる直径 640 ピクセルの大きな円の配置に成功するために，何度も手書きストロークを描画しなければならないケースがよく見受けられたが，これは MFG-B のもっとも低いグリッド解像度（20 ピクセル）では，このサイズの円にはまだ高すぎるからである．対照的に，MFG-A では，もっとも解像度の低い 80 ピクセルのグリッド間隔がこの円に適しており，もっとも解像度の高い 20 ピクセルのグリッド間隔が，コンパスの上部にあるループを形成する小さな円弧のペアに適していた．それゆえ，ユーザはより少ない手書きストローク数でより早くタスクを完了することができた．IFG-A と IFG-B については，無限解像度のファジィグリッドに MFG-A の多解像度ファジィグリッドが含まれているため，どちらのグリッド[*7]でも MFG-A と同等のスピードでタスクを完了することができた．

これらの結果は，4.2.4 節で述べた MFGS の問題点，すなわち，MFG の設定が描画される幾何オブジェクトの大きさの範囲に適していない場合に，MFGS のパフォーマンスが低下することを

[*7] なお，IFG-A と IFG-B は，IFG の設定や表示グリッドが異なるものの，スナッピング特性は同等であり，平均作図時間もほぼ同等であった．

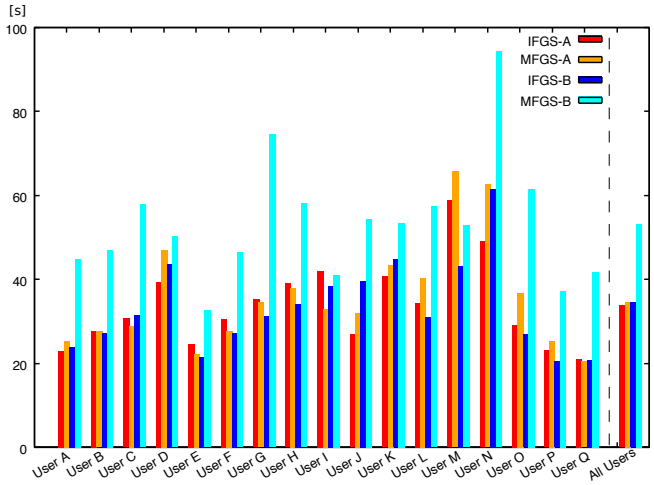


図 4.15: 図 4.13 の目標図形を完成させるために要した平均作図時間

示している．これに対し，IFGS は IFG の設定に関係なく動作するため，この問題を克服することができた．

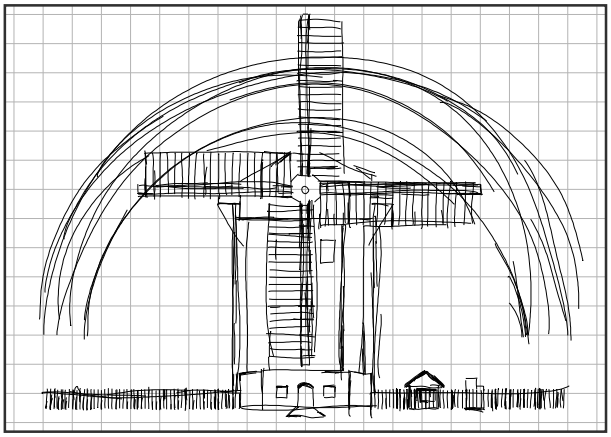
4.4.2 複雑な目標図形を用いたスナッピング特性の比較

次に，前の実験で単純な目標図形を対象とした場合に良好に動作した MFG-A と IFG-A のスナッピング特性を，約 300 個の幾何オブジェクトからなる複雑な図形を作図する状況で比較した．図 4.16 は，ユーザが描画した手書きストロークと，その手書きストロークから同定されたファジィ幾何プリミティブを示している．手書きストロークは，直径が約 1,500 ピクセルの大きな虹の形から，列の間隔が 10 ピクセルの細かいフェンスまでの範囲に及ぶ．このように，図 4.13 の目標図形に比べて，幾何オブジェクトのサイズのダイナミックレンジが非常に広くなっている．

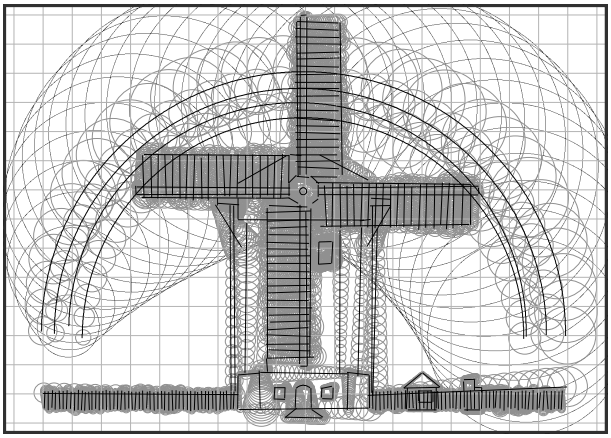
図 4.17(a) は，図 4.16 の入力を MFG-A に用いた場合の結果を示している．この場合，虹を構成する大きな図形が正しく配置されていないことがわかる．これは，もっとも低いグリッド解像度である 80 ピクセルのグリッド間隔では，このような大きな幾何オブジェクトにはまだ細かすぎて微妙なずれが生じたためである．また，フェンスの一部にある細かい図形も位置が合っていないことがわかる．これは，もっとも高いグリッド解像度である 20 ピクセルのグリッド間隔では，このような細かいものを揃えるには粗すぎたためである．このような状況において，MFGS がすべてのオブジェクトを適切にスナッピングするためには，ユーザが手動で多重度 n を増やし，追加のファジィグリッドのプロパティを設定する必要がある．

図 4.17(b) は，IFG-A の出力結果である[*8]．すべての幾何オブジェクトがサイズのダイナミックレンジに関わらず適切に配置されていることがわかる．これは，IFGS が無限の解像度を利用できるからである．この結果から，IFGS のユーザは，幾何オブジェクトのサイズが大きく異なっても

[*8] なお，IFG-B は，IFG-A と実質的に等価であるため，同じ出力が得られることに注意する．



(a) 手書きストローク

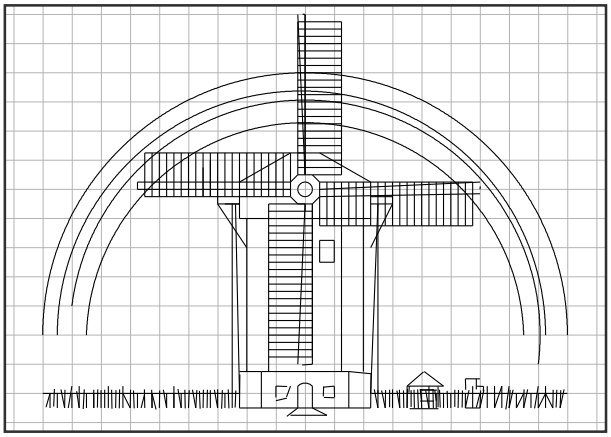


(b) 同定されたファジィ幾何プリミティブ

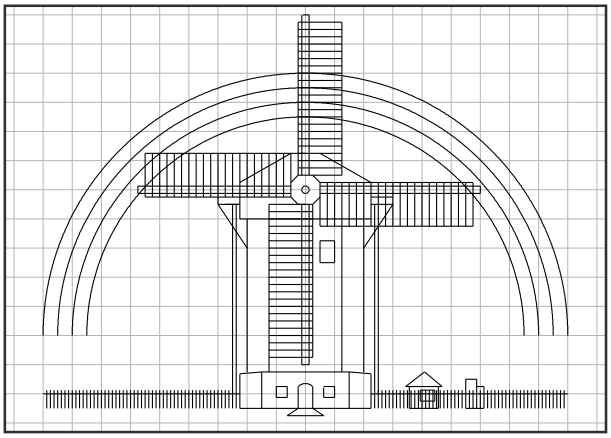
図 4.16: 80 ピクセルのグリッド間隔のグリッド上に示された，複雑な図形を完成させるための手書きストロークと手書きストロークから同定されたファジィ幾何プリミティブ

グリッドを再設定するという面倒な手動操作を回避できることがわかる．

実験ではグリッドの原点をディスプレイの中央に設定していたが，この原点を変更すると IFGS と MFGS の両方で同じ結果が得られないことに注意する．たとえば，4.18 は，原点を変更したとき，IFG-A を用いた IFGS によって与えられる結果の違いを示している．これらの結果では，大きな虹のスナッピングは原点の変更によって影響を受けるが，他の小さなオブジェクトは同じスナッピング位置に保たれている．これは，表示グリッドよりも解像度の低いグリッドのレイアウトが原点の変更によって変化したためである．このことは，ユーザはこのような低解像度のグリッドのレイアウトを意識する必要があることを意味している．IFGS に残された課題は，ユーザが図形を作図するときに，これらのグリッドのレイアウトをどのように認識できるかということである．経験的には，原点をディスプレイの中心に設定することがもっともシンプルで効果的な解決策である．



(a) MFG-A を用いた MFGS により与えられた結果



(b) IFG-A を用いた IFGS により与えられた結果

図 4.17: 図 4.16 に示された入力からスナップされた出力結果

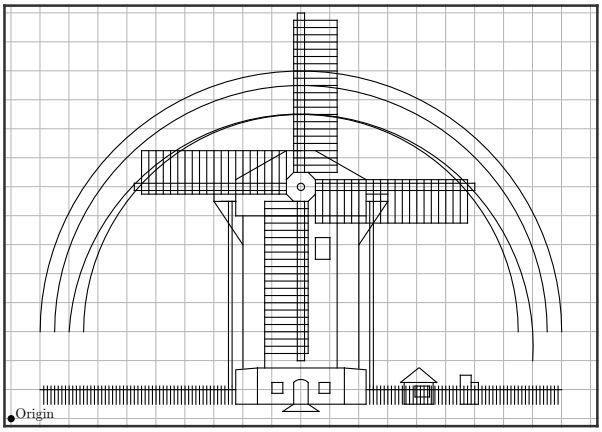
4.4.3 計算効率の比較

MFGS は，多重度 n を増やすことで，IFGS と同じ結果，すなわち図 4.17(b) に示す結果を得ることが可能である．しかしながら，IFGS は MFGS よりも低い計算コストとなる．これを実証するために，IFGS と MFGS の計算効率を，必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ への参照回数の観点で比較する．

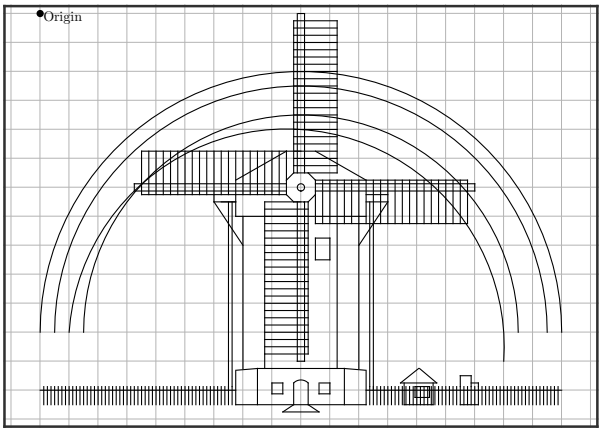
図 4.17(b) に示す結果を得たときのファジィ特徴点の総数は 588 個だった．IFG-A のグリッド G_k にスナッピングした特徴点の頻度は，これを F_{G_k} と表すこととすると，図 4.19 のようになる．MFGS で同じ結果を得るためには，次のような多重度 10 の最小限の MFG 設定が必要である．

MFG-C 図 4.17(b) を得るための最小限の MFG 設定

- $n = 10$
- $(S_{G_i}, r_{G_i}) = (20 \times 2^{i-6}, 5 \times 2^{i-6})$ ピクセル $(i = 1, \ldots, 10)$



(a)



(b)

図 4.18: 異なる原点に対する IFG-A を用いた IFGS により与えられた異なる結果

MFG-C を用いた MFGS の計算コストは，これを $C_{\text{MFG-C}}$ と表記すれば，4.2.3 節にしたがい，$588 \times \left(\frac{n(n+1)}{2} + n\right) = 38,220$ となる．対照的に，IFG-A を用いた IFGS の計算コストは．これを $C_{\text{IFG-A}}$ と表記すれば式 (4.7) にしたがい，$\sum_{k\in\mathbb{Z}} F_{G_k} C_{G_k} = \sum_{k\in\mathbb{Z}} F_{G_k}(|k-1|+2) = 2,041$ となる．したがって，IFG-A の MFG-C に対する相対的な計算コストは $\frac{C_{\text{IFG-A}}}{C_{MFG-C}} = \frac{2,041}{38,220} \doteqdot 0.053$ となる．IFGS では，以下のいずれの IFG 設定でも IFG-A と同等のスナッピング特性が得られる．

IFG-(i) $(i \in \mathbb{Z})$ IFG-A と同等の IFG 設定

- $(S_{G^*}, r_{G^*}) = (20 \times 2^{i-1}, 5 \times 2^{i-1})$ ピクセル
- $f = 2$

しかしながら，これらの計算コストは，これらを $C_{\text{IFG-}(i)}$ と表記すれば，i に応じて $\sum_{k\in\mathbb{Z}} F_{G_k}(|k-i|+2)$ と変化する．IFG-(i) の MFG-C に対する相対的な計算コスト，すなわち $\frac{C_{\text{IFG-}(i)}}{C_{MFG-C}}$ の計算コストを図 4.20 に示す．ここで，MFGS が様々なデバイスの解像度に対応するためには多重度 n

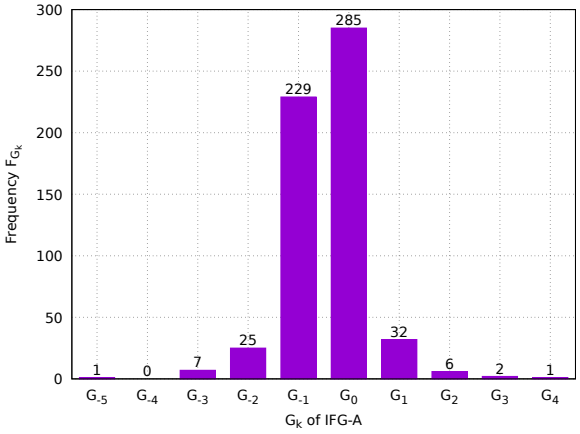


図 4.19: G_k がスナッピンググリッドとして選択される頻度

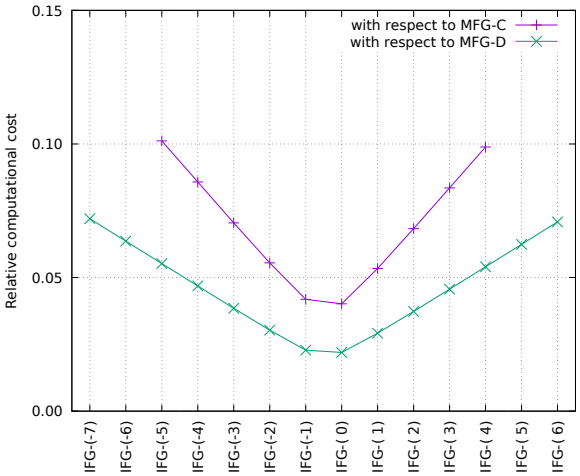


図 4.20: IFGS の MFGS に対する計算コスト

を大きくする必要がある．たとえば，今回の実験で使用した液晶タブレットの 4 倍の解像度を持つ高解像度のタブレットコンピュータと，その液晶タブレットの 4 分の 1 の解像度を持つ低解像度の電子黒板の両方に対応するためには，次のような多重度 14 の MFG 設定が必要となる．

