

# 隣接線分構造解析法によるオンライン手書き図形入力方式

児島治彦  
NTT電気通信研究所

本稿では、オンライン手書き図形入力方式における高精度認識手法とマンマシン性に優れた編集法について述べた。図形認識法では、隣接する線分の構造に着目し、入力と並行して認識を行う「隣接線分構造解析法」を提案した。基本図形認識実験で98.8%の認識率を示し、本手法の有効性を確認した。また、編集作業の効率化と認識の支援をめざし、編集機能をシンボル化した手書き図形コマンド(30種)による対話型編集法を提案し、簡易で高速な図形コマンド認識法と文字・図形・図形コマンド識別法を示した。本方式にオンライン手書き文字認識技術を組み合わせることにより、入力・編集を一括した文字・図形混在入力法を実現できる見通しを得た。

"On-line Hand-sketches Line Figure Input System by Adjacent Strawks Structure Analysis Method" (in Japanese)

by Haruhiko KOJIMA

(NTT Electrical Communications Laboratories, Yokosuka-shi, 238-03, Japan)

This paper describes an efficient recognition method, which is called "Adjacent Strawks Structure Analysis Method (ASSAM)", and an interactive editor by hand-written figure command symbols, which symbolize some editing functions, of an on-line hand-sketches line figure input system.

ASSAM is an interactive recognition method using adjacent strawks features. Applying this method to fundamental figures, recognition rate of 98.8% was achieved.

As for the interactive editor, 30 figure command symbols are presented, and a method which distinguish them from characters and line figures and recognize them is described.

## 1. まえがき

OA, FAの普及に伴い、図形の初期入力効率化が重要な課題となってきた。日本語文書作成においても、図表作成の効率化が求められている。このため、従来の清書段階のみならず、思考段階から編集段階まで一貫した図面作成支援技術の確立が必要である。思考を阻害することなく入力と編集を一括して行える図形入力法の満足すべき最も重要な条件として、人間の作図動作との適合性があり、それに最適な図形入力法として、オンライン手書き図形入力法がある。

従来、オンライン手書き図形入力法では、特長として任意の画数・筆順で筆記でき、かつシンボル間のセグメンテーション指示を不要とするものや、タブレット<sup>1)</sup>に入力しつつある図面を逐次認識するもの<sup>2)</sup>などがあった。このうち、前者については一図面を筆記しないと認識できないため、応答性に欠ける、後者については筆記制限がある、拡張性に乏しい、などの問題があった。また、両者の特長を兼備した方式の検討も行われているが<sup>3)</sup>、実用レベルには達していない<sup>4)</sup>。

本稿では、自由手書き線図形を対象とし、応答性と拡張性に優れたオンライン手書き図形認識法の開発を目的として、隣接する線分の構造に着目した認識法（隣接線分構造解析法）を提案し、そのアルゴリズムについて述べる。本手法は、1)特徴抽出、2)セグメント化、3)識別の3段階から構成される。ここでは課題として特に、1)高精度認識手法の確立、2)修正を容易にする手法の確立を設定し、本稿でそれらの解決法を提案する。

まず、認識の高精度化については、手ぶれや変形に対して安定して認識が行えるようにするため、抽出する特徴の選択法、使用法について述べ、その有効性

を明らかにする。次に、ストローク入力と並行してセグメントを自動抽出する手法と抽出した特徴を用いた識別法について述べる。また、修正を容易にするため、図形コマンドによる編集法について述べる。図形コマンドによる編集法とは、編集操作をシンボル化した図形コマンドを手書き入力し、この図形コマンドを実時間で認識して編集を行うものであり、筆者らはこれまでにオンライン手書き漢字入力方式において文章編集に適用し、その有効性を確認している<sup>5)</sup>。また、FAX入力による原稿・図面自動編集も試みられている<sup>6),7)</sup>。本稿では特に、隣接線分構造解析法によるオンライン手書き図形認識法について、通常の図形編集機能のほかに、認識を補完する機能も持たせた図形コマンドの具備条件を明らかにするとともに、図形コマンドの認識法および図形コマンドと図形・文字情報との識別法を提案する。

