

候補ラティス法による手書きフローチャートの オンライン認識

正員 村瀬 洋[†] 正員 若原 徹[†] 正員 梅田三千雄[†]

Online Recognition Algorithm for Hand-Sketched Flowchart by Candidate Lattice Method

Hiroshi MURASE[†], Toru WAKAHARA[†] and Michio UMEDA[†],
Regular Members

あらまし 手書き線図形の認識は、手書き図表の自動清書や手書き図形による計算機との対話等、幅広い応用範囲を持つ。本論文は、手書きフローチャートを認識対象として、手書き線図形のオンライン認識手法を検討したものである。特に筆者の負担の軽減をねらい、シンボル(図形要素)間のセグメンテーションの指示を不要にし、且つ、各シンボルの画数と筆順を自由にするために、以下の3段階より構成される認識手法を提案した。(1)シンボルの形状の構造を有向グラフで表現し、グラフ上の径路の探索により、任意の画数と筆順に対処する。(2)手書き変形に対処するため、ストロークのDPマッチングを行う。(3)入力図形から候補図形を抽出し、これを候補ラティス(表形式)で表現し、候補ラティスを探索することにより、入力図形に含まれるシンボルの、セグメンテーションと認識を同時に実現する。この認識手法を120枚の手書きフローチャートに適用した結果、97.9%の認識率が得られ、本手法の有効性が確認された。

1. ま え が き

近年、文字や図形を含む手書き文書の自動清書や、手書き図形による計算機との円滑な対話を実現しようとする動きが強まっている。本論文は、この方式を実現するため、手書きフローチャートを認識対象として、自由書式で手書きされた線図形の認識アルゴリズムを検討したものである。フローチャートの認識技術は、文字認識技術と組合せることにより、フローチャートを含む手書き文書の自動清書や、手書きフローチャート入力による計算機プログラムの自動作成等、幅広く応用できる。

手書き線図形の認識アルゴリズムは大別して、既に手書きされた図面をFAXから入力し認識するオフライン型のもの^{(1)~(4)}と、タブレット上に手書きした線図形を即時に認識するオンライン型^{(5),(6)}の2つに分けられる。本論文では、後者のオンライン型の認識を扱う。従来のオンライン認識アルゴリズム⁽⁵⁾は、ストローク

の単純なマッチングを線図形認識に応用したものであり、①シンボル(図形要素)間のセグメンテーションを人間が指示する、②各シンボルを決められた画数と筆順で筆記する、等の筆記時の制約が多く、素人には扱いにくいという問題点を抱えていた。これらの制約が取り除かれれば、オンライン認識型の入力は非常に扱い易いものとなる。本論文では、①セグメンテーションの指示なし、②任意の画数と筆順、で手書きされた自由書式の線図形を認識するアルゴリズムについて検討⁽⁷⁾した。

認識アルゴリズムは次に示す3段階から構成した。

- (1) 候補図形の抽出：入力図形の内部からシンボらしい部分図形を候補図形としてすべて抽出する。その際、シンボルの形状の構造を有向グラフで表現し、そのグラフのパスを探索することにより、任意の画数と筆順に対処する。
- (2) 相異度の計算：抽出された候補図形とシンボルの形状との間で相異度を計算する。その際、手書きによる形状変形を吸収するため、DPマッチングを採用する。
- (3) 候補ラティス：抽出されたすべての候補図形に対するシンボルの名称と相異度を表(以下、この

[†] 日電公社武蔵野電気通信研究所, 武蔵野市
Musashino Electrical Communication Laboratory, N.T.T.,
Musashino-shi, 180 Japan
論文番号: 昭58-論 287 [D-81]

表を候補ラティスと吸ぶことにする。但し、候補ラティスは音声認識で使用される音韻ラティス⁽⁸⁾と同様の考え方に基づくものである)に登録する。この候補ラティスから径路を探索することにより、図形全体として最適な候補図形系列を選択する。つまり図形全体の認識をセグメンテーションと同時にを行う。

2. 認識対象と前処理

フローチャートの図形部分(以下、単にフローチャートと記述する)は、“terminal”、“decision”等のシンボル、及びシンボル間を接続する直線から構成される。今回扱うフローチャートは、図1に示す9種類のシンボルと、直線から構成されるものには設定した。この範囲で大部分のフローチャートは表現可能である。手書き情報の入力にはデータタブレットを用いる。タブレットはペンのアップ・ダウン情報とXY座標の値を一定時間間隔でサンプリングする機能を有する。ペンのn回目のダウンからアップまでの座標値系列を第nストロークと呼ぶ。タブレット入力による手書きフローチャートの一例を図2に示す。但し、図中の番号はストローク番号を表す。

サンプリングされた座標値系列に対し、前処理として、標本化および正規化の処理を行い、すべてのストロークを一定数の座標点列で表現し、かつ各ストローク内では隣り合う座標点間の距離が等しくなるようにする。

3. 辞書の記述

辞書の記述の決定に先立ち、予備実験を行なった。予備実験データは、図1に示す9種類のシンボルを20名の筆記者に、各4回ずつ(計720個)制約なしに筆記させたものである。このデータを分析した結果、画数および筆順にはほとんど規則性はみられないが、ストロークの端点となる可能性のある箇所は有限個に限

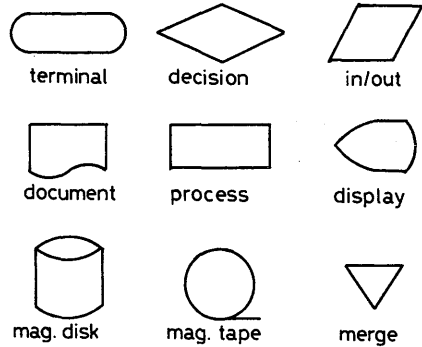


図1 シンボル
Fig. 1 - List of symbols.