MFG-D さまざまな解像度のデバイスで図 4.17(b) を得るための MFG 設定

- $n = 14$
- $(S_{G_i}, r_{G_i}) = (20 \times 2^{i-8}, 5 \times 2^{i-8})$ ピクセル $(i = 1, \ldots, 14)$

また，IFG-(i) の MFG-D に対する相対的な計算コストすなわち $\frac{C_{\text{IFG-}(i)}}{C_{MFG-D}}$ についても図 4.20 に示す．

以上の結果から，IFGS は，基準グリッドが実際に使用されるスナッピンググリッドの範囲から大きく逸脱しない限り，MFGS に対して低い計算コスト（高い計算効率）となることがわかる．ま

たこれらの結果は，MFGS の多重度 n が大きくなるほど，IFGS の相対的な計算コストが減少することも示している．これらの結果は，描画される幾何オブジェクトが既知であるという最良の状況に基づいている．描画される幾何オブジェクトのサイズが未知である実際の状況では，n をさらに増加させる必要があり，IFGS の相対的な計算コストは減少する．

4.5 本章のまとめ

MFGS において問題となるグリッド設定が必要となってしまうことを克服するために，可能な解像度の数を無限に増やす IFGS という MFGS の拡張を提案した．IFGS のアルゴリズムとして有限のステップ数で完了できるものを説明した．くわえて，MFGS を搭載した手書き CAD システムに IFGS を追加し，IFGS の有効性と効率性について MFGS と比較して検証する実験を行った．実験結果は，MFGS の性能はグリッド間隔によっては劣化するのに対し，IFGS の性能は異なるグリッドの設定であっても高いレベルを維持していることを明確に示した．ユーザが，幾何オブジェクトのサイズが幅広い範囲にある複雑な図形を描画した場合におけるスナッピング特性の比較実験では，IFGS は MFGS と異なり，幾何オブジェクトのサイズが大きく異なる場合でも，グリッド設定を変更するための面倒な手動操作を必要としないことが示された．

第 5 章

ファジィ論理に基づいて複数のオブジェクトを横断したジオメトリ・トポロジ編集を実現するインタラクティブな手書き CAD インターフェース

デジタルペンタブレットやタッチディスプレイの普及により，スケッチベースのインタラクティブな CAD ユーザインターフェースが大きく注目されている．SKIT と呼ばれる既存のインターフェースでは，しかしながら，重ね書きストロークを介して実行される編集操作は一度に一つの幾何オブジェクトしか対象にすることができない．この限界を克服するために，本章では 2 次元 CAD システムで使用するために開発された新しい汎用手書きインターフェースを提案する．提案するインターフェースでは，スケッチベースの編集操作により，重ね書きを介して複数の幾何オブジェクトのジオメトリとトポロジを同時に修正できる．提案インターフェースは，SKIT のファジィ論理ベースの戦略に基づいて開発されている．このインターフェースを使用すると，ユーザは，たとえば，ラフスケッチから始めて，重ね書きの繰り返しを通してプログレッシブに詳細なデザインを作成するような，クリエイティブな作図を行うことができる．実施される実験の結果では，熟練者が提案されたインタラクティブな作図インターフェースを制御でき，初心者がそれを制御することに慣れることができることが示される．

5.1 本章の目的

ペンタブレットやタッチディスプレイの普及に伴い，ユーザがスケッチによりインタラクティブに作図することができる手書きインターフェースの研究が活発に行われている．この研究の一部

が一連のファジィ論理ベースの研究 [22, 24, 23, 25, 26, 27] であり，ユーザが，7 つのクラス（線分（L），円（C），円弧（CA），楕円（E），楕円円弧（EA），閉自由曲線（FC），および開自由曲線（FO））の幾何プリミティブの列として表現された幾何オブジェクトを入力および編集することができる汎用手書き CAD インターフェースを実現した．ここでは，佐賀ら [24, 23] が初めて，描画のあいまいさをストロークデータに追加することにより，手書きストロークをファジィスプライン曲線（FSC）として扱う手法を提案した．その後，彼らは FSC の形状とあいまいさに基づいて FSC を幾何プリミティブからなる幾何オブジェクトとして同定するための FSC 同定法 [22, 23] として知られる手法を提案した．続いて，渡辺ら [27] は同定された幾何オブジェクトの特徴点をグリッドシステムにスナッピングする無限解像度ファジィグリッドスナッピング（IFGS）と呼ばれる手法を提案した．これにより，ユーザが幾何オブジェクトを自由に入力および位置合わせできる手書きインターフェースを実現した．さらに，佐藤ら [25] は S-FSCG を提案し，重ね書き[*1] によって FSC を修正し，ユーザが既存の幾何オブジェクトを編集できる手書きインターフェースを実現した．上記の研究により，以下の 5 クラスの操作を可能にするスケッチベースのインターフェースが作成された[*2]．

1. **幾何オブジェクトの入力：** 手書きストロークを描画することによって実行される入力操作であり，これにより，新しい幾何オブジェクトが作成される．
2. **幾何プリミティブの変形：** 重ね書きストロークを描画することによって実行されるジオメトリ編集操作であり，これにより，1 つの幾何オブジェクト内の 1 つの幾何プリミティブの形状が修正される．
3. **幾何プリミティブの分割：** 重ね書きストロークを描画することによって実行されるトポロジ編集操作であり，これにより，1 つの幾何オブジェクト内で 1 つの幾何プリミティブが 2 つの幾何プリミティブに分割される．
4. **幾何プリミティブの統合：** 重ね書きストロークを描画することによって実行されるトポロジ編集操作であり，これにより，1 つの幾何オブジェクト内で 2 つの幾何プリミティブが 1 つの幾何プリミティブに統合される．
5. **1 から 4 の組み合わせ：** 重ね書きストロークを描画することによって実行される操作 1 から 4 の組み合わせであり，これにより，1 つの幾何オブジェクト内の複数の幾何プリミティブのジオメトリやトポロジが修正される．

このインターフェースにより，たとえば，ユーザは図 1.1(a) の手書きストロークを描画することにより図 1.1(b) に示されている幾何オブジェクトを入力できる．その後，ユーザは図 1.1(b) の幾何オブジェクトを重ね書きによって編集することで図 1.1(c) のように修正することができる．しかしながら，上記の編集操作では，それぞれの重ね書きストロークが一度に 1 つの幾何オブジェクトのみを対象としており，複数の幾何オブジェクトを横断して図 1.1(c) の幾何オブジェクトを図

[*1] 本章では，「重ね書き」とは既存のオブジェクトの上に新たに手書きストロークを描画することを意味する．
[*2] これらの操作の典型例を図 5.22 に示す．

1.1(d) の幾何オブジェクトに修正することはできない．そのため，ユーザが従来のペンと紙の作図と同じように，ラフスケッチから始めて詳細な形状を段階的に決定することで創造的に作図することは困難である．

上記の研究を基にして，上の操作に加えて以下の 3 タイプの操作を可能にする新しい手書きインターフェースを提案し，複数の幾何オブジェクトを横断する形状と位相の編集を実現する．

6. **幾何オブジェクトの分解：** 重ね書きストロークを描画することによって実行されるトポロジ編集操作であり，これにより，1 つの幾何オブジェクトは 2 つの幾何オブジェクトに分解される．
7. **幾何オブジェクトの連結：** 重ね書きストロークを描画することによって実行されるトポロジ編集操作であり，これにより，2 つの幾何オブジェクトは 1 つの幾何オブジェクトに連結される．
8. **1 から 7 の組み合わせ：** 重ね書きストロークを描画することによって実行される操作 1 から 4 の組み合わせであり，これにより，複数の幾何オブジェクト内の複数の幾 何プリミティブのジオメトリやトポロジが修正される．

本章では，提案インターフェースに実装されたプロセスの詳細と実験結果を示す．まず，FSC グラフと呼ばれるデータ構造を導入する．これは，複数の幾何オブジェクトの幾何形状と幾何プリミティブ間の接続関係を保持する．その後，FSC グラフを基盤としてファジィ論理ベースの上記研究で開発された要素技術を統合することにより，複数の幾何オブジェクトに対する手書き編集操作を実現する新しいプロセスを提案する．続いて，提案されたプロセスを実装した作図インターフェースを使用して実施されたいくつかの実験について説明する．これらの実験は，ユーザが，このインターフェースを使用して，複数の幾何オブジェクトに対して操作 1 から 8 までをインタラクティブに実行しながら，自在に作図できることを示すために実施されたものである．

5.2 個別の幾何オブジェクトに対する既存の手書き編集操作

この節ではまず個別の幾何オブジェクトに対して動作する既存の手書き編集操作を要約する．これは，次節で提案する操作の基礎を形成するものであり，その要素技術である FSC 生成法，FSC 融合法，FSC フラグメンテーション法，FSC 同定法，IFGS は，提案する操作でも使用される．既存の手書き編集操作のプロセスを図 5.1 に示す．新しい重ね書きストロークが入力されるたびにこのプロセスがリアルタイムで繰り返される．これによりユーザは，図 5.2 に示すように，幾何オブジェクトをインタラクティブに編集しながら作図を進めることができる．

5.2.1 FSC 生成法

[24, 23] で提案されている FSC 生成法は，描画動作の雑さに応じて，新たに描画された手書きストロークに位置情報のファジネス（またはあいまいさ）を追加することによって FSC を生成

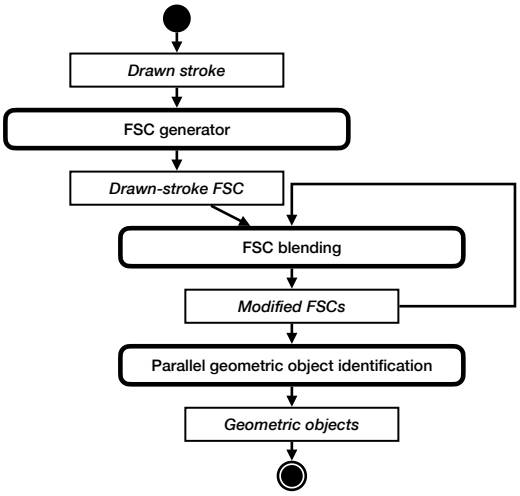
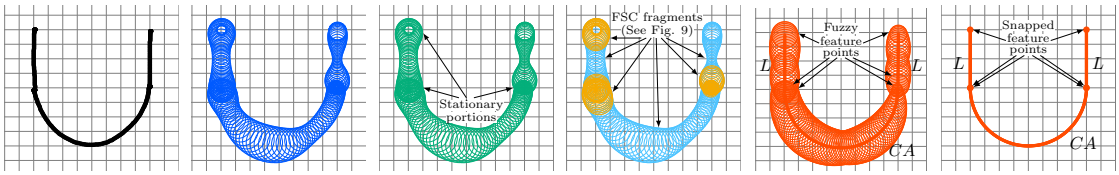
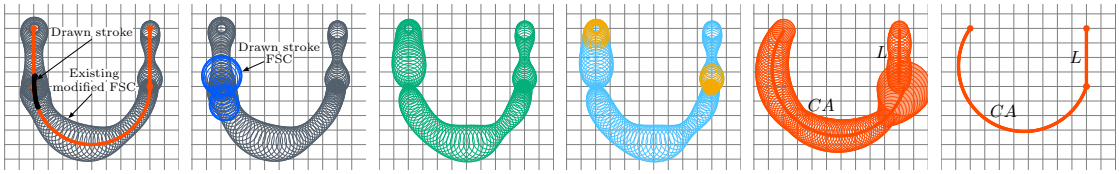


図 5.1: 個別の幾何オブジェクトに対する既存の手書き編集操作のプロセス



(a) 手書きストローク (b) 手書きストロークに対する FSC (c) 修正 FSC (d) FSC フラグメント列 (e) ファジィ幾何プリミティブ ェクト列 (f) 幾何オブジ



(g) 手書きストローク (h) 手書きストロークに対する FSC (i) 修正 FSC (j) FSC フラグメント列 (k) ファジィ幾何プリミティブ ェクト列 (l) 幾何オブジ

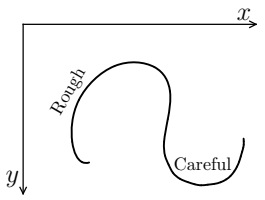
図 5.2: 単一の幾何オブジェクトに対する既存の操作の例

する．ここで，FSC は時刻 $t \in [\alpha_{\tilde{\boldsymbol{s}}}, \beta_{\tilde{\boldsymbol{s}}}]$ を 2D の円錐型ファジィ点 $\tilde{\boldsymbol{s}}(t)$ に対応させるような写像 $\tilde{\boldsymbol{s}} : [\alpha_{\tilde{\boldsymbol{s}}}, \beta_{\tilde{\boldsymbol{s}}}] \to P$ として表される．ここで，$\alpha_{\tilde{\boldsymbol{s}}}$ および $\beta_{\tilde{\boldsymbol{s}}}$ はそれぞれ手書きストロークの開始時刻と終了時刻であり，P はすべての 2D の円錐型ファジィ点の集合である．$\tilde{\boldsymbol{p}} = \langle \boldsymbol{p}, r_{\tilde{\boldsymbol{p}}} \rangle$ と表されるある 2D の円錐型ファジィ点は，以下の円錐型メンバシップ関数によって特徴付けられる，図 5.3 に示されるようなファジィ集合である．

$$\mu_{\tilde{\boldsymbol{p}}}(\boldsymbol{w}) = \max\left\{1 - \frac{\|\boldsymbol{w} - \boldsymbol{p}\|}{r_{\tilde{\boldsymbol{p}}}}, 0\right\} \tag{5.1}$$

(a) 投影図　　(b) 上面図

図 5.3: 円錐型ファジィ点 $\tilde{\boldsymbol{p}} = \langle \boldsymbol{p}, r_{\tilde{\boldsymbol{p}}} \rangle$ のメンバシップ関数



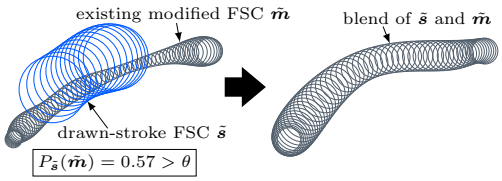
(a) 手書きストローク

(b) FSC の投影図　　(c) FSC の上面図

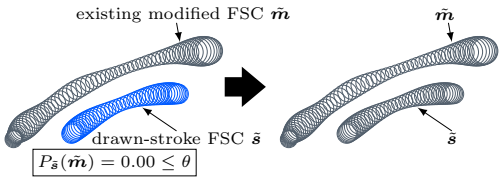
図 5.4: 手書きストロークから生成された，円錐型ファジィ点の軌跡を描く FSC

ここで，$\boldsymbol{w}$ は 2 次元変数位置ベクトル，$\boldsymbol{p}$ は $\tilde{\boldsymbol{p}}$ の 2 次元位置ベクトル，$r_{\tilde{\boldsymbol{p}}}$ は $\tilde{\boldsymbol{p}}$ のファジネスである．ここで，$r_{\tilde{\boldsymbol{p}}}$ は，手書きストロークの丁寧に描画された部分を小さなファジネス（細かいポジショニング）に関連付け，雑に描画された部分を大きなファジネス（粗いポジショニング）に関連付けるファジネス生成モデル [35] を使用して生成される．その結果，図 5.4 に示されているように，FSC $\tilde{\boldsymbol{s}}$ は時刻 $t \in [\alpha_{\tilde{\boldsymbol{s}}}, \beta_{\tilde{\boldsymbol{s}}}]$ が変化するにつれて円錐型ファジィ点の軌跡を描く．FSC 生成法の詳細は，[24] の 2 節および [23] の 2 節で説明されている．