## 2. 概要

ここでは、隣接線分構造解析法によるオンライン手書き図形認識法に図形コマンドによる編集法を組み合わせた図形入力・編集法を述べる。

1)人間の作図動作との適合性、2)修正の容易性、3)応答の迅速性という3つの条件を満足し、図面の種類によらず、整形のレベルから理解のレベルまで柔軟に対応できる図形入力法を実現するため、筆者らは図1に示すようなオンライン手書き図形入力方式を検討している<sup>8)</sup>。本方式では、手書き入力された図形からストローク単位で特徴を抽出し、人間の作図動作特性に基づく自動セグメント化手法を用いてセグメントを抽出した時点で認識を行い、候補を出力する。認識（識別）は直線分、円弧、円、三角形などの基本図形単位と、

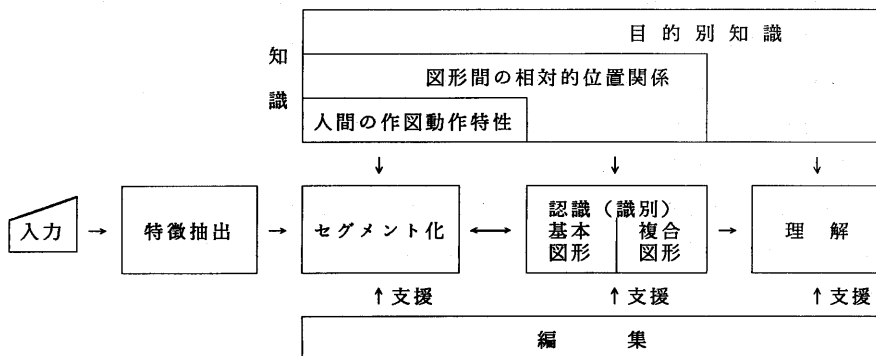


図1 処理の流れ

それらを組み合わせた複合図形単位の2段階から構成される。複合図形の認識には、基本図形間の相対的位置関係を利用する。さらに図面の理解を行う場合は、図面の種類に応じた目的別知識を利用する。また、修正の容易性を満足するため、筆者らは編集機能をシンボル化した手書き図形コマンドによる編集法を検討している。図形コマンドは、通常の図形編集機能のほか、図形認識を支援する機能も有している。

本方式の特徴として、以下の点があげられる。

[1] 知識を階層化することにより、図面の種類によらずに処理できるほか、整形のみを目的とした場合にも、理解を含めた場合にも柔軟に対応できる

[2] 編集機能をそれぞれの処理に対して用意することにより、処理のレベルに応じて知識を補完することができる

[3] 入力を重ねて行く過程で、必要に応じて処理を前に戻すことにより、処理の過程であい味さを許容しながら、最終的に正確に入力することができる

本稿では、本方式の基本となる基本図形を対象とした認識アルゴリズムについて説明する。

### 3. 認識対象

日本機械工業連合会では、図記号標準化分科会において図記号の標準化に関する検討がなされた。<sup>9)</sup>これによれば、図記号は「丸」、「線」、「弧」、「三角」、「四角」、「多角」などに大分類し、それぞれについて、たとえば「多角」であれば、五角形、六角形などに分類されている。本研究では基本図形として、直線分、三角形、四角形、五角形、六角形、円弧、扇形、円、波線の9種を対象とした。表現の違いこそあれ、ここで対象とした基本図形の集合で、日常生活で用いられている図形をすべてカバーすることができる。基本図形それぞれについて、サブカテゴリが存在する。たとえば、四角形であれば、長方形(正方形)、菱形、平行四辺形、等脚台形、直角台形がサブカテゴリである。ここでは、まず入力図形を9種に分類し、次に図形の頂点間の位置関係からサブカテゴリに分類する。

## 4. 隣接線分構造解析法

### 4.1 特徴抽出法

図形の形状を判定する基本的な特徴として、1) 頂点の個数、2) 連続する頂点間の線分の種別に着目し、ここでは、頂点の候補点として、ストロークの端点、水平・垂直方向の極大点・極小点をあげ、線分の分類

については、頂点の候補点におけるストロークの前後の角度をもとに行った。さらに、手ぶれや変形に対して安定した認識が行えるように、ペンアップ時のはねに対する処理、ノイズ対策などを付加した。

また、本処理のうち、特徴抽出部ではストロークの頂点候補点と候補点の前後におけるストロークの角度を特徴として算出し(一次特徴)、セグメント化処理部でセグメントを抽出した時点で、セグメントを構成する各ストロークの一次特徴から頂点抽出と頂点間の線分の分類を行うことにした(二次特徴)。これは、予備実験の結果、個々のストロークから頂点抽出と頂点間の線分の分類を行うよりも、それらをまとめた1つのセグメントから頂点抽出、頂点間の線分の分類を行うほうが、安定して認識が行えることが分かったからである。

## 4.2 ストロークの特徴(一次特徴)の算出

### (1) 頂点候補点の抽出

ストロークの頂点候補点として、ここでは1)ストロークの端点、2)ストロークの水平・垂直方向の極大点・極小点、3)ストロークの一部が水平線・垂直線となっている場合は、その方向の変化点を用いた。抽出した頂点候補点の例を図2に示す。

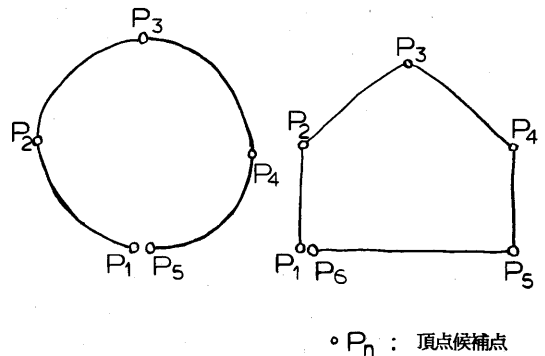


図2 頂点候補点の例

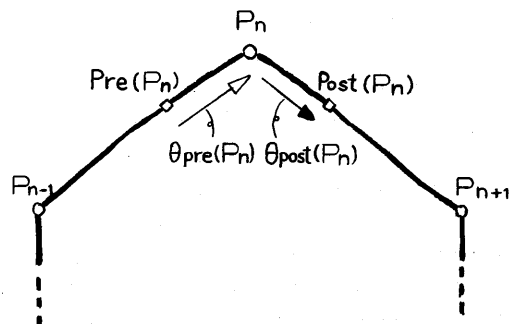


図3 特徴の算出

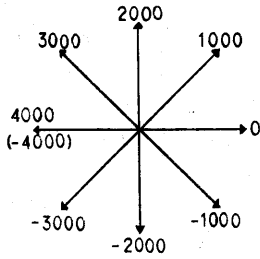


図4  $\theta_{pre}$ ,  $\theta_{post}$ の値と  
実際の方向の関係

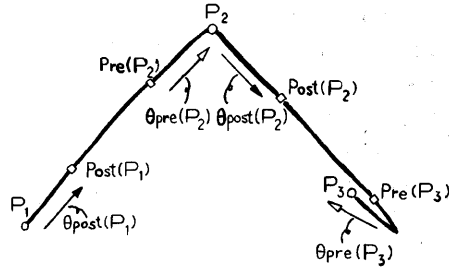


図5 ペンアップ時のはねにより  
誤った角度が検出される例

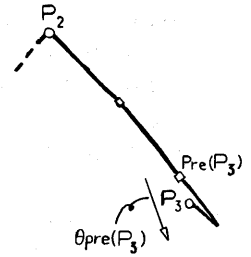


図6 はね対策により角度が  
修正された例

(2) 候補点の前後における角度の算出

頂点候補点の数をlast, n番目 ( $1 \leq n \leq \text{last}$ ) の頂点候補点を  $P_n$ ,  $P_n$ と  $P_{n-1}$ を1:2に内分する点を  $\text{Pre}(P_n)$ ,  $P_n$ と  $P_{n+1}$ を1:2に内分する点を  $\text{Post}(P_n)$ としたとき(図3参照), 頂点候補点  $P_n$  におけるストロークの前後の角度  $\theta_{pre}(P_n)$ ,  $\theta_{post}(P_n)$ は次式により求める。

[特徴の算出式]

$$\theta_{pre}(P_n) = \tan^{-1} \left( \frac{Y(P_n) - Y(\text{Pre}(P_n))}{X(P_n) - X(\text{Pre}(P_n))} \right) \times \frac{1000}{k} \quad (1)$$

$$\theta_{post}(P_n) = \tan^{-1} \left( \frac{Y(P_n) - Y(\text{Post}(P_n))}{X(P_n) - X(\text{Post}(P_n))} \right) \times \frac{1000}{k} \quad (2)$$