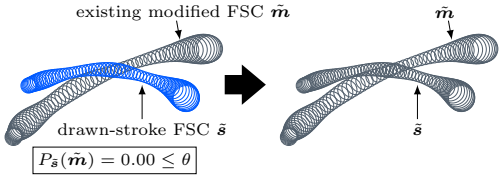
たとえば，図 5.2 に示すように，(a) の手書きストロークを描画すると，(b) の手書きストローク FSC が生成される．(g) に示すように，既存の図形の上に手書きストロークを描画した場合でも，この段階では (h) のように手書きストローク FSC のみが生成される．



(a) $\tilde{s}$ と $\tilde{m}$ が融合されるケース



(b) $\tilde{s}$ と $\tilde{m}$ が融合されないケース



(c) $\tilde{s}$ と $\tilde{m}$ が融合されないもう一つのケース

図 5.5: θ が 0.5 である時の FSC 融合法の例

5.2.2 FSC 融合法

佐藤ら [25] によって提案された FSC 融合法は，前のイテレーションで得られた既存の修正 FSC から $P_{\tilde{s}}(\tilde{m})$ がもっとも高くなるような FSC $\tilde{m}$ を 1 つ決定する．ここで，$P_{\tilde{s}}(\tilde{m})$ は，図 5.5 に示すように，$\tilde{m}$ が手書きストローク FSC $\tilde{s}$ と重なっている可能性の程度を表し，ファジィ理論 [36] の可能性測度に基づいて $[0, 1]$ の範囲の実数として計算される．$P_{\tilde{s}}(\tilde{m})$ がしきい値 $\theta \in [0, 1]$ を超える場合，FSC 融合法は $\tilde{m}$ を融合対象として扱って $\tilde{s}$ と融合させることによって新たに修正 FSC を生成し，その後，図 5.5(a) に示すように $\tilde{m}$ を新たな修正 FSC に置き換える．それ以外の場合，FSC 融合法では，図 5.5(b)，(c) に示すように，$\tilde{s}$ を修正 FSC の新しい要素として追加する．既存の修正 FSC が存在しない初期条件下では，FSC 融合法は，$\tilde{s}$ を修正 FSC の最初の要素とする．FSC 融合法の詳細は，[25] の 3 節に記載されている．

たとえば，図 5.2(b) に示すように手書きストローク FSC の融合対象が存在しない場合，手書きストローク FSC は (c) に示すように，修正 FSC の最初の要素として単に追加される．一方，(h) に示すように手書きストローク FSC の融合対象が存在する場合，(c) の既存の修正 FSC は (i) に示されている新たな修正 FSC に置き換えられる．

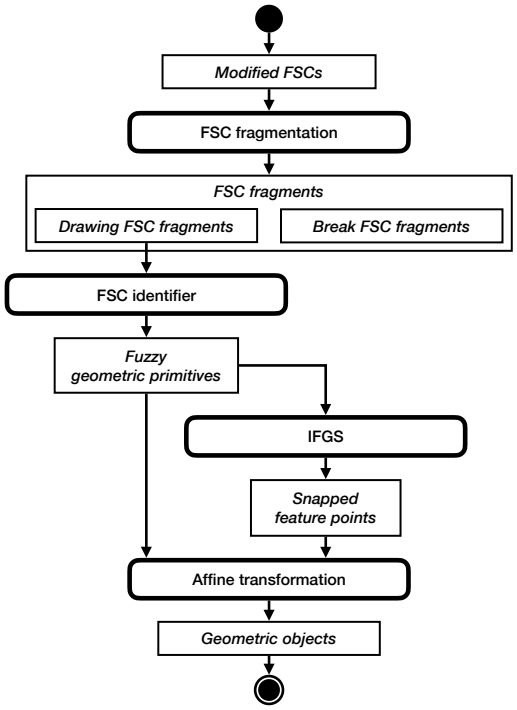


図 5.6: 並列的な幾何オブジェクト同定のプロセス

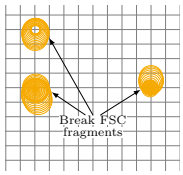
5.2.3 並列的な幾何オブジェクト同定

修正 FSC のそれぞれを，図 5.6 に示されているプロセスに従って並列的に幾何オブジェクトとして同定する．このプロセスのそれぞれの部分を以下で説明する．

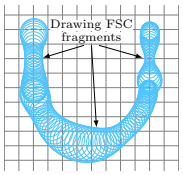
5.2.3.1 FSC フラグメンテーション法

FSC フラグメンテーション法 [26] は，FSC の停止部分（すなわち，元の手書きストロークの停止動作）を検出することにより，修正 FSC を FSC フラグメントの列に分割する．この場合，それぞれの FSC フラグメントは，部分 FSC とラベルで構成されるペアとして表現される．部分 FSC は，元の FSC の一部を切り取ったものである．部分 FSC が停止部分として分類される場合，ラベルは Break として設定され，それ以外の場合，ラベルは Drawing として設定される．ブレイク FSC フラグメント（ラベルが Break である FSC フラグメント）とドローイング FSC フラグメント（ラベルが Drawing である FSC フラグメント）は，FSC フラグメント列の中で交互に配置されることに注意する．FSC フラグメンテーション法の詳細は，[26] の 3 節で説明されている．

たとえば，図 5.2 に示すように，(c) の修正 FSC は，(d) に示すように 6 つの FSC フラグメントの列に分割される．この列は，図 5.7(a) に示されている 3 つのブレイク FSC フラグメントと図 5.7(b) に示されている 3 つのドローイング FSC フラグメントで構成されている．



(a) ブレイク FSC フラグメント



(b) ドローイング FSC フラグメント

図 5.7: 図 5.2(d) の FSC フラグメント列

5.2.3.2 FSC 同定法

[22, 23] で提案されている FSC 同定法は，ドローイング FSC フラグメントから描画軌跡として部分 FSC を抽出することにより，それぞれの描画軌跡を 7 クラスの幾何プリミティブ（L，C，CA，E，EA，FC，FO）のいずれかとして同定し，ファジィ幾何プリミティブ列を生成する．ここで，FSC 同定法は，図 1.2 に示されている包含関係を考慮して，ファジネスが大きい FSC をより単純な幾何プリミティブとして推論し，ファジネスが小さい FSC をより複雑な幾何プリミティブとして推論しようとする傾向のある戦略を採用している．FSC 同定法の詳細は，[22] の 3 節および [23] の 3 節に記載されている．

たとえば，図 5.2 に示すように，(d) の FSC フラグメント列の中のドローイング FSC フラグメントからは，(e) に示すように 3 つのファジィ幾何プリミティブ L，CA，L で構成されたファジィ幾何プリミティブ列が生成される．

5.2.3.3 IFGS

IFGS[27] は，すべてのファジィ幾何プリミティブのそれぞれファジィ特徴点を，そのファジネスを考慮しながらグリッドシステムにスナッピングする．ここで，IFGS は，ファジネスが大きいファジィ点を低解像度グリッドにスナッピングし，ファジネスが小さいファジィ点を高解像度グリッドにスナッピングしようとする傾向のある戦略を採用している．IFGS の詳細は，[27] の 3 節に記載されている．

5.2.3.4 アフィン変換

ファジィ特徴点のスナッピング結果に従ってアフィン変換を行うことで，それぞれのファジィ幾何プリミティブを配置し，幾何オブジェクトを生成する．

たとえば，図 5.2 に示すように，(e) のファジィ幾何プリミティブ列からは，L，CA，L で構成される (f) の幾何オブジェクトが生成される．

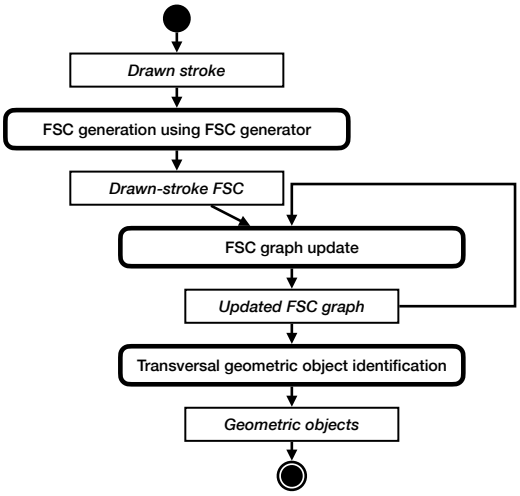


図 5.8: 複数の幾何オブジェクトに対する提案手書き編集操作のプロセス

5.3 複数の幾何オブジェクトに対する手書き編集操作の提案

前節で概説した既存の手書き編集操作は，一度に 1 つの幾何オブジェクトのみを対象とし，一度に 1 つの FSC を修正する重ね書きによって実現される．複数の幾何オブジェクトを横断的に対象にする手書き編集操作を実現するには，複数の FSC とそれらのトポロジカルな関係を同時に修正する重ね書きを実装する必要がある．この節ではまず，FSC とそれらのトポロジを柔軟に表現することが可能な FSC グラフと呼ばれるデータ構造を導入する．その後，図 5.8 に示すような複数の幾何オブジェクトに対する手書き編集操作を実現するための新しいプロセスを提案する．このプロセスがリアルタイムで繰り返されることとなれば，ユーザは図 5.9 に示すように，新しい重ね書きストロークを入力するたびに複数の幾何オブジェクトを横断的に編集しながら，インタラクティブに作図を進めることができる． FSC グラフと提案するプロセスの 3 つの部分（FSC 生成法を使用した FSC 生成，FSC グラフの更新，横断的な幾何オブジェクト同定）について以下に説明する．

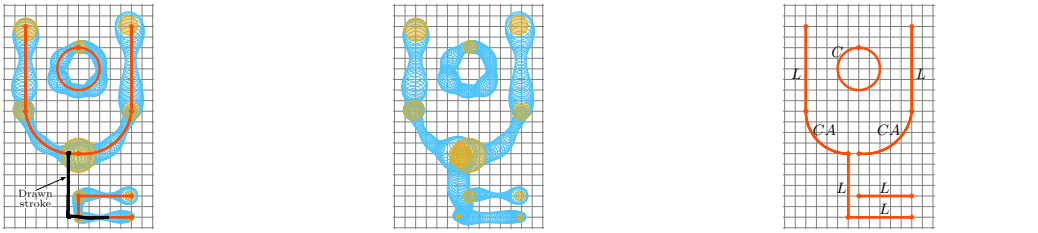
5.3.1 FSC グラフの定義

FSC グラフとそれに関連する用語を以下に定義する．

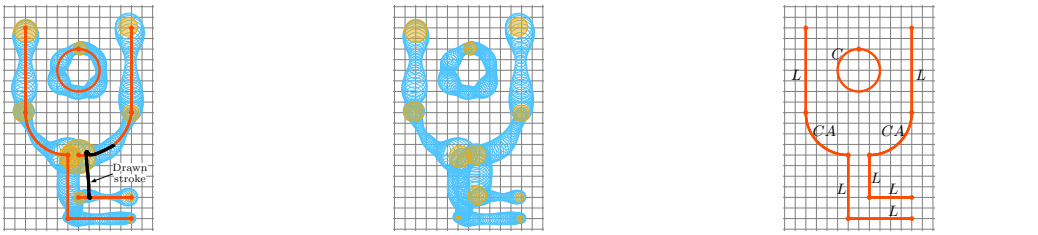
FSC グラフ： FSC グラフは，ドローイングノードとコネクタノードからなるノード集合で構成される有向グラフであり，すべての連結成分が FSC パスグラフである．

ドローイングノード： ドローイングノードは，描画軌跡に関する要素情報を表す．これは，ドローイング FSC フラグメントの部分 FSC として実体化される．

コネクタノード： コネクタノードは，接続点に関する要素情報を表す．これは，ブレーク FSC フラグメントの部分 FSC から生成されるファジィ点として実体化される．



(a) 手書きストローク　(b) 更新された FSC グラフ　(c) 編集された幾何オブジェクト



(d) 手書きストローク　(e) 更新された FSC グラフ　(f) 編集された幾何オブジェクト

図 5.9: 複数の幾何オブジェクトに対する提案手書き編集操作の例

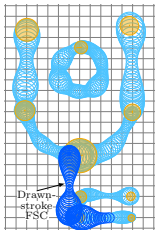


図 5.10: 図 5.9(a) の手書きストロークから生成された手書きストローク FSC

FSC パスグラフ： FSC パスグラフは，交互に並んだドローイングノードとコネクタノードで構成されるノード集合で構成される有向グラフであり，循環することもある．1 つの FSC パスグラフは，1 つの幾何オブジェクトに対応する．

初期条件では幾何オブジェクトが存在しないため，FSC グラフ G は $V_G = \emptyset$，$E_G = \emptyset$ と初期化される．ここで，V_G および E_G はそれぞれ G のノード集合および有向エッジ集合を表す．

5.3.2 FSC 生成法を使用した FSC 生成

手書きストロークから，FSC 生成法（5.2.1 節を参照）を使用して手書きストローク FSC $\tilde{\boldsymbol{s}}$ を生成する．

たとえば，図 5.9(a) の新しい手書きストロークを既存の FSC グラフの上に描画すると，図 5.10 に示すように手書きストローク FSC が生成される．

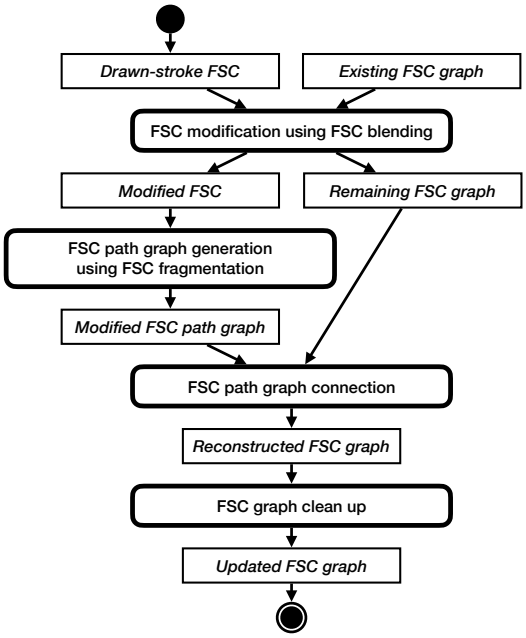


図 5.11: FSC グラフ更新プロセス

$N \leftarrow 0$.
for all $\tilde{\boldsymbol{v}} \in V_G$ **do** ▷ Candidate selection
  if $\tilde{\boldsymbol{v}}'$ is drawing node and $P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}) > \theta$ **then**
    $\tilde{\boldsymbol{o}}_N \leftarrow \tilde{\boldsymbol{v}}$.
    $N \leftarrow N + 1$.
Sort $\tilde{\boldsymbol{o}}_i$ $(0 \leq i < N)$ in descending order by $P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{o}}_i)$.
$\tilde{\boldsymbol{b}} \leftarrow \tilde{\boldsymbol{s}}$
for $i \leftarrow 0$ to $N - 1$ **do** ▷ Blending attempts
  if $P_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{o}}_i) > \theta$ **then**
    $\tilde{\boldsymbol{b}} \leftarrow \mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}}_i)$
    $V_G \leftarrow V_G \setminus \{\tilde{\boldsymbol{o}}_i\}$
    $E_G \leftarrow E_G \setminus \{(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \mid (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \in E_G, \tilde{\boldsymbol{u}} = \tilde{\boldsymbol{o}}_i \text{ or } \tilde{\boldsymbol{v}} = \tilde{\boldsymbol{o}}_i\}$
  else
    break

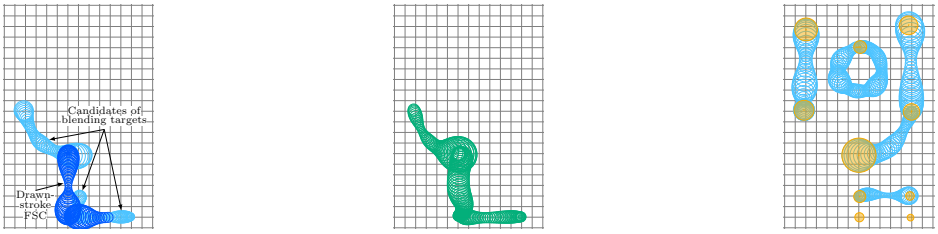
図 5.12: FSC 融合法を使用した FSC 修正のためのアルゴリズム

5.3.3 FSC グラフの更新

手書きストローク FSC $\tilde{\boldsymbol{s}}$ を用いて，図 5.11 に示されているプロセスを通じて FSC グラフ G を更新する．このプロセスのそれぞれ部分を以下に説明する．

5.3.3.1 FSC 融合法を使用した FSC 修正

図 5.12 に示されている手順に従って，修正 FSC $\tilde{\boldsymbol{b}}$ を生成する．ここではまず，FSC グラフ G から融合対象の候補としていくつかのドローイングノード $\tilde{\boldsymbol{o}}_i$ を選択する．その後，FSC 融合法（5.2.2 節を参照）を使用して順次 $\tilde{\boldsymbol{o}}_i$ と手書きストローク FSC $\tilde{\boldsymbol{s}}$ の融合を試みていき，融合



(a) 融合対象の候補　(b) 修正 FSC　(c) 残存 FSC グラフ

図 5.13: 図 5.10 に示す手書きストローク FSC を用いた FSC 修正

対象として融合に関与した $\tilde{\boldsymbol{o}}_i$ は残存 FSC グラフ G から削除する．このアルゴリズムにおいて，$P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}})$ は $\tilde{\boldsymbol{v}}$ が $\tilde{\boldsymbol{s}}$ と重なっている可能性のグレードを表し，FSC 融合法によって計算される[*3]．関数 $\mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ は次のように定義される．

関数 1　$\mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ は，2 つの FSC $\tilde{\boldsymbol{b}}$ と $\tilde{\boldsymbol{o}}$ を取り，FSC 融合法による $\tilde{\boldsymbol{b}}$ と $\tilde{\boldsymbol{o}}$ の融合結果である FSC を返す関数である[*4]．

例を図 5.13 に示す．図 5.10 の FSC グラフから (a) の融合対象候補が選択される．これらを手書きストローク FSC と融合すると，(b) の修正 FSC が生成され，(c) の FSC グラフが残存する．

5.3.3.2　FSC フラグメンテーション法を使用した FSC パスグラフ生成

まず，修正 FSC $\tilde{\boldsymbol{b}}$ から，FSC フラグメンテーション法（5.2.3.1 節を参照）を使用して FSC フラグメント列

$$F_{\tilde{\boldsymbol{b}}} = \{ f_i^{\tilde{\boldsymbol{b}}} = (\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}, l^{f_i^{\tilde{\boldsymbol{b}}}}) \mid 0 \leq i < N^{F_{\tilde{\boldsymbol{b}}}} \} \tag{5.2}$$

を生成する．ただし，$f_i^{\tilde{\boldsymbol{b}}}$ は FSC フラグメント，$N^{F_{\tilde{\boldsymbol{b}}}}$ は $F_{\tilde{\boldsymbol{b}}}$ の FSC フラグメントの個数，$\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}$ は $f_i^{\tilde{\boldsymbol{b}}}$ の部分 FSC，$l^{f_i^{\tilde{\boldsymbol{b}}}} \in \{\mathrm{Drawing}, \mathrm{Break}\}$ は $f_i^{\tilde{\boldsymbol{b}}}$ のラベルである．次に，ドローイングノード $\tilde{\boldsymbol{d}}_i = \tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}$ を生成する．ただし，$\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}$ はドローイング FSC フラグメント $f_i^{\tilde{\boldsymbol{b}}}$ $(l^{f_i^{\tilde{\boldsymbol{b}}}} = \mathrm{Drawing})$ の部分 FSC である．また，コネクタノード $\tilde{\boldsymbol{c}}_i = \tilde{\boldsymbol{g}}_{\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}}$ を生成する．ただし，$\tilde{\boldsymbol{g}}_{\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}}$ は，ブレーク FSC フラグメント $f_i^{\tilde{\boldsymbol{b}}}$ $(l^{f_i^{\tilde{\boldsymbol{b}}}} = \mathrm{Break})$ の部分 FSC $\tilde{\boldsymbol{s}}^{f_i^{\tilde{\boldsymbol{b}}}}$ の重心として導出されたファジィ点である．その後，以下のように FSC パスグラフ S を構築する．

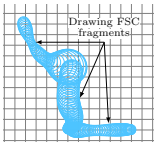
$$V_S = \{ \tilde{\boldsymbol{v}}_i^S \mid 0 \leq i < N^{F_{\tilde{\boldsymbol{b}}}} \}, \tag{5.3}$$

$$E_S = \{ (\tilde{\boldsymbol{v}}_i^S, \tilde{\boldsymbol{v}}_{i+1}^S) \mid 0 \leq i < N^{F_{\tilde{\boldsymbol{b}}}} - 1 \}, \tag{5.4}$$

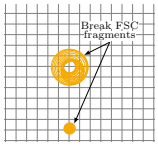
$$\tilde{\boldsymbol{v}}_i^S = \begin{cases} \tilde{\boldsymbol{d}}_i & (l^{f_i^{\tilde{\boldsymbol{b}}}} = \mathrm{Drawing}) \\ \tilde{\boldsymbol{c}}_i & (l^{f_i^{\tilde{\boldsymbol{b}}}} = \mathrm{Break}) \end{cases}, \tag{5.5}$$

ただし，V_S は S のノード集合であり，E_S は S の有向エッジ集合である．

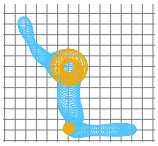
[*3] $P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}})$ の実装は付録 B に示す．
[*4] $\mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ の実装は付録 C に示す．



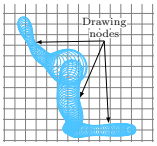
(a) ドローイング FSC フラグメント



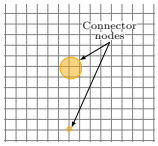
(b) ブレーク FSC フラグメント



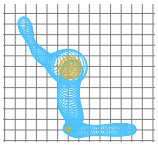
(c) FSC フラグメント列



(d) ドローイングノード



(e) コネクタノード



(f) FSC パスグラフ

図 5.14: 図 5.13(b) の修正 FSC からの FSC パスグラフ生成

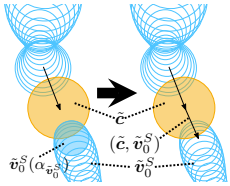
if $\tilde{\boldsymbol{v}}_0^S$ is a drawing node **then**
  $\tilde{\boldsymbol{c}} \leftarrow \mathrm{FindConnectionCandidate}(G, \tilde{\boldsymbol{v}}_0^S)$
  if $\tilde{\boldsymbol{c}}$ is not Null **then** ▷ Case 1 (図 5.16 (a))
    $E_G \leftarrow E_G \cup \{(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{v}}_0^S)\}$
  else ▷ Case 2 (図 5.16 (b))
    $\tilde{\boldsymbol{c}} \leftarrow \tilde{\boldsymbol{v}}_0^S(\alpha_{\tilde{\boldsymbol{v}}_0^S})$
    $V_G \leftarrow V_G \cup \{\tilde{\boldsymbol{c}}\}$
    $E_G \leftarrow E_G \cup \{(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{v}}_0^S)\}$
else
  $\tilde{\boldsymbol{c}} \leftarrow \mathrm{FindBlendingCandidate}(G, \tilde{\boldsymbol{v}}_0^S)$
  if $\tilde{\boldsymbol{c}}$ is not Null **then** ▷ Case 3 (図 5.16 (c))
    $\tilde{\boldsymbol{c}} \leftarrow \frac{\tilde{\boldsymbol{c}}+\tilde{\boldsymbol{v}}_0^S}{2} = \left\langle \frac{\boldsymbol{c}+\boldsymbol{v}_0^S}{2}, \frac{r_{\tilde{\boldsymbol{c}}}+r_{\tilde{\boldsymbol{v}}_0^S}}{2} \right\rangle$
    $E_G \leftarrow E_G \cup \{(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{v}}) \mid \tilde{\boldsymbol{v}} \in V_G, \tilde{\boldsymbol{v}} \text{ is a successor of } \tilde{\boldsymbol{v}}_0^S.\}$
    $E_G \leftarrow E_G \setminus \{(\tilde{\boldsymbol{v}}_0^S, \tilde{\boldsymbol{v}}) \mid \tilde{\boldsymbol{v}} \in V_G\}$
    $V_G \leftarrow V_G \setminus \{\tilde{\boldsymbol{v}}_0^S\}$
  else ▷ Case 4 (図 5.16 (d))
    Do nothing.

図 5.15: $\tilde{\boldsymbol{v}}_0^S$ に関する FSC パスグラフ接続のためのアルゴリズム

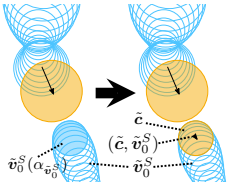
例を図 5.14 に示す．まず，図 5.13(b) に示した修正 FSC から，(a) に示したドローイング FSC フラグメントと (b) に示したブレーク FSC フラグメントからなる (c) の FSC フラグメント列が生成される．次に，これらのドローイング FSC フラグメントとブレーク FSC フラグメントから (d) のドローイングノードと (e) のコネクタノードが生成される．その後，(f) に示すように，ドローイングノードとコネクタノードで構成される FSC パスグラフが生成される．

5.3.3.3 FSC パスグラフ接続

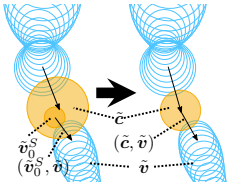
まず，FSC パスグラフ S を残存 FSC グラフ G に追加する．次に，図 5.15 に示されている手順を適用することにより，S の先頭のノード $\tilde{\boldsymbol{v}}_0^S$ に関する FSC パスグラフ接続を実行する．ここでは，図 5.16 に示すそれぞれの場合に応じて，コネクタノード接続，コネクタノード生成，コネクタノード融合，何もしない処理のいずれかを実行する．関数 $\mathrm{FindConnectionCandidate}(G, \tilde{\boldsymbol{v}}_0^S)$ お



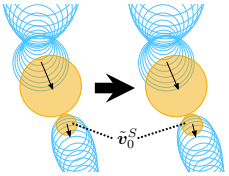
(a) Case 1：$\tilde{\boldsymbol{v}}_0^S$ を $\tilde{\boldsymbol{c}}$ へ接続するコネクタノード接続



(b) Case 2：新たな $\tilde{\boldsymbol{c}}$ を生成するコネクタノード生成



(c) Case 3：$\tilde{\boldsymbol{v}}_0^S$ を $\tilde{\boldsymbol{c}}$ と融合するコネクタノード融合



(d) Case 4：何もしない処理

図 5.16: $\tilde{\boldsymbol{v}}_0^S$ に関する FSC パスグラフ接続における 4 つの場合

よび FindBlendingCandidate$(G, \tilde{\boldsymbol{v}}_0^S)$ は，コネクタノード $\tilde{\boldsymbol{c}}$ がファジィ点 $\tilde{\boldsymbol{p}}$ と重なっている可能性のグレードを表す可能性測度 $\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{p}})$（式 (5.6)）を使用して，次のように定義される．

関数 2　FindConnectionCandidate$(G, \tilde{\boldsymbol{v}}_0^S)$ は，FSC グラフ G とドローイングノード $\tilde{\boldsymbol{v}}_0^S$ を入力として受け取り，以下の条件を満たし $\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{v}}_0^S(\alpha_{\tilde{\boldsymbol{v}}_0^S}))$ を最大化するコネクタノード $\tilde{\boldsymbol{c}} \in V_G$ を返す関数である．ただし，$\alpha_{\tilde{\boldsymbol{v}}_0^S}$ は $\tilde{\boldsymbol{v}}_0^S$ の開始時刻であり，$\tilde{\boldsymbol{v}}_0^S(\alpha_{\tilde{\boldsymbol{v}}_0^S})$ は $\tilde{\boldsymbol{v}}_0^S$ のファジィ始点である．

1. コネクタノード $\tilde{\boldsymbol{c}}$ は後続ノードを持たない．
2. $\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{v}}_0^S(\alpha_{\tilde{\boldsymbol{v}}_0^S}))$ がしきい値 θ を超える．

返すコネクタノードが存在しない場合は，Null が返される．

関数 3　FindBlendingCandidate$(G, \tilde{\boldsymbol{v}}_0^S)$ は，FSC グラフ G とコネクタノード $\tilde{\boldsymbol{v}}_0^S$ を入力として受け取り，以下の条件を満たし $\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{v}}_0^S)$ を最大化するコネクタノード $\tilde{\boldsymbol{c}} \in V_G$ を返す関数である．

1. コネクタノード $\tilde{\boldsymbol{c}}$ は後続ノードを持たない．
2. $\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{v}}_0^S)$ がしきい値 θ を超える．

返すコネクタノードが存在しない場合は，Null が返される．

$\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{p}})$ の具体的な定義は，ファジィ理論 [36] に基づいて以下のように与えられる．

$$\pi_{\tilde{\boldsymbol{c}}}(\tilde{\boldsymbol{p}}) = \sup_{\boldsymbol{w} \in \mathbb{E}^2} \mu_{\tilde{\boldsymbol{p}}}(\boldsymbol{w}) \wedge \mu_{\tilde{\boldsymbol{c}}}(\boldsymbol{w}) \tag{5.6}$$

ここで，$\mathbb{E}^2$ は 2D のユークリッド空間，$\wedge$ は min 演算で計算される論理積を表す．その後，同様の手順を適用することにより，S の末尾ノード $\tilde{\boldsymbol{v}}_{N^{F_{\tilde{\boldsymbol{b}}}}-1}^S$ にに関する FSC パスグラフ接続を実行する．

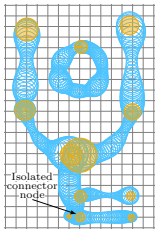


図 5.17: 図 5.14(f) の FSC パスグラフと図 5.13(c) の残存 FSC グラフから再構築される FSC グラフ

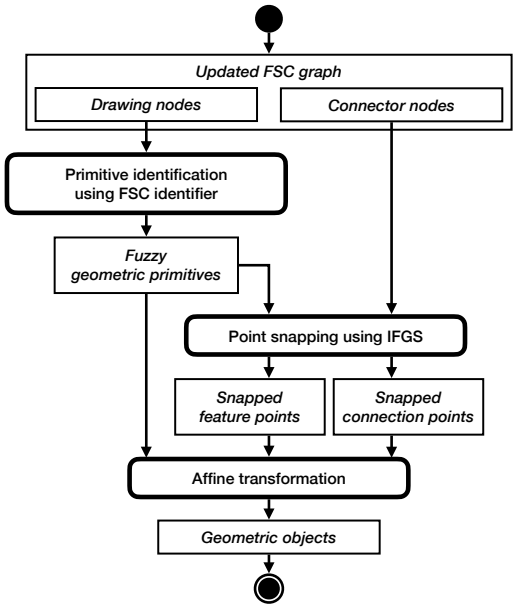


図 5.18: 横断的な幾何オブジェクト同定のプロセス

たとえば，図 5.14(f) の FSC パスグラフを使用すると，図 5.13(c) の残存 FSC グラフが再構築されて図 5.17 の FSC グラフが形成される．

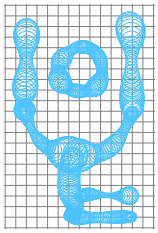
5.3.3.4　FSC グラフクリーンアップ

FSC グラフ G からすべての孤立したコネクタノードを削除する．

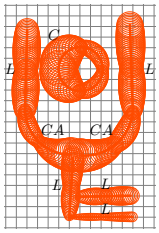
たとえば，図 5.17 に示されている 1 つの孤立したコネクタノードが削除され，図 5.9(b) に示されているように FSC グラフが更新される．