ただし,

$$\theta_{pre}(P_1) = 0 \quad (3)$$

$$\theta_{post}(P_{\text{last}}) = 0 \quad (4)$$

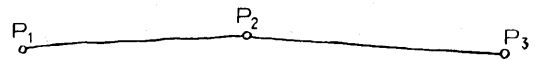
ここに,

$$k = \tan^{-1} \quad (5)$$

$\theta_{pre}$ ,  $\theta_{post}$ の値と実際の方向との関係を図4に示す。なお, 角度決定に用いる筆点については, 予備実験の結果, 安定して識別できる値として頂点候補点と隣接する頂点候補点とを1:2に内分する点とした。

しかし, ストロークの端点付近は, はねなどにより安定していないことがある。例を図5に示す。不安定な要素をもった頂点候補点において適切な角度を求めるには, 角度検出に用いる筆点の選びかたを工夫する必要がある。ここでは, はねが生じている場合, 本来のストロークの筆記方向と, はねの部分の方向とが著しく異なることに着目した。方針は, 角度を検出する頂点候補点からみて, 隣接する頂点候補点の方向と, 決定に用いようとしている筆点の方向が著しく異なっている場合は, 隣接する頂点候補点により近い筆点に変更して再び判断するものである。これにより角度が修正された例を図6に示す。

(1) 原ストローク (直線)



(2) 頂点候補点の抽出結果 (P2:ノイズ)

図7 ノイズを頂点候補点として抽出する例

(3) ノイズ除去

求めた頂点候補点のなかには本来選ばれるべきでない点(ノイズ)が含まれている場合がある。これは, 頂点候補点抽出に用いたしきい値では, 十数cm程度の長いストロークについてはノイズを拾いやすいためである。例を図7に示す。ストロークの長さに応じてしきい値の大きさを可変にする方法もあるが, 必ずしもストロークの長さとの出現率とは対応していないので, ここでは頂点候補点におけるストロークの前後の角度差  $|\theta_{pre}(P_n) - \theta_{post}(P_n)|$  が小さい点をノイズとして除去する手法を選んだ。残った頂点候補点の特徴点列  $\Sigma(X(P_n), Y(P_n), \theta_{pre}(P_n), \theta_{post}(P_n))$  をストロークの特徴点列として, セグメント化処理部へ出力する。

4.3 自動セグメント化法

図面の種類によらない知識を利用した自動セグメント化について検討するため, 人間の作図動作の解析を行った。

図8に示すフローチャート作図データ16人分を解析した結果, 次のような作図ルールが存在することが分かった。

[作図ルール]

- (1) 閉ループを1つつつ完成させながら入力する

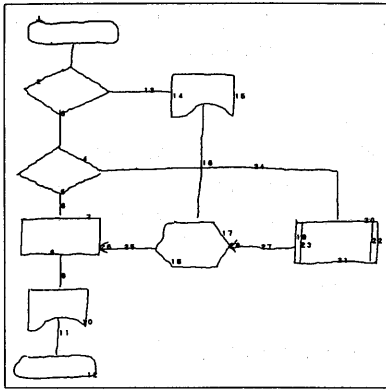


図8 フローチャート作図データ

- (2) ストロークの端点が閉ループの頂点になる
- (2-1) 閉ループをいくつかのストロークに分けて入力する場合、入力ストロークの端点のどちらかは、それまでに入力したストロークの端点の近傍にある
- (2-2) 閉ループと接続線を1本のストロークで入力する場合、それらの連結点は屈曲点で、かつその近傍に、直前、直後、または自身のストロークの端点がある

これらを実験的知識として、ストローク入力と並行して閉ループを抽出する手法を以下に述べる。

はじめに使用する語句を定義し、次に閉ループ抽出の基本式を示す。

Sc を、今、入力したストロークとしたとき、統合ストローク S<sub>c</sub> とは以下の式で定義されるものである。

$$S_c = \begin{cases} S_n & \text{(直前に閉ループが抽出された時, 初期状態)} \\ S_c + S_n & \text{(S}_n \text{の端点の近傍に (6) S}_c \text{の端点がある時)} \\ S_c & \text{(otherwise)} \end{cases}$$