5.3.4　横断的な幾何オブジェクト同定

FSC グラフが更新されるたびに，図 5.18 に示されているプロセスを通じて，FSC グラフ全体を複数の幾何オブジェクトとして同定する．このプロセスのそれぞれの部分を以下に説明する．



(a) FSC グラフ内のドローイングノード



(b) 同定されたファジィ幾何プリミティブ

図 5.19: 図 5.9(b) の FSC グラフに対する FSC 同定法を使用した幾何プリミティブ同定

5.3.4.1 FSC 同定法を使用した幾何プリミティブ同定

FSC 同定法（5.2.3.2 節を参照）を使用して FSC グラフ G のそれぞれのドローイングノードを同定し，ファジィ幾何プリミティブの集合

$$P = \{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}} = \text{Identify}(\tilde{\boldsymbol{v}}) \mid \tilde{\boldsymbol{v}} \in V_G, \tilde{\boldsymbol{v}} \text{ is a drawing node}\} \tag{5.7}$$

を生成する．関数 $\text{Identify}(\tilde{\boldsymbol{v}})$ は次のように定義される．

関数 4 $\text{Identify}(\tilde{\boldsymbol{v}})$ は，FSC $\tilde{\boldsymbol{v}}$ を受け取り，FSC 同定法によって同定されたファジィ幾何プリミティブを返す関数である．

図 5.19 に例を示す．図 5.9(b) の FSC グラフから (a) のように抽出されたすべてのドローイングノードは，それぞれ (b) に示されているファジィ幾何プリミティブとして同定される．

5.3.4.2 IFGS を使用したポイントスナッピング

ファジィ幾何プリミティブを配置する準備として，IFGS（5.2.3.3 節を参照）を使用して，FSC グラフ G 内でファジィ幾何プリミティブの始点と終点のそれぞれに隣接するコネクタノードをそれぞれスナッピングする．そのために，接続点の集合

$$\Gamma = \{\boldsymbol{\gamma}_{\tilde{\boldsymbol{v}}} = \text{Snap}(\tilde{\boldsymbol{v}}) \mid \tilde{\boldsymbol{v}} \in V_G, \tilde{\boldsymbol{v}} \text{ is a connector node}\} \tag{5.8}$$

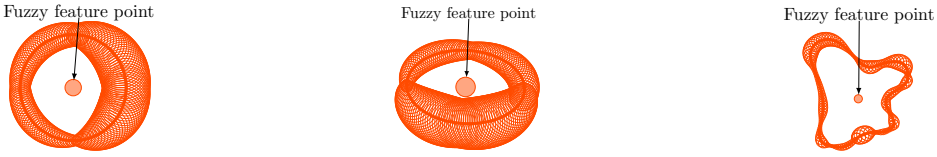
を生成する．関数 $\text{Snap}(\tilde{\boldsymbol{p}})$ は次のように定義される．

関数 5 $\text{Snap}(\tilde{\boldsymbol{p}})$ は，ファジィ点 $\tilde{\boldsymbol{p}}$ を取り，IFGS を使用して $\tilde{\boldsymbol{p}}$ をスナッピングした結果として得られる点を返す関数である．

さらに，IFGS を使用して，閉じた幾何プリミティブ $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ のファジィ特徴点 $\tilde{\boldsymbol{g}}_{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}}$ をスナッピングし，特徴点の集合

$$\Phi = \{\boldsymbol{\varphi}_{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}} = \text{Snap}(\tilde{\boldsymbol{g}}_{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}}) \mid \tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}} \in P, \text{curve class of } \tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}} \text{ is either } C, E, \text{or } FC\} \tag{5.9}$$

を生成する．ここで，ファジィ特徴点 $\tilde{\boldsymbol{g}}_{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}}$ は，図 5.20 に示すように，閉じたファジィ幾何プリミ



(a) C　(b) E　(c) FC

図 5.20: 閉じたファジィ幾何プリミティブのファジィ特徴点

(a) コネクタノードとファジィ特徴点　(b) スナッピングされた接続点と特徴点

図 5.21: 図 5.9(b) の FSC グラフに関する IFGS を使用したポイントスナッピング

ティブ $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の重心とする[*5]．こうするのは，閉じた曲線の始点と終点は互いに非常に接近しており準縮退状態になってしまっているためである．

例を図 5.21 に示す．図 5.21(a) に示すように，図 5.9(b) の FSC グラフから抽出されたすべてのコネクタノードと，図 5.19(b) の C から導出したファジィ特徴点がある場合，IFGS は (b) に示すようにそれらをグリッド点にスナッピングする．

5.3.4.3　アフィン変換

接続点と特徴点に従って，それぞれのファジィ幾何プリミティブ $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}} \in P$ を配置し，以下のように幾何オブジェクトを構築する．

- $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の曲線クラスが開いている（L，CA，EA，FO）場合は，相似変換を $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ に適用し，$\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の始点は $\boldsymbol{\gamma}_{\mathrm{Pred}(G,\tilde{\boldsymbol{v}})} \in \Gamma$ に，$\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の終点は $\boldsymbol{\gamma}_{\mathrm{Succ}(G,\tilde{\boldsymbol{v}})} \in \Gamma$ に移動させる．
- $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の曲線クラスが閉じている（C，E，FC）場合は，相似変換を $\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ に適用し，$\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の始点は $\boldsymbol{\gamma}_{\mathrm{Pred}(G,\tilde{\boldsymbol{v}})} \in \Gamma$ に，$\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}$ の特徴点は $\boldsymbol{\varphi}_{\tilde{\boldsymbol{p}}_{\tilde{\boldsymbol{v}}}} \in \Phi$ に移動させる．

ここで，$\mathrm{Succ}(G,\tilde{\boldsymbol{v}})$ と $\mathrm{Pred}(G,\tilde{\boldsymbol{v}})$ は次のように定義される．

関数 6　$\mathrm{Succ}(G,\tilde{\boldsymbol{v}})$ は，FSC グラフ G とドローイングノード $\tilde{\boldsymbol{v}}$ を取り，G の $\tilde{\boldsymbol{v}}$ の後続ノードを返す．

[*5] ファジィ特徴点の決定方法にはさらなる議論が必要であると思われるが，ここでは，ファジィ幾何プリミティブの重心を特徴点として使用することとした．

関数 7　Pred$(G, \tilde{\boldsymbol{v}})$ は，FSC グラフ G とドローイングノード $\tilde{\boldsymbol{v}}$ を取り，G の $\tilde{\boldsymbol{v}}$ の先行ノードを返す．

たとえば，図 5.19(b) のファジィ幾何プリミティブを使用すると，図 5.9(c) に示すように，4 つの幾何オブジェクト（L-CA-L-L，CA-L，C，L）が生成される．

5.4　実装と実験

前節で提案したプロセスを実装した作図インターフェースを構築した後，ユーザが重ね書きを繰り返すことを通して複数の幾何オブジェクトを横断的かつインタラクティブに編集しながら作図を完了できることを示すために，いくつかの実験を実施した[*6]．

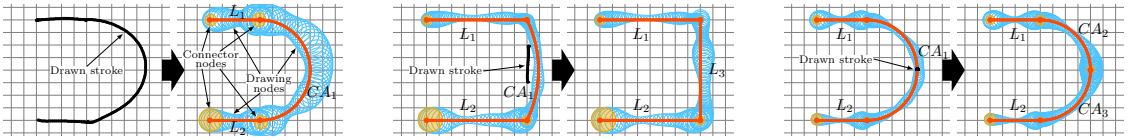
5.4.1　提案する手書き編集操作の実装

提案した操作のプロセスを実装したインタラクティブな作図インターフェースを構築し，Intel (R) Core (TM) i5-7400 CPU @ 3.00 GHz と 16 GB の RAM を備えたパーソナルコンピュータに，液晶ディスプレイ（LCD）が一体化したペンタブレットを接続して，動作させた．2 つのハッシュテーブルを使用して FSC グラフ G を実装した．一方は，G のノードをその後続ノードに関連付け，もう一方は G のノードをその先行ノードに関連付けた．該当するノードがない場合は，Null を関連付けた．したがって，提案したプロセスの $|V_G|$ に関する時間計算量は，図 5.12 の 6 行目でソートされるため，$O(|V_G| \log(|V_G|))$ となる．ここで，$|V_G|$ は FSC グラフ G のノード集合 V_G の要素数である．実際，それぞれの手書きストロークに対する平均応答時間（1 つの手書きストロークが描画されてから編集された幾何オブジェクトが得られるまでの時間）が 0.5 秒以内であったため，インターフェースはリアルタイムシステムと見なすことができる．このインターフェースでは，ユーザが手書きストロークを描画するたびに，提案されたプロセスが実行され，更新された FSC グラフと結果の幾何オブジェクトの両方が即座に表示される．したがって，ユーザは FSC グラフと幾何オブジェクトの変化を考慮して，重ね書き操作のシーケンスをインタラクティブに計画できる．

5.4.2　提案した手書き編集操作の例

インターフェースを使用した典型的な編集操作の例を図 5.22 に示す．この結果は，5.1 節で説明したように，提案されたプロセスが，個別の幾何オブジェクトと複数の幾何オブジェクトの両方で一貫した編集操作を実現できることを示している．

[*6] 5.4.2 節および 5.4.3 節の実験を再現するためのプログラムソースコードとデータは，`https://github.com/nUhLXJCRwnKG/Program-source-code-and-data` で入手できる．



(a) L_1-CA_1-L_2 の入力　(b) CA_1 の L_3 への変形　(c) CA_1 の CA_2 と CA_3 への分割

(d) CA_1 と CA_2 の CA_3 への統合　(e) L_1-L_2-L_3 の L_1，L_3，L_4 への分解　(f) L_1 と L_2 の L_2-CA_1-L_1 への連結

(g) CA_1-CA_2 と CA_3-CA_4 の分解と CA_1 と CA_4 の L_1 への統合の組み合わせ　(h) CA_1-CA_2-CA_3-L_1 の分解と CA_1 と L_1 の L_2 への統合の組み合わせ　(i) FO_1-FO_2-FO_3 の分解と L_1，L_2，L_3，FO_2 の CA_1 への統合の組み合わせ

図 5.22: 典型的な編集操作の例

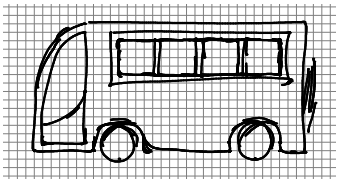


図 5.23: バスをデザインするための全ての手書きストローク

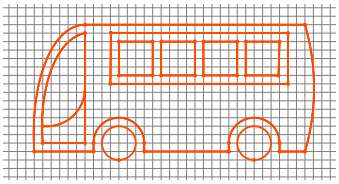


図 5.24: バスの最終結果

5.4.3　提案した手書き編集操作を用いたプログレッシブな作図の例

構築されたインターフェースを使用して，提案した手書き編集操作を繰り返しながら作図を行なった．描画されたすべての手書きストロークを図 5.23 に，最終結果は図 5.24 に示す．　作図過程を抜粋したものを，図 5.25 に示す．この場合，ユーザはラフなボディから始めて，提案された

編集操作を繰り返して，タイヤ，ウィンドウ，ドアなどの詳細なパーツを段階的に形成し，複数の幾何オブジェクトで構成される図面を完成させた．手書きストローク数は 78 個で，作図完了までに 3.63 分かかった．

図を詳しく見てみると，(a) から (d) までに示すようにタイヤをボディから分離してフェンダを形成したり，(e) から (f) までに示すように内側の長方形を分解し，柱を追加してウィンドウを形成するような，複数の幾何オブジェクトのトポロジカルな構造を修正する編集操作を実行できたことが確認できる．

別の作図例の手書きストローク，最終結果および作図過程の抜粋を図 5.26 から 5.28 までに示す． ここでは，手書きストロークの数は 122 個で，作図には 6.16 分かかった．より実際的な作図例における手書きストロークと最終結果を図 5.29 と 5.30 に示す．ここでは，手書きストロークの数は 651 個で，作図には 44.0 分かかった．

上記の結果は，構築されたインターフェースでは，たとえばユーザがラフスケッチから始めて，繰り返し重ね書きを行うことで詳細なデザインを完了するといったような，創造的な作図を行うことの可能な編集操作が実現されたことを示している．

5.4.4 提案した手書き編集操作のプログレッシブな作図におけるユーザビリティ

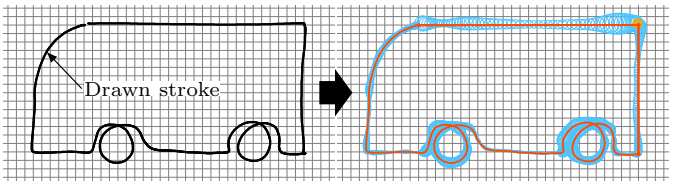
提案したインターフェースでは，操作シーケンスの計画は，ユーザの創造性に大きく依存して変化する．したがって，定量的評価において，目標図形のみが与えられ，被験者が計画を自由に立てられるという条件下で，作図タスクを完了するまでの時間を測定し，効率の観点から調査することは，適切ではない．そこで，まず，それぞれの被験者に，限定された作図タスクの操作シーケンスに関する被験者ごとの計画を確立させた．次に，限定された作図タスクの完了時間を測定することにより，提案されたインターフェースをコントローラビリティの観点から定量的に評価した．図 5.31 と図 5.32 に示すように，限定された作図タスクを設定した．それぞれの被験者は，図 5.31 の初期状態の FSC グラフと幾何オブジェクトから開始し，図 5.32 に示す幾何オブジェクトを完成させるまで，提案された編集操作をプログレッシブに繰り返す必要があった．このタスクは，ユーザがラフスケッチから開始し，操作 1 を除くすべての提案された操作を使用して詳細なデザインを完了する状況を想定したものである．このタスクでは，屋根とボディを連結するために操作 6 が必要となり，ウィンドウを分解したり，タイヤをボディから分離するために操作 7 が必要となる．タスクの完了時間を測定する前には，それぞれの被験者がその被験者自身の計画を立てるために，最初に，ビデオを通じてタスクを完了するための操作シーケンスの例を提示した．また，スーパーバイザのアドバイスのもとで，タスクを最初から最後まで一通り体験してもらった．その後，被験者ごとにタスク完了時間を 10 回測定した．被験者は操作中にスーパーバイザーにアドバイスを請うことが許された．

結果が図 5.33 に，手書きストローク数が図 5.34 に示されている．SKIT[1] の扱いに慣れている Expert A から D までは，平均タスク完了時間 2.68 分でタスクを完了した．これは，このような編集操作としては十分短い時間と考えられる．この結果は，提案されたインターフェースが訓練を

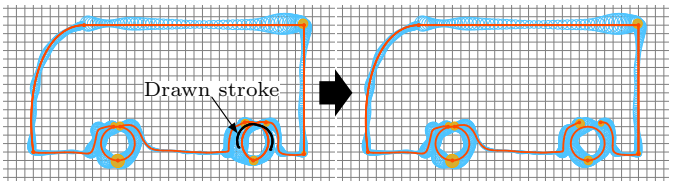
受けたユーザにとって制御可能であることを示している．SKIT に不慣れな Beginner A から F までは，Expert A から D までに比べて平均 9.11 分という長い時間でタスクを完了したが，個人差はあるものの，タスク完了時間が短くなる傾向があることが確認できた．この結果は，初心者が提案されたインターフェースの制御に慣れていくことができることを示している．