さらに、S<sub>n</sub>の屈曲点を I<sub>n</sub>、S<sub>n</sub>を I<sub>n</sub>で分割してできる2本のストロークを S<sub>n1</sub>、S<sub>n2</sub>、S<sub>c</sub>が2本生成される場合の2本目の統合ストロークを S<sub>φ</sub> としたとき、閉ループ抽出の基本式は次式で示される。

$$d(T_c, T_n) < \Delta d \rightarrow S_c = S_c + S_n \quad (7.1)$$

$$d(T_c, I_n) < \Delta d \rightarrow \begin{cases} S_c = S_c + S_{n1} \\ S_\phi = S_c + S_{n2} \end{cases} \quad (7.2)$$

$$d(I_c, T_n) < \Delta d \rightarrow \begin{cases} S_c = S_{c1} + S_n \\ S_\phi = S_{c2} + S_n \end{cases} \quad (7.3)$$

$$d(T_{cs}, T_{ce}) < \Delta d \rightarrow S_c = [\text{閉ループ}] \quad (7.4)$$

式(7.1)、(7.2)、(7.4)に対応する処理例を図9に示す。

これらの式と統合ストロークの状態をもとにストロー

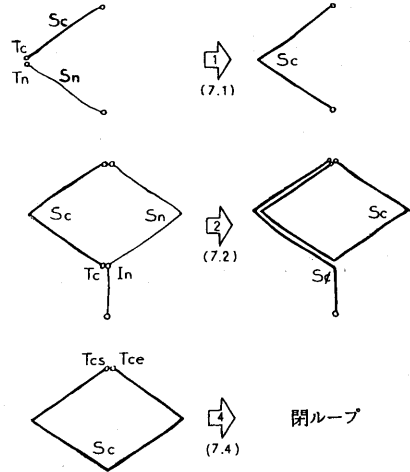


図9 閉ループ抽出の基本式説明図

ク入力と並行してセグメントを自動抽出するプロダクションルールを作成した。閉ループを抽出したときに閉ループの要素とならなかった統合ストローク、あるいは入力ストロークの一部を接続線とみなす。また統合ストロークが3本以上になる場合、3本のうち、はじめに入力したストロークを接続線とみなし、統合ストロークの数はつねに2本以下にする。1回の処理で抽出される閉ループの数は0~1個、接続線の数は0~2個である。

#### 4.4 セグメントの特徴(二次特徴)算出

以下では、頂点候補点間の線分をプリミティブと呼ぶ。セグメント化処理部で出力された特徴を、頂点単位、プリミティブ単位に集積する。

一次特徴の差を表わす関数diffを次式で定義する。

$$\text{diff}(A, B) = \min(\text{abs}(A - B), 8000 - \text{abs}(A - B)) \quad (8)$$

n番目の頂点候補点を P<sub>n</sub> としたとき、P<sub>n</sub> における頂点単位の特徴 pnt(n) は次式により求める。

$$\text{pnt}(n) = \text{diff}(\theta \text{ post}(P_n), \theta \text{ pre}(P_n)) \quad (9)$$

n番目のプリミティブ(P<sub>n</sub>とP<sub>n+1</sub>の間のプリミティブ)を Primitive(n) としたとき、Primitive(n)におけるプリミティブ単位の特徴 prm(n) は次式により求める。

$$\text{prm}(n) = \text{diff}(\theta \text{ pre}(P_{n+1}), \theta \text{ post}(P_n)) \quad (10)$$

プリミティブの種別判定には、prm(n)を用いる。

$$\text{if } \text{prm}(n) < \text{Th} \text{ then Primitive}(n) = \text{LINE} \quad (11)$$

$$\text{else Primitive}(n) = \text{ARC}$$

(Th: threshold)

例を図10に示す。

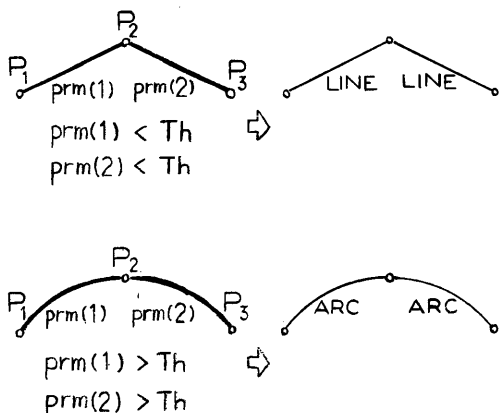


図10 プリミティブの種別判定

#### 4.5 識別処理

##### (1) 閉ループの識別

特徴点の数とプリミティブの種別をもって図形の形状を判定する。円弧プリミティブ (ARC) の数が3以上なら円, 1なら1/4円扇形, 2なら半円扇形と識別する。ただし, ARCの数が1で, 各プリミティブ及び頂点の相対的關係が1/4円扇形にふさわしくない場合, ARCとみなしたプリミティブをLINE (直線プリミティブ) に置き換える。これは不正確に入力された直線と判断する。ARCの数が2の場合も同様である。LINEの数がn個のとき, これをn角形とみなす。ただし, nが7以上の場合, 円とする。

##### (2) 分類後の各図形の識別

分類後は, 各頂点の相対的位置關係から図形の種別 (たとえば三角形であれば, 二等辺三角形か直角三角形かなど), 方向を判断する。四角形を例にとると, 四角形と判定された図形に含まれる頂点の数は4である。この図形の外接長方形を5×5分割したときに,

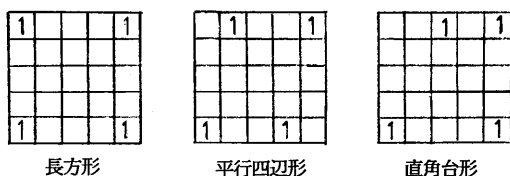


図11 特徴点座標テーブルの例

表2 各基本図形の認識率 (はね, ノイズ対策有り) [単位: %]

直線分	三角形	四角形	五角形	六角形	円弧	扇形	円・楕円	波形
99.6	99.0	98.3	94.0	95.0	99.8	95.4	99.3	99.5

各特徴点がどのブロックに属するかを調べ, 予め定められている特徴点座標テーブルと照合して四角形の種別, 方向を決定する。特徴点座標テーブルの例を図11に示す。

##### (3) 接続線の識別

接続線 (直線, 折れ線, 円弧, 波線) の識別には, プリミティブの種類, 数, 両端点におけるストロークの方向と両端点を結ぶ線分の方向を用いる。

## 5. 実験

4章で述べた特徴抽出部, セグメント化処理部, 識別処理部のシミュレーションプログラムを作成し, 基本図形認識実験, セグメント自動抽出実験を行った。

### 5.1 基本図形認識実験

今回使用した特徴の回転や変形に対する安定性, 形状判定の有効性を評価するため, 予めセグメント化してある基本図形について特徴抽出実験と認識実験を行った。使用データは, 汎用性を考慮し, 日ごろ図形を書く機会が少ない50人により筆記された4111パターンで, 筆順・画数などの制限をいっさい設けていない。対象図形は大きさや方向を4~30種ほど変化させている。

結果を表1, 2に示す。はね対策, ノイズ除去などの論理を加えたことにより, 認識率が2.5%向上した。五角形・六角形・扇形の認識率が若干低いのは, 頂点の候補点を, 変動を起こしやすい曲線成分のノイズと判断したことによる。曲線成分は個人により不正確に筆記されるので, さらに認識精度を上げるためには, 各頂点候補点間の相対的な位置關係に関する多大な知識処理が必要であり, 本アプローチでそこまでは追求しない。基本図形全体で98.8%の認識率を得て, 本特徴の安定性・有効性を確認できた。

### 5.2 セグメント自動抽出実験

自動セグメント化法の有効性を評価するため, 19

表1 基本図形認識実験結果

はね, ノイズ対策無し	96.3%
はね, ノイズ対策有り	98.8%

表3 フローチャート図形  
セグメント抽出／認識実験結果

セグメント抽出率	98.2%
認識率	97.0%

人分のフローチャート作図データを対象としてセグメント自動抽出実験と認識実験を行った。フローチャートに存在するセグメントの総数は、接続線を除いて168ボタンである。結果を表3に示す。98.2%のセグメント抽出率を得た。誤抽出した図形は、いずれも基本図形を組み合わせた複合図形である。これは、今回の手法では基本的には閉ループを対象としており、基本図形を組み合わせた複合図形については対処していないことによる。閉ループなど基本図形については、本手法が自動セグメント化手法に有効であることを確認できた。複合図形など、より複雑な形状を有するシンボルのセグメント化手法の確立が今後の課題である。