5.5 本章のまとめ

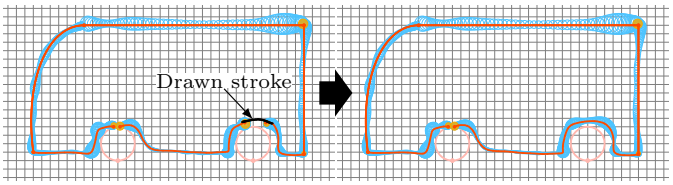
本章では，7 つのクラスの幾何プリミティブ（線，円，円弧，楕円，楕円弧，閉自由曲線，開自由曲線）で構成される複数の幾何オブジェクトに対する手書き編集操作を提案し，汎用手書き CAD インターフェースを実現した．まず描画軌跡と接続点およびそれらの位相関係に関する情報を保持する FSC グラフを導入した．その後，FSC グラフに基づいて，FSC 生成法，FSC 融合法，FSC フラグメンテーション法，FSC 同定法，IFGS などの要素技術を統合することにより，上の手書き編集操作を実現する新しいプロセスを提案した．提案したプロセスを実装する作図インターフェースを使用した実験結果では，編集操作が，分解や連結などの複数の幾何オブジェクト間での修正と，入力，変換，分割，統合などの個別の幾何オブジェクト内での修正を実現できることを示している．また，プログレッシブな作図例を使用して，編集操作によってユーザが創造的に作図できることを示した．さらに，限定された作図タスクを完了する時間を測定することにより，訓練したユーザが実装したインターフェースを制御でき，初心者もその制御に慣れることができることを確認した．



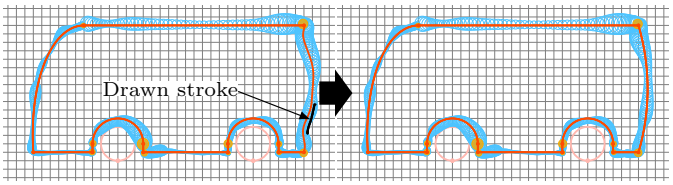
(a) 入力



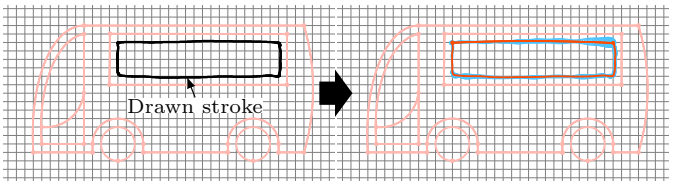
(b) 分解と統合の組み合わせ



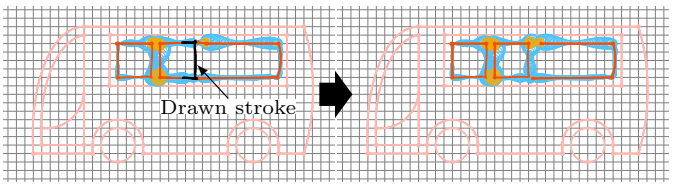
(c) 連結と統合の組み合わせ



(d) 変形



(e) 入力



(f) 分解と統合の組み合わせ

図 5.25: バスの作図過程の抜粋

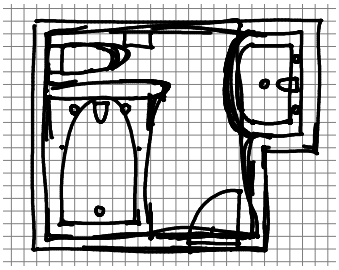


図 5.26: バスルームをデザインするための全ての手書きストローク

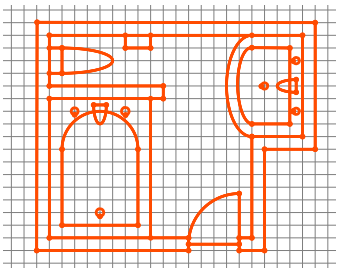
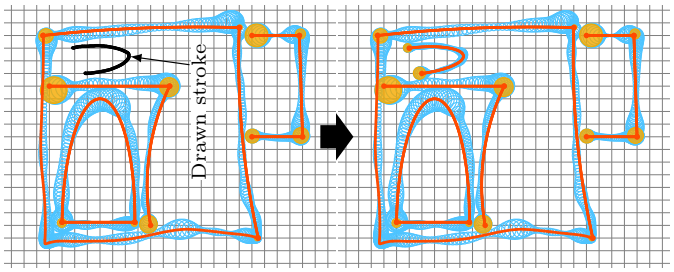
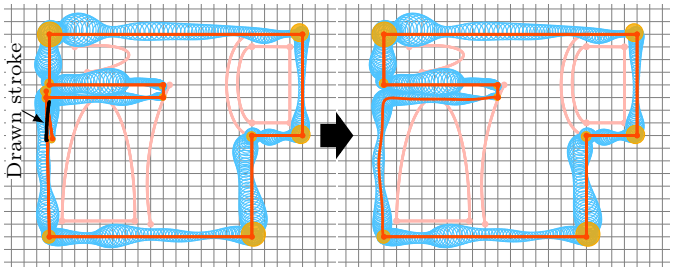


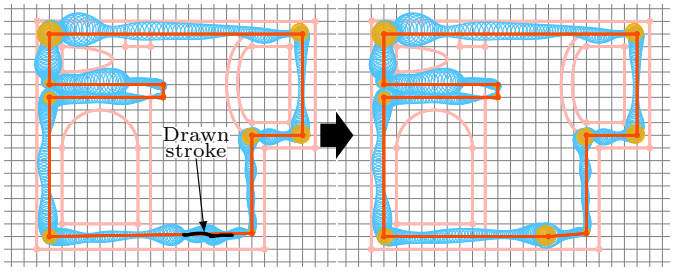
図 5.27: バスルームの最終結果



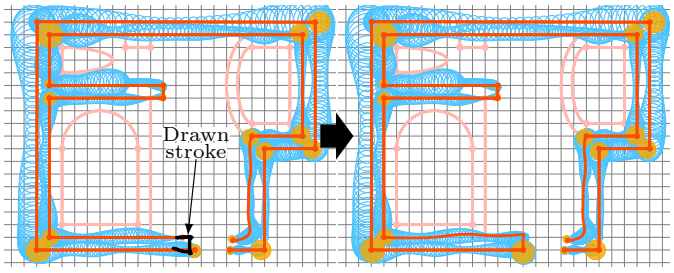
(a) 入力



(b) 統合



(c) 分割



(d) 連結

図 5.28: バスルームの作図過程の抜粋

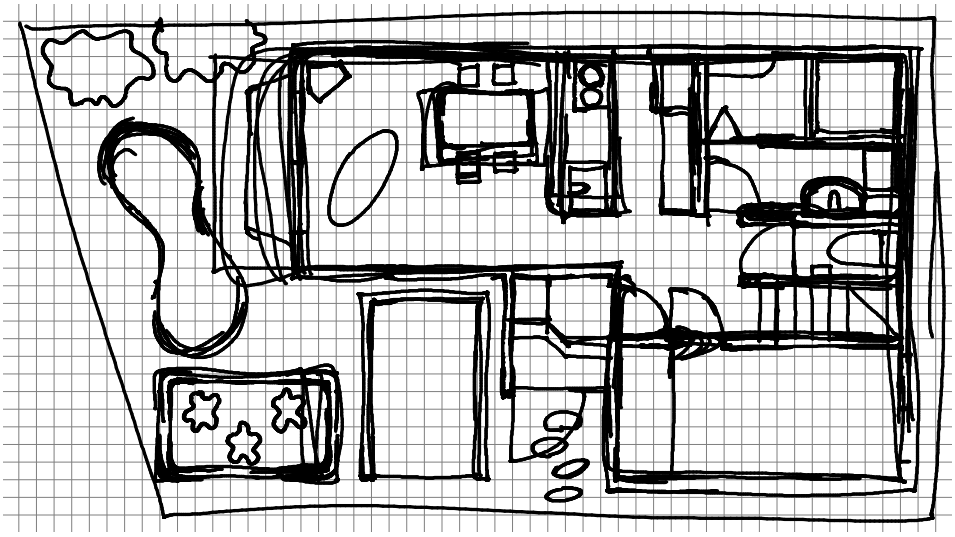


図 5.29: フロアプランをデザインするための全ての手書きストローク

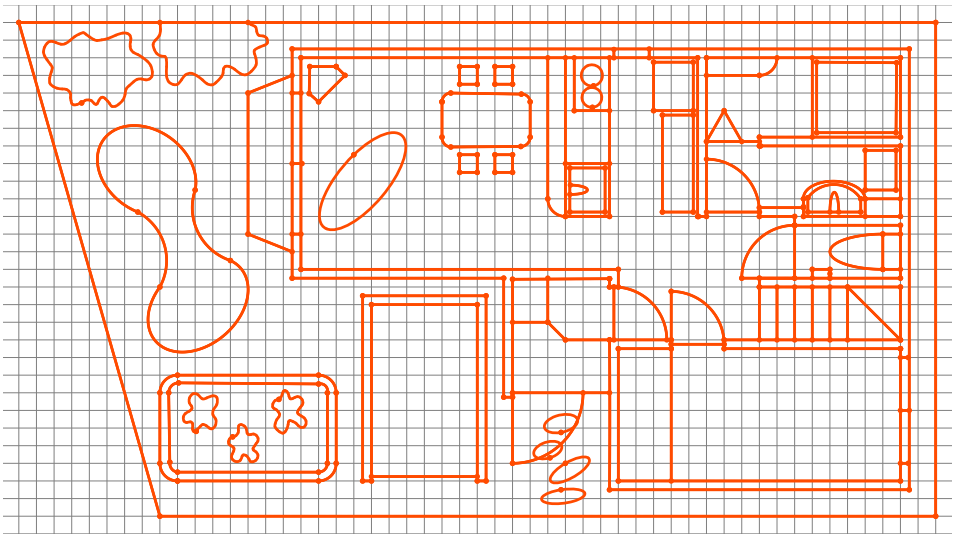


図 5.30: フロアプランの最終結果

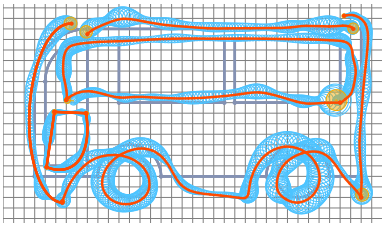


図 5.31: 目標図形上の初期状態の FSC グラフと幾何オブジェクト

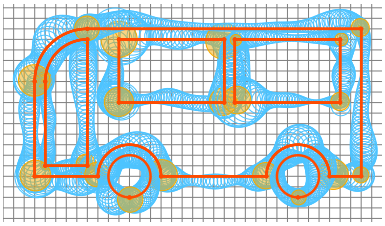


図 5.32: 目標図形にマッチする完成された FSC グラフと幾何オブジェクトの例

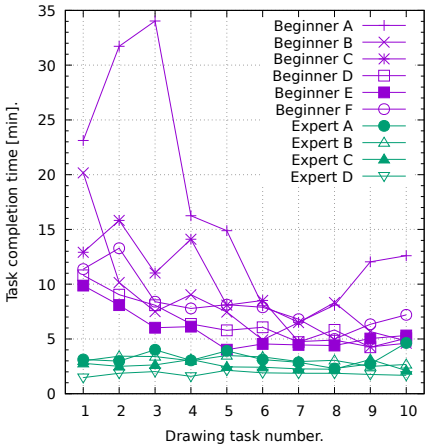


図 5.33: それぞれの作図タスクを完成させるために要した時間

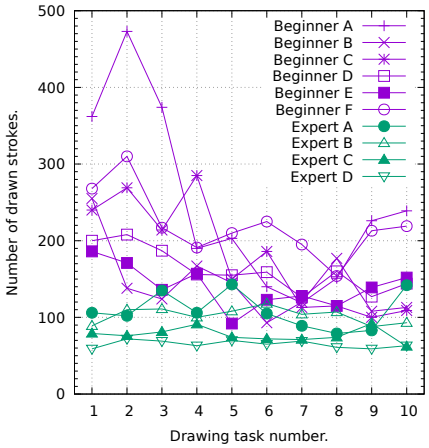


図 5.34: それぞれの作図タスクを完成させるために要した手書きストロークの数

第 6 章

結論

6.1 各章のまとめ

1 章から 5 章までで示したことを以下にまとめる.

1 章 ここでは，2D の CAD システムのための様々な既存の手書きインターフェースについて，扱うことのできる幾何オブジェクトと実行可能な手書き操作の観点から問題点を指摘した．その上で，汎用手書き CAD インターフェースに必要な手書き操作を網羅するインターフェースを初めて実現するという本研究のコントリビューションを明確にした.

2 章 ここでは，FSC 同定法の幾何プリミティブ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法を提案した．さらに，これ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.

3 章 ここでは，従来の FSC 同定法の同定アルゴリズムの仮説ファジィモデルとファジィ推論規則を部分的に改変することで，形状パラメータ量子化を用いずに直接 $n/4$ 円弧および $n/4$ 楕円弧を同定する手法，すなわちサブ曲線同定法を提案した．さらに，サブ曲線同定法が実用的な同定性能を有することを示し，これが手書き CAD インターフェースとして有効に機能することを示した.

4 章 ここでは，MFGS におけるグリッド設定が必要となってしまう問題を克服するために，利用可能なグリッド解像度の数を無限に増やす IFGS という MFGS の拡張を提案した．そして IFGS の有効性と効率性について MFGS と比較して検証する実験を行い，IFGS は幾何オブジェクトのサイズが大きく異なる場合でも，グリッド設定を変更するための面倒な手動操作を必要としないことを示した.

5 章 描画軌跡と接続点およびそれらの位相関係に関する情報を保持する FSC グラフを導入し，FSC グラフに基づいて，FSC 生成法，FSC 融合法，FSC フラグメンテーション法，FSC 同定法，IFGS などの要素技術を統合することにより，自在な手書き編集操作を実現する新しいプロセスを提案した．さらに提案したプロセスを実装する作図インターフェースを使用

した実験結果により，編集操作によってユーザが創造的に作図できることを示した．

6.2 本研究のまとめ

本研究では，SKIT のファジィ論理ベースの要素技術を拡張的に再構成することにより，複数の幾何オブジェクトの形状と位相の自在な重ね書き編集が可能な手書き CAD インターフェースを実現した．これは，7 つのクラスの幾何プリミティブ（線分，円，円弧，楕円，楕円弧，閉自由曲線，開自由曲線）で構成される幾何オブジェクトに対して，分解や連結などの複数の幾何オブジェクト間での編集操作，入力，変換，分割，統合などの個別の幾何オブジェクト内での編集操作およびこれらの複合操作の全てが網羅された初めての手書き CAD インターフェースである．

これを実現するためにまず，準備として 2 章，3 章，4 章において SKIT の要素技術を拡充する提案を行なった．

その後，5 章において，FSC グラフとその更新プロセスを導入し，これらに基づいて実装した手書き CAD インターフェースにより，複数の幾何オブジェクトに対する自在な手書き編集操作を実現した．また，ここでは実験により，プログレッシブな作図例を使用して，編集操作によってユーザが創造的に作図できることを示した．さらに，限定された作図タスクを完了する時間を測定することにより，訓練したユーザが実装したインターフェースを制御でき，初心者もその制御に慣れることができることを確認した．

6.3 実用化に向けた課題と展望

手書きインターフェースの可能性を追求する観点では，本研究で得られた結果は，提案した重ね書き編集操作が 2DCAD システム用の汎用手書きインターフェースを実現することを示していると言える．一方で，インターフェースの効率の観点では，提案したインターフェースは，幾何オブジェクトの位相構造または形状のプログレッシブな修正に十分に使用可能であることが確認できたが，ドラスティックな修正を実行する場合には常に効率的であるとは限らなかった．今後は実用化に向けて，提案したスケッチベースの編集操作と従来の CAD システムで用いられてきた（ドラスティックな修正に対してはより効率的であるかもしれない）コマンドベースの操作を組み合わせる方法について検討する必要がある．

謝辞

本論文は筆者が室蘭工業大学大学院工学研究科工学専攻博士後期課程に在籍中の研究成果をまとめたものである．本研究の遂行にあたりご指導いただいた佐賀聡人教授に深く感謝いたします．本研究に対し副査として助言をいただいた工藤康生教授，岡田吉史准教授に深く感謝いたします．本研究の遂行にあたりサポートしていただいた油谷凛氏，金子輝良氏，中島幸佑氏，吉川友人氏，三輪雄斗氏，田中良樹氏，割田怜氏，佐藤幸夫氏，渡邊天氏，山本奈央氏，三田興史氏，鈴木洋平氏，神谷葉月氏，畑本直哉氏，佐々木海人氏，香田暁人氏，小野寛和氏，佐藤遼氏，飯沼大樹氏，井村駿介氏，為国翔太氏，村越駿平氏，芳村健太氏，藤田佳宏氏，脇本翼氏，前鼻裕朗氏，堀祐太氏，千坂一輝氏，高橋秀輔氏，小向祥実氏，加納友梨奈氏，村木翔太氏，関田拓真氏，鈴木裕太氏，佐々木健氏，坂野巧卓氏，稲葉史郎氏，石川義勝氏，藤田隼輔氏，長畑光紀氏，中野匠氏，塔野岡玲央氏，工藤颯人氏，有澤恭兵氏，荒川和輝氏，菅野一平氏，戀塚雅之氏，花房三太郎氏，藤生涼雅氏，川鰭瞭氏，灰原渉氏，堀越亮我氏，須坂元晴氏に深く感謝いたします．

参考文献

[1] Ryota Kawai, Akira Nishikawa, and Sato Saga. A freehand sketch input front-end processor: SKIT (in japanse). *The IEICE transactions on information and systems Pt. 2*, J88-D-II(5):897–905, may 2005.