## 6. 図形コマンドによる編集法

### 6.1 図形コマンドの機能

手書き図形コマンドの機能としては、次のものがあげられる。

- [1] 一般の図形入力装置がサポートしている機能
- [2] 手書き図形認識を支援する機能

[1]については、OA、LA、CADの各分野の装置から機能を抽出した。[2]については、図形間の関係に関する情報や、位置や大きさなどに関する正確な情報を指示する機能などを抽出した。

図形コマンドの形状については、「覚えやすい、忘れにくい」ことが必要である。そのためには、

- [1] 形状が簡単であること
- [2] 日常の編集操作をシンボル化したもの
- [3] 編集操作名の頭文字の形状と組み合わせることなどの工夫やアイデアが必要である。

図形コマンドは、日常では筆記する習慣がないという点で、文字や図形と大きな違いを有する。コマンドの種類が増えると、どうしても形状を覚えにくくなる。そこで、形状の数はなるべく少なくし、同じ形状の図形コマンドであっても、筆記位置あるいは操作対象により機能を自動的に選択することにした。

以上の検討をもとに、図形コマンドの機能を抽出し、それぞれの機能に形状を割り当てた。機能および形状の一覧を表4に示す。

### 6.2 図形コマンドと図形・文字との識別

今から入力するストロークが図形であるか図形コマンドであるか、ユーザーがいちいちファンクション操作などを通じて指示しなくてもよいようにするには、入力・編集のモードレス化が必要である。入力ストロークが図形であるか図形コマンドであるか、さらには文字であるかを自動識別できる技術が必要である。

図形を編集する上で問題となるのは重畳である。基本的に文字列では重畳は起こりえないが、図形の場合は同じ位置に幾重にも重なることがある。したがって、図形編集をする上では必ずどの図形を対象とするかを指示するためのピック操作が必要である。この点に着目して図形・文字・図形コマンドを識別する手法を提案した。

図形と文字とは基本的にはストロークの長さにより識別可能である。図形入力中に文字を書き出すと、ストロークの長さが短いので、文字であると自動識別する。文字入力中に図形を書き始めた場合も同様である。一方、図形コマンドの形状は文字に準じたものであり、かつ大きさも文字とほぼ等しい。そこで、図形コマンドと文字との差異を明確化するなんらかの手法が必要である。

図形編集を行う場合、まず対象となる図形をピックした後に図形コマンドを入力する。ピック操作とは具体的には対象図形へのペンタッチ操作である。ペンタッチした位置が、ある図形の外接長方形内であれば、その図形がピックされたことになり、ブリンクもしくは高輝度表示する。いくつかの図形が重なっている場合は、ペンタッチ操作を繰返すことにより、図形を選択する。図形のピック操作を加えることにより、図形コマンドと文字の識別が可能になる。これとストロークの長さによる識別を組み合わせることにより、文字・図形・図形コマンドの自動識別が行える。

### 6.3 図形コマンド認識法

図形コマンド認識は1)形状認識、2)コマンド識別の2段階で構成される。2)は、記入された図形コマンドの入力位置から、実際にそのコマンドが何を意味するかを判断するものである。図形コマンドの形状に関する特徴として、1)種類が少ない、2)互いの形状の違いを明確にすることができる、などがある。この特徴をいかした簡易で高速な識別法として、コマンドストロークの外接長方形を2×2のメッシュに分解し、メッシュ間のストローク遷移で識別を行う手法を開発した。本手法の有効性を評価するため、表4にあげた図形コマンドのうち11種について認識実験を行った。8

表4 図形コマンド一覧

機能	形状	〈前〉 使用例 〈後〉
消去	V	
移動	m	
複写	C	
拡大縮小	S	
鏡像	^	
回転	r	
グループ化	7	 1つのセグメントとしてまとめて操作できる
平行化	→	
等長化	+	
等角化	∠	
直角化	⊥	
同心円	○	
同一線上配置	∩	
2図形の接触化	V	
接点点の平滑化	∩	

機能	形状	〈前〉 使用例 〈後〉
n角形の正規化	-	
楕円の正円化	-	
等曲率化	↷	
角とり(円弧)	∩	
円弧大きさ変換	∩	
角とり(直線)	∩	
等間隔化	=	
合同化	≡	
相似化	∝	
正n角形作成	(n)	
角度指示	n°	
均等割り付け	∩	
対称軸定義	∩	
対称点定義	.	
2等分線定義	∩	



表5 図形コマンド認識実験結果

種類	11種類
内訳	V C m S O 7 A < > - /
認識率	99.9%

人の筆者に11種の図形コマンドを各10回ずつ筆記させ、合計880個のデータを対象として実験を行った結果、表5に示すように、99.9%の認識率を得た。本手法は識別対象の増加に対して容易に拡張できる。また、形状認識したものにコマンド識別の手法を加えることにより、表4の図形コマンドをすべて扱うことが可能である。

この結果、図形コマンドによる編集法が図形認識に負担をかけずに図形編集を効率化できること、さらに文字・図形混在で、かつ入力と編集が一体化した優れたマンマシンインタフェースを有する入力法を実現できる見通しを得た。

## 7. むすび

本報告では、隣接する線分の構造に着目し、ストローク入力と並行してセグメント化および認識を行うオンライン自由手書き図形認識法である“隣接線分構造解析法”、および編集作業の効率化と認識の支援を目的とした手書き図形コマンドによる編集法について述べた。

(1) 隣接線分構造解析法については、基本図形を対象とした認識実験で98.8%の認識率、フローチャートを対象としたセグメント自動抽出実験で98.2%の抽出率を得て、用いる特徴の安定性・有効性、および図面の種類によらない知識を利用した自動セグメント化手法の有効性を確認した。

(2) 手書き図形コマンドによる編集法については、30種の編集機能を抽出し、それぞれの機能に対して図形コマンドの形状を割り当てた。簡易で高速な図形コマンド認識アルゴリズムを提案し、11種類の図形コマンドについて認識実験を行い、有効性を確認した。

(3) 入力したストロークが図形であるか、文字であるか、図形コマンドであるかを自動識別できるように設計しているので、オンライン手書き文字認識技術を組み合わせることにより、入力と編集を一括して行える文字・図形混在入力法を実現できる見通しを得た。

今後は、以下について検討する予定である。

(1) 認識対象を複合図形まで拡張するため、複合図形自動セグメント化手法を確立する

(2) 図形コマンド認識対象を拡張して評価を行い、形状や入力法の最適化を図るとともに、より高度な入力・編集環境を設計する

[謝辞]

日頃ご指導頂く小森宅内部主席研究員、酒井入力装置研究室長、小橋主幹研究員、戸井田主任研究員、ならびに有益なご助言・ご協力を頂いた関係各位に深謝します。

[参考文献]

- 1) 村瀬, 若原, 梅田: “接続ルールを導入した候補ラティス法によるオンライン手書き線図形認識”, 信学論(D), J67-D, No.3 (1984).
- 2) Kato, Iwase, Yoshida and Tanahashi: “Interactive hand-drawn diagram Input System”, PRIP(1982).
- 3) 正嶋, 葛貫, 横山, 古賀: “高度マンマシンシステム(3)~オンライン手書き図形認識方式~”, 情処学会第30回全大, 5F-3 (1985).
- 4) 御船, 正嶋, 葛貫, 横山: “オンライン手書き入力システム(2)一線分セグメント法オンライン手書き図形認識の評価一”, 情処学会第31回全大, 1N-5 (1985).
- 5) 戸井田, 児島: “オンライン手書き入力における文字修正法”, 昭58信学総全大, 2291 (1983).
- 6) 名倉, 末永: “FAXと特殊マークを用いた手書き図面の図形データ構造への変換法”, 第10回画像工学コンファレンス, 8-6 (1979).
- 7) 名倉, 末永, “文字認識機能を導入したファクシミリベースの手書き図面消書法”, 信学技報, IE81-12 (1981).
- 8) 児島, “隣接線分構造解析法によるオンライン手書き図形認識”, 情処学会第31回全大, 6G-7 (1985).
- 9) “図記号(グラフィック・シンボル)の標準化に関する調査研究(図記号標準化分科会報告)”, (社)日本機械工業連合会 (1983).