[2] Gunay Orbay and Levent Burak Kara. Beautification of design sketches using trainable stroke clustering and curve fitting. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):694–708, 2011.

[3] Xueting Liu, Tien-Tsin Wong, and Pheng-Ann Heng. Closure-aware sketch simplification. *ACM Trans. Graph.*, 34(6), October 2015.

[4] Toru Ogawa, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch simplification by classifying strokes. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1065–1070, 2016.

[5] Chenxi Liu, Enrique Rosales, and Alla Sheffer. StrokeAggregator: Consolidating raw sketches into artist-intended curve drawings. *ACM Trans. Graph.*, 37(4), July 2018.

[6] Gayoung Lee, Dohyun Kim, Youngjoon Yoo, Dongyoon Han, Jung-Woo Ha, and Jaehyuk Chang. Unpaired sketch-to-line translation via synthesis of sketches. In *SIGGRAPH Asia 2019 Technical Briefs*, SA '19, page 45 – 48, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Thomas Baudel. A mark-based interaction paradigm for free-hand drawing. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, page 185 – 192, New York, NY, USA, 1994. Association for Computing Machinery.

[8] Pascal Barla, Joelle Thollot, and François X. Sillion. Geometric clustering for line drawing simplification. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, page 183 – 192, Goslar, DEU, 2005. Eurographics Association.

[9] Timo Fleisch, Florian Rechel, Pedro Santos, and André Stork. Constraint stroke-based oversketching for 3D curves. In *Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'04, page 161 – 165, Goslar, DEU, 2004. Eurographics Association.

[10] Dae-Hyun Kim and Myoung-Jun Kim. A new cubic B-splines design method for pen

input environment. *Ieice Transactions - IEICE*, E92D:69–77, 01 2009.

[11] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. ILoveSketch: As-natural-as-possible sketching system for creating 3D curve models. pages 151–160, 01 2008.

[12] Cindy Grimm and Pushkar Joshi. Just DrawIt: A 3D sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, page 121 – 130, Goslar, DEU, 2012. Eurographics Association.

[13] Heloise Hwawen Hse and A. Richard Newton. Recognition and beautification of multi-stroke symbols in digital ink. *Comput. Graph.*, 29(4):533 – 546, August 2005.

[14] James Arvo and Kevin Novins. Fluid Sketches: Continuous recognition and morphing of simple hand-drawn shapes. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 73–80, New York, NY, USA, 2000. ACM.

[15] Joaquim A. Jorge and Manuel J. Fonseca. A simple approach to recognise geometric shapes interactively. In Atul K. Chhabra and Dov Dori, editors, *Graphics Recognition Recent Advances*, pages 266–274, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[16] Bo Yu and Shijie Cai. A domain-independent system for sketch recognition. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, GRAPHITE '03, page 141 – 146, New York, NY, USA, 2003. Association for Computing Machinery.

[17] Brandon Paulson and Tracy Hammond. PaleoSketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI ' 08, page 1 – 10, New York, NY, USA, 2008. Association for Computing Machinery.

[18] Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. Interactive Beautification: A technique for rapid geometric design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 105–114, New York, NY, USA, 1997. ACM.

[19] Jakub Fišer, Paul Asente, Stephen Schiller, and Daniel Sýkora. Advanced drawing beautification with ShipShape. *Computers & Graphics*, 56:46–58, 2016.

[20] James McCrae and Karan Singh. Sketching piecewise clothoid curves. *Computers & Graphics*, 33(4):452 – 461, 2009.

[21] Ilya Baran, Jaakko Lehtinen, and Jovan Popović. Sketching clothoid splines using shortest paths. *Computer Graphics Forum*, 29(2):655–664, 2010.

[22] Sato Saga, Hiromi Makino, and Junichi Sasaki. The fuzzy spline curve identifier (in japanese). *The Transactions of the Institute of Electronics,Information and Communication Engineers.*, J77-D-II(8):1620–1629, aug 1994.

[23] Sato Saga and Hiromi Makino. Fuzzy spline interpolation and its application to online

freehand curve identification. In *Proceedings 1993 Second IEEE International Conference on Fuzzy Systems*, volume 2, pages 1183–1190, March 1993.

[24] Sato Saga, Hiromi Makino, and Junichi Sasaki. A method for modeling freehand curves - the fuzzy spline interpolation - (in japanese). *The Transactions of the Institute of Electronics,Information and Communication Engineers.*, J77-D-II(8):1610–1619, aug 1994.

[25] Yoichi Sato, Naofumi Yasufuku, and Sato Saga. Sequential fuzzy spline curve generator for drawing interface by sketch (in japanese). *The Transactions of the Institute of Electronics,Information and Communication Engineers.*, 86(2):242–251, feb 2003.

[26] Akira Nishikawa, Sato Saga, and Junji Maeda. Performance improvement of geometric curve sequence recognition in the freehand curve identifier FSCI (in japanese). *Journal of Information Processing*, 51(2):380–390, feb 2010.

[27] Ten Watanabe, Tomohito Yoshikawa, Tomohiko Ito, Yuto Miwa, Takeshi Shibata, and Sato Saga. An infinite-resolution grid snapping technique based on fuzzy theory. *Applied Soft Computing*, 89:106112, 2020.

[28] Sheng Feng Qin, David K. Wright, and Ivan N. Jordanov. On-line segmentation of freehand sketches by knowledge-based nonlinear thresholding operations. *Pattern Recognition*, 34(10):1885–1893, 2001.

[29] C. L. Philip Chen and Sen Xie. Freehand drawing system using a fuzzy logic concept. *Computer-Aided Design*, 28(2):77–89, 1996.

[30] Shuxia Wang, Sheng Feng Qin, and Mantun Gao. New grouping and fitting methods for interactive overtraced sketches. *Vis. Comput.*, 30(3):285 – 297, mar 2014.

[31] Qamar Uddin Khand, Dematapitiya Sumudu, Saga Sato, and Maeda Junji. A multiresolution grid snapping technique based on fuzzy theory. *Journal of Information Processing*, 48(4):1874–1882, apr 2007.

[32] 八木麻理子, 川田洋平, 藤澤誠, and 三浦憲二郎. ペンタブレット入力による G1 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**, 7(3):97–101, 2008.

[33] 森本有紀 and 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**, 56(5):1329–1338, 2015.

[34] Yannick Thiel, Karan Singh, and Ravin Balakrishnan. Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2D curves. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 383 – 392, New York, NY, USA, 2011. Association for Computing Machinery.

[35] Tetsuya Ohkawa and Sato Saga. Refinement of fuzziness generator in the freehand curve identifier FSCI. *The Transactions of the Institute of Electronics,Information and Communication Engineers.*, 82(5):634–643, may 1999.

[36] Lotfi A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.*, 100:9 – 34, April 1999.

[37] 正嶋博. 画数, 筆順, 回転, 区切りに依存しないオンライン手書図形認識方式. **情報処理学会論文誌**, 27(5):492–498, 1986.

[38] Dematapitiya Sumudu, Kawazoe Masatoshi, Nishikawa Akira, Sakurai Masaki, and Saga Sato. Snapping of fuzzy objects using the multi-resolution fuzzy grid snapping technique. *Journal of Information Processing*, 50(2):904–915, feb 2009.

[39] 櫻井将樹 and 佐賀聡人. 多重解像度ファジィグリッドスナッピングを用いた幾何曲線スナッピング—楕円，楕円弧への適用—. volume 2006, pages 119–126, 2006.

[40] Sheng Feng Qin, David K. Wright, and Ivan N. Jordanov. From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge. *Computer-Aided Design*, 32(14):851–866, 2000.

[41] 滝川裕康, 安福尚文, and 佐賀聡人. 手書き曲線同定法 FSCI における同定アルゴリズムの改善. **電子情報通信学会論文誌** *D*, 85(11):1683–1691, 2002.

[42] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—II. *Information Sciences*, 8(4):301–357, 1975.

[43] 森さおり and 佐賀聡人. ファジーニューラルネットワークを用いた学習型手書き曲線同定法. **電子情報通信学会論文誌** *D*, 83(3):375–383, 2000.

[44] 七條直樹 and 佐賀聡人. ファジィ円弧およびファジィ楕円弧のスナッピング法の改善. In **平成** *24* **年度電気情報関連学会北海道支部連合大会講演論文集**, volume 87, 2012.

[45] Ivan E. Sutherland. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, DAC '64, page 6.329 – 6.346, New York, NY, USA, 1964. Association for Computing Machinery.

[46] Michael Gleicher and Andrew Witkin. Drawing with constraints. *The Visual Computer*, 11(1):39–51, 1994.

[47] Eric A. Bier and Maureen C. Stone. Snap-dragging. *SIGGRAPH Comput. Graph.*, 20(4):233–240, August 1986.

[48] Toshiyuki Masui. Hyper snapping. In *Proceedings of the IEEE 2001 Symposia on Human Centric Computing Languages and Environments (HCC'01)*, HCC '01, pages 188–194, Washington, DC, USA, 2001. IEEE Computer Society.

[49] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Adam Eversole. Snap-and-go: Helping users align objects without the modality of traditional snapping. pages 301–310, 2005.

[50] George Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

付録 A

$\mu(\tilde{\boldsymbol{g}}_i)$, $\mu(\tilde{\boldsymbol{g}}_{i-1})$ と $N^{\tilde{\boldsymbol{g}}_i}$ の性質

G_i を解像度に関して降順に並べるようにグリッドインデックス i を割り当てるとき，必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ が i に関して単調非増加となるため，以下の 2 つの性質が導かれる．

$$N^{\tilde{\boldsymbol{g}}_i} \geq 0.5 \Rightarrow \mu(\tilde{\boldsymbol{g}}_i) \geq \mu(\tilde{\boldsymbol{g}}_{i-1}) \tag{A.1}$$

$$N^{\tilde{\boldsymbol{g}}_i} < 0.5 \Rightarrow \mu(\tilde{\boldsymbol{g}}_i) \leq \mu(\tilde{\boldsymbol{g}}_{i-1}) \tag{A.2}$$

命題 (A.1) は以下のように証明される．$N^{\tilde{\boldsymbol{g}}_i} \geq 0.5$ ならば $N^{\tilde{\boldsymbol{g}}_i} \geq (1 - N^{\tilde{\boldsymbol{g}}_i})$ である．必然性値 $N^{\tilde{\boldsymbol{g}}_i}$ は単調非増加となるため $N^{\tilde{\boldsymbol{g}}_{i-1}} \geq N^{\tilde{\boldsymbol{g}}_i}$ が成り立つ．したがって，$N^{\tilde{\boldsymbol{g}}_i} \geq (1 - N^{\tilde{\boldsymbol{g}}_i})$ のとき $N^{\tilde{\boldsymbol{g}}_{i-1}} \geq (1 - N^{\tilde{\boldsymbol{g}}_i})$ となる．それゆえに，

$$\mu(\tilde{\boldsymbol{g}}_{i-1}) = (1 - N^{\tilde{\boldsymbol{g}}_i}) \wedge N^{\tilde{\boldsymbol{g}}_{i-1}} = (1 - N^{\tilde{\boldsymbol{g}}_i}) \tag{A.3}$$

となる．対照的に，$(1 - N^{\tilde{\boldsymbol{g}}_i})$ は単調非減少であるため，$(1 - N^{\tilde{\boldsymbol{g}}_i}) \leq (1 - N^{\tilde{\boldsymbol{g}}_{i+1}})$ が成り立つ．したがって，$N^{\tilde{\boldsymbol{g}}_i} \geq (1 - N^{\tilde{\boldsymbol{g}}_i})$ のとき

$$\begin{aligned} \mu(\tilde{\boldsymbol{g}}_i) &= (1 - N^{\tilde{\boldsymbol{g}}_{i+1}}) \wedge N^{\tilde{\boldsymbol{g}}_i} \\ &\geq (1 - N^{\tilde{\boldsymbol{g}}_i}) \wedge N^{\tilde{\boldsymbol{g}}_i} = (1 - N^{\tilde{\boldsymbol{g}}_i}) \end{aligned} \tag{A.4}$$

となる．式 (A.3) と (A.4) から，次の式が得られ，命題 (A.1) が成り立つ．

$$\mu(\tilde{\boldsymbol{g}}_i) \geq \mu(\tilde{\boldsymbol{g}}_{i-1}), \tag{A.5}$$

命題 (A.2) も同様にして証明できる．

付録 B

$P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}})$ の実装

[25] に基づいて，図 B.1 に示すように EvaluateOverlapping$(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{v}})$ の一部として $P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}})$ を求めるプロセスを実装した．このプロセスにおいて，$\alpha_{\tilde{\boldsymbol{s}}}$ と $\beta_{\tilde{\boldsymbol{s}}}$ は $\tilde{\boldsymbol{s}}$ の開始時刻と終了時刻であり，$\alpha_{\tilde{\boldsymbol{v}}}$ と $\beta_{\tilde{\boldsymbol{v}}}$ は $\tilde{\boldsymbol{v}}$ の開始時刻と終了時刻であり，Δu はサンプリング時間間隔を表す実数である．関数 $\mathrm{Find}_{r_1}(U, i, j)$ は図 B.2 に示すプロセスで定義される．ここで，U は行列である．関数 $\mathrm{Append}(A, a)$ は以下のように定義される．

関数 B. 1　$\mathrm{Append}(A, a)$ は列 A と要素 a を引数としてとり，A の末尾に a を追加して得られる列を返す関数である．

関数 $f_\delta(r)$ は以下のように定義される．

$$f_\delta(r) = \begin{cases} (i_1 - i_0) + (j_1 - j_0) \ (r \text{ is not empty}) \\ -1 \ (\text{otherwise}) \end{cases} \tag{B.1}$$

ここで，r は整数の組の列であり，$(i_0, j_0) = \mathrm{First}(r)$，$(i_1, j_1) = \mathrm{Last}(r)$ である．関数 $\mathrm{First}(A)$ と $\mathrm{Last}(A)$ は以下のように定義される．

関数 B. 2　$\mathrm{First}(A)$ は列 A を引数としてとり，A の先頭の要素を返す関数である．

関数 B. 3　$\mathrm{Last}(A)$ は列 A を引数としてとり，A の末尾の要素を返す関数である．

関数 $\mathrm{Collect}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r)$ は図 B.3 に示すプロセスで定義される．関数 $\mathrm{Prepend}(A, a)$ は以下のように定義される．

関数 B. 4　$\mathrm{Prepend}(A, a)$ は列 A と要素 a を引数としてとり，A の先頭に a を追加して得られる列を返す関数である．

図 B.3 における関数 $\mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j)$ は図 B.4 に示すプロセスで定義される．図 B.5 における関数 $\mathrm{Collect}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r)$ は図 B.5 に示すプロセスで定義される．図 B.5 における関数 $\mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j)$ は図 B.6 に示すプロセスで定義される．関数 $\mathrm{Find}_{r_2}(U, Q, i, j)$ は図 B.7 に示すプロセスで定義される．ここで，Q は整数のペアの集合である．関数 $f_\pi(U, r)$ は

$N^{\tilde{\boldsymbol{s}}} \leftarrow \lceil \frac{\beta_{\tilde{\boldsymbol{s}}} - \alpha_{\tilde{\boldsymbol{s}}}}{\Delta u} \rceil + 1$
$T^{\tilde{\boldsymbol{s}}} \leftarrow \{t_i^{\tilde{\boldsymbol{s}}} = \alpha_{\tilde{\boldsymbol{s}}} + \frac{\beta_{\tilde{\boldsymbol{s}}} - \alpha_{\tilde{\boldsymbol{s}}}}{N^{\tilde{\boldsymbol{s}}} - 1} i \mid 0 \leq i < N^{\tilde{\boldsymbol{s}}}\}$
$P^{\tilde{\boldsymbol{s}}} \leftarrow \{\tilde{\boldsymbol{p}}_i^{\tilde{\boldsymbol{s}}} = \tilde{\boldsymbol{s}}(t_i^{\tilde{\boldsymbol{s}}}) \mid 0 \leq i < N^{\tilde{\boldsymbol{s}}}\}$
$N^{\tilde{\boldsymbol{v}}} \leftarrow \lceil \frac{\beta_{\tilde{\boldsymbol{v}}} - \alpha_{\tilde{\boldsymbol{v}}}}{\Delta u} \rceil + 1$
$T^{\tilde{\boldsymbol{v}}} \leftarrow \{t_j^{\tilde{\boldsymbol{v}}} = \alpha_{\tilde{\boldsymbol{v}}} + \frac{\beta_{\tilde{\boldsymbol{v}}} - \alpha_{\tilde{\boldsymbol{v}}}}{N^{\tilde{\boldsymbol{v}}} - 1} j \mid 0 \leq j < N^{\tilde{\boldsymbol{v}}}\}$
$P^{\tilde{\boldsymbol{v}}} \leftarrow \{\tilde{\boldsymbol{p}}_j^{\tilde{\boldsymbol{v}}} = \tilde{\boldsymbol{v}}(t_j^{\tilde{\boldsymbol{v}}}) \mid 0 \leq j < N^{\tilde{\boldsymbol{v}}}\}$
$U(i,j) \leftarrow \sup_{\boldsymbol{w} \in \mathbb{E}^2} (\mu_{\tilde{\boldsymbol{p}}_i^{\tilde{\boldsymbol{s}}}}(\boldsymbol{w}) \wedge \mu_{\tilde{\boldsymbol{p}}_j^{\tilde{\boldsymbol{v}}}}(\boldsymbol{w}))$ $(0 \leq i < N^{\tilde{\boldsymbol{s}}}, 0 \leq j < N^{\tilde{\boldsymbol{v}}})$
$X \leftarrow \{(i,j) \mid i = N^{\tilde{\boldsymbol{s}}} - 1, 0 \leq j < N^{\tilde{\boldsymbol{v}}}\}$
$Y \leftarrow \{(i,j) \mid 0 \leq i < N^{\tilde{\boldsymbol{s}}}, j = N^{\tilde{\boldsymbol{v}}} - 1\}$
$R_1 \leftarrow \{\mathrm{Find}_{r_1}(U,i,j) \mid (i,j) \in X \cup Y\}$ ▷ See Fig. B.2.
$r_1 \leftarrow \arg\max_{r \in R_1} f_\delta(r)$
if r_1 is empty **then**
    $P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}) \leftarrow 0$
    return $(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, \emptyset, \{\}, N^{\tilde{\boldsymbol{s}}}, T^{\tilde{\boldsymbol{s}}}, P^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, T^{\tilde{\boldsymbol{v}}}, P^{\tilde{\boldsymbol{v}}})$
$Q' \leftarrow \mathrm{Collect}_Q^0(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r_1) \cup \mathrm{Collect}_Q^1(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r_1)$ ▷ See Fig. B.3 and B.5.
$R_2 \leftarrow \{\mathrm{Find}_{r_2}(U,Q',i,j) \mid (i,j) \in X \cup Y\}$ ▷ See Fig. B.7.
$r_2 \leftarrow \arg\max_{r \in R_2} f_\pi(U,r)$
$P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}) \leftarrow f_\pi(U,r_2)$
$R_3 \leftarrow \{\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}),U,Q',i,j) \mid (i,j) \in X \cup Y\}$ ▷ See Fig. B.8.
$r_3 \leftarrow \arg\max_{r \in R_3} f_\delta(r)$
$Q \leftarrow \mathrm{Collect}_Q^0(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r_3) \cup \mathrm{Collect}_Q^1(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r_3)$
return $(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, r_3, N^{\tilde{\boldsymbol{s}}}, T^{\tilde{\boldsymbol{s}}}, P^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, T^{\tilde{\boldsymbol{v}}}, P^{\tilde{\boldsymbol{v}}})$

図 B.1: EvaluateOverlapping$(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{v}})$ のプロセス

if $U(i,j) = 0$ **then**
    return $\{\}$
if $i = 0$ and $j = 0$ **then**
    $R' \leftarrow \{\{\}\}$
else if $i = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_1}(U,i,j-1), \{\}\}$
else if $j = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_1}(U,i-1,j), \{\}\}$
else
    $R' \leftarrow \{\mathrm{Find}_{r_1}(U,i-1,j-1), \mathrm{Find}_{r_1}(U,i-1,j), \mathrm{Find}_{r_1}(U,i,j-1)\}$
    $R' \leftarrow \{r' \mid r' \in R', r' \text{ is not empty}\}$
    if $R' = \emptyset$ **then**
        return $\{\}$
$R \leftarrow \{r = \mathrm{Append}(r', (i,j)) \mid r' \in R'\}$
return $\arg\max_{r \in R} f_\delta(r)$

図 B.2: $\mathrm{Find}_{r_1}(U,i,j)$ のプロセス

以下のように定義される.

$$f_\pi(U,r) = \begin{cases} \bigwedge_{(i,j) \in r} U(i,j) \ (r \text{ is not empty}) \\ 0 \ (\text{otherwise}) \end{cases} \tag{B.2}$$

関数 $\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i, j)$ は図 B.8 に示すプロセスで定義される.

$r' \leftarrow r$
$(i, j) \leftarrow \mathrm{First}(r)$
for $k \leftarrow j - 1$ to 0 **do**
    if $U(i, k) = 0$ **then**
        break
    $r' \leftarrow \mathrm{Prepend}(r', (i, k))$
$(i, j) \leftarrow \mathrm{Last}(r)$
for $k \leftarrow i + 1$ to $N^{\tilde{\boldsymbol{s}}} - 1$ **do**
    if $U(k, j) = 0$ **then**
        break
    $r' \leftarrow \mathrm{Append}(r', (k, j))$
return $\{(i, j) \mid 0 \leq i < N^{\tilde{\boldsymbol{s}}}, 0 \leq j < N^{\tilde{\boldsymbol{v}}}, \mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r', i, j) = 1\}$ ▷ See Fig. A.4.

図 B.3: $\mathrm{Collect}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r)$ のプロセス

if $i < 0$ or $j \geq N^{\tilde{\boldsymbol{v}}}$ **then**
    return 0
if $U(i, j) = 0$ **then**
    return 0
if $(i, j) \in r$ or $(i-1, j) \in r$ or $(i, j+1) \in r$ or $(i-1, j+1) \in r$ **then**
    return 1
return $\mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i - 1, j) \wedge \mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j + 1) \wedge \mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i - 1, j + 1)$

図 B.4: $\mathrm{IsMember}^0_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j)$ のプロセス

$r' \leftarrow r$
$(i, j) \leftarrow \mathrm{First}(r)$
for $k \leftarrow i - 1$ to 0 **do**
    if $U(k, j) = 0$ **then**
        break
    $r' \leftarrow \mathrm{Prepend}(r', (k, j))$
$(i, j) \leftarrow \mathrm{Last}(r)$
for $k \leftarrow j + 1$ to $N^{\tilde{\boldsymbol{v}}} - 1$ **do**
    if $U(i, k) = 0$ **then**
        break
    $r' \leftarrow \mathrm{Append}(r', (i, k))$
return $\{(i, j) \mid 0 \leq i < N^{\tilde{\boldsymbol{s}}}, 0 \leq j < N^{\tilde{\boldsymbol{v}}}, \mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r', i, j) = 1\}$ ▷ See Fig. A.6.

図 B.5: $\mathrm{Collect}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r)$ のプロセス

if $i \geq N^{\tilde{\boldsymbol{s}}}$ or $j < 0$ **then**
    return 0
if $U(i,j) = 0$ **then**
    return 0
if $(i,j) \in r$ or $(i+1,j) \in r$ or $(i,j-1) \in r$ or $(i+1,j-1) \in r$ **then**
    return 1
return $\mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i+1, j) \wedge \mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j-1) \wedge \mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i+1, j-1)$

図 B.6: $\mathrm{IsMember}^1_Q(U, N^{\tilde{\boldsymbol{s}}}, N^{\tilde{\boldsymbol{v}}}, r, i, j)$ のプロセス

if $(i,j) \notin Q$ **then**
    return $\{\}$
if $i = 0$ and $j = 0$ **then**
    $R' \leftarrow \{\{\}\}$
else if $i = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_2}(U, Q, i, j-1), \{\}\}$
else if $j = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_2}(U, Q, i-1, j), \{\}\}$
else
    $R' \leftarrow \{\mathrm{Find}_{r_2}(U, Q, i-1, j-1), \mathrm{Find}_{r_2}(U, Q, i-1, j), \mathrm{Find}_{r_2}(U, Q, i, j-1)\}$
    $R' \leftarrow \{r' \mid r' \in R', r' \text{ is not empty}\}$
    if $R' = \emptyset$ **then**
        return $\{\}$
$R \leftarrow \{r = \mathrm{Append}(r', (i,j)) \mid r' \in R'\}$
return $\arg\max_{r \in R} f_\pi(U, r)$

図 B.7: $\mathrm{Find}_{r_2}(U, Q, i, j)$ のプロセス

if $U(i,j) \leq P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}})$ or $(i,j) \notin Q$ **then**
    return $\{\}$
if $i = 0$ and $j = 0$ **then**
    $R' \leftarrow \{\{\}\}$
else if $i = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i, j-1), \{\}\}$
else if $j = 0$ **then**
    $R' \leftarrow \{\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i-1, j), \{\}\}$
else
    $R' \leftarrow \{\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i-1, j-1), \mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i-1, j), \mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i, j-1)\}$
    $R' \leftarrow \{r' \mid r' \in R', r' \text{ is not empty}\}$
    if $R' = \emptyset$ **then**
        return $\{\}$
$R \leftarrow \{r = \mathrm{Append}(r', (i,j)) \mid r' \in R'\}$
return $\arg\max_{r \in R} f_\delta(r)$

図 B.8: $\mathrm{Find}_{r_3}(P_{\tilde{\boldsymbol{s}}}(\tilde{\boldsymbol{v}}), U, Q, i, j)$ のプロセス

付録 C

$\mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ の実装

[25] に基づいて，図 C.1 に示すように $\mathrm{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ を実装した．ここで，λ は [**?**] で B_r として表される融合比率である．関数 GenerateFSC(D) は以下のように定義される．

関数 C. 1　GenerateFSC(D) は点，時刻，重み係数のタプルの集合 D を引数としてとり，FSC 生成法によって生成される FSC を返す関数である．

$(P_{\tilde{\boldsymbol{b}}}(\tilde{\boldsymbol{o}}), U, Q, r, N^{\tilde{\boldsymbol{b}}}, T^{\tilde{\boldsymbol{b}}}, P^{\tilde{\boldsymbol{b}}}, N^{\tilde{\boldsymbol{o}}}, T^{\tilde{\boldsymbol{o}}}, P^{\tilde{\boldsymbol{o}}}) \leftarrow \text{EvaluateOverlapping}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$
$D \leftarrow \{((1-\lambda)\tilde{\boldsymbol{p}}_i^{\tilde{\boldsymbol{b}}} + \lambda\tilde{\boldsymbol{p}}_j^{\tilde{\boldsymbol{o}}}, (1-\lambda)t_i^{\tilde{\boldsymbol{b}}} + \lambda t_j^{\tilde{\boldsymbol{o}}}, U(i,j)) \mid (i,j) \in Q\}$
$(i_0, j_0) \leftarrow \text{First}(r)$
for $i \leftarrow 0$ to $i_0 - 1$ **do**
    if $(i, 0) \in Q$ **then**
        break
    $D \leftarrow D \cup \{(\tilde{\boldsymbol{p}}_i^{\tilde{\boldsymbol{b}}}, t_i^{\tilde{\boldsymbol{b}}} + \lambda(t_{j_0}^{\tilde{\boldsymbol{o}}} - t_{i_0}^{\tilde{\boldsymbol{b}}}), 1)\}$
for $j \leftarrow 0$ to $j_0 - 1$ **do**
    if $(0, j) \in Q$ **then**
        break
    $D \leftarrow D \cup \{(\tilde{\boldsymbol{p}}_j^{\tilde{\boldsymbol{o}}}, t_j^{\tilde{\boldsymbol{o}}} - (1-\lambda)(t_{j_0}^{\tilde{\boldsymbol{o}}} - t_{i_0}^{\tilde{\boldsymbol{b}}}), 1)\}$
$(i_1, j_1) \leftarrow \text{Last}(r)$
for $i \leftarrow N^{\tilde{\boldsymbol{b}}} - 1$ to $i_1 + 1$ **do**
    if $(i, N^{\tilde{\boldsymbol{o}}} - 1) \in Q$ **then**
        break
    $D \leftarrow D \cup \{(\tilde{\boldsymbol{p}}_i^{\tilde{\boldsymbol{b}}}, t_i^{\tilde{\boldsymbol{b}}} + \lambda(t_{j_1}^{\tilde{\boldsymbol{o}}} - t_{i_1}^{\tilde{\boldsymbol{b}}}), 1)\}$
for $j \leftarrow N^{\tilde{\boldsymbol{o}}} - 1$ to $j_1 + 1$ **do**
    if $(N^{\tilde{\boldsymbol{b}}} - 1, j) \in Q$ **then**
        break
    $D \leftarrow D \cup \{(\tilde{\boldsymbol{p}}_j^{\tilde{\boldsymbol{o}}}, t_j^{\tilde{\boldsymbol{o}}} - (1-\lambda)(t_{j_1}^{\tilde{\boldsymbol{o}}} - t_{i_1}^{\tilde{\boldsymbol{b}}}), 1)\}$
return $\text{GenerateFSC}(D)$

図 C.1: $\text{Blend}(\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{o}})$ のプロセス

付録 D

参考論文

D.1　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

D.2 手書き曲線同定法 FSCI に基づく n/4 円弧および n/4 楕円弧の同定

D.3 An infinite-resolution grid snapping technique based on fuzzy theory

D.4 An interactive sketch-based CAD interface realizing geometrical and topological editing across multiple objects based on fuzzy logic