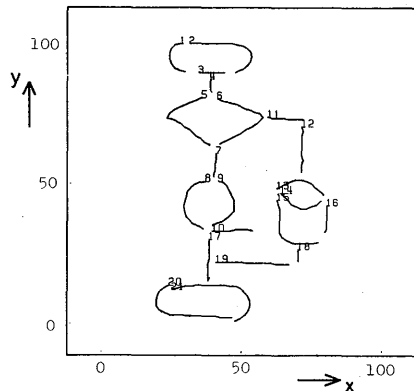


図2 フローチャート例
Fig. 2 - Example of flowchart.

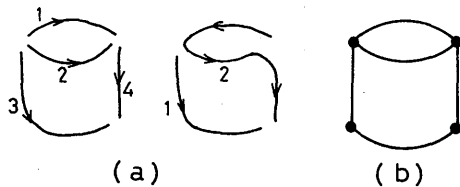


図3 (a) 画数と筆順の例, (b) 候補端点
Fig. 3 - (a) Examples of stroke order, (b) candidate end point.

表1 辞書の記述例(“mag. disk”)

ストロークラベル	種類	始点	終点	中点
A	円弧	(0, 5)	(4, 5)	(2, 6)
B	円弧	(0, 5)	(4, 5)	(2, 4)
C	円弧	(0, 1)	(4, 1)	(2, 0)
D	直線	(0, 1)	(0, 5)	
E	直線	(4, 1)	(4, 5)	

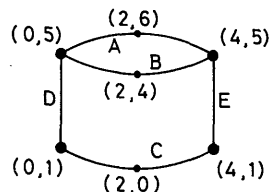


図4 辞書の記述(“mag. disk”)
Fig. 4 - Description of standard symbol(“mag. disk”).

定されるという結果が得られた。それら有限個の点を候補端点と呼ぶことにする。予備実験データにおける画数と筆順の一例を図3(a)に、シンボル“mag. disk”の候補端点を図3(b)に示す。

シンボルに関して計算機に付与する辞書は、認識対象の拡張性を考慮すれば、簡単な記述で表現できることが望ましい。予備実験の結果から、ここでは各シンボルを、候補端点を端点とするストローク群で表現し、各ストロークは直線と円弧の組合せで記述する。直線は始点と終点の座標値で、円弧は始点と終点と中間点の座標値で表現される。シンボル“mag. disk”に関する辞書の記述例を表1と図4に示す。

4. 候補図形の抽出

入力図形はシンボルと直線から構成されるフローチャートである。候補図形とは、ある種類のシンボルに対し、そのシンボルである可能性を持つ入力図形中の部分図形のことである。候補図形の抽出のために、トップダウン手法を採用する。つまり、ある種類のシンボルの存在を仮定し(以下「仮定したシンボル」と呼ぶ)、入力図形中の着目した部分図形が「仮定したシンボル」であるための条件を調べ、条件が満足される場合に、その部分図形を「仮定したシンボル」に対する候補図形とする手法である。以下、詳細を述べる。

4.1 候補端点との対応付け

入力図形中の着目した部分図形を構成するストロークを入力ストロークと呼ぶことにする。もし入力ストロークが「仮定したシンボル」であるならば、入力ストロークの端点は「仮定したシンボル」上の候補端点のどれかに対応する。そこで、「仮定したシンボル」を着目した部分図形と同じ大きさに正規化し(縦横独立にその最大値を一致させる)、その後に入力ストロークに含まれる端点をそれぞれ最近傍の候補端点に対

表2 入力ストローク端点と候補端点の対応例

ストローク番号	始点	終点
1	①	④
2	③	④

応付ける。図5(a)に示す入力ストロークを例に取れば、図5(b)の“mag. disk”が仮定された場合には、表2に示す対応結果が得られる。

手書き変形のため入力ストロークの端点の位置は変動するが、その変動は一定範囲内にある。もし入力ストロークの端点が「仮定したシンボル」のどの候補端点からも一定範囲内にはない場合には、着目した部分図形は「仮定したシンボル」ではないとし、他の部分図形を着目する処理に移行する。

4.2 候補ストローク列の生成

着目した部分図形が、「仮定したシンボル」として妥当か否かを位相構造的な観点から調べ、妥当ならば候補端点の対応結果に基づき、「仮定したシンボル」上での予想されるペンの動きを、候補ストローク列としてすべて列挙する手法について以下に述べる。

着目した部分図形、すなわち入力ストロークは N 本から構成されるとする。一方、「仮定したシンボル」は辞書の記述(3.参照)に基づいて、候補端点をノードとし、直線や曲線をブランチとする有向グラフで表現する。有向グラフを採用した理由は、ブランチ上でのペンの動く方向を陽に記述するためである。ノード名を v 、ブランチ名を a とすれば、有向グラフは計算機上で、接続行列 $D=(d_{va})$,

$$d_{va} = \begin{cases} 1 & (\text{ノード } v \text{ がブランチ } a \text{ の始点のとき}) \\ -1 & (\text{ノード } v \text{ がブランチ } a \text{ の終点のとき}) \\ 0 & (\text{その他}) \end{cases}$$

により表現できる。

例えば、“mag. disk”の有向グラフ表現は図5(c)となり、接続行列は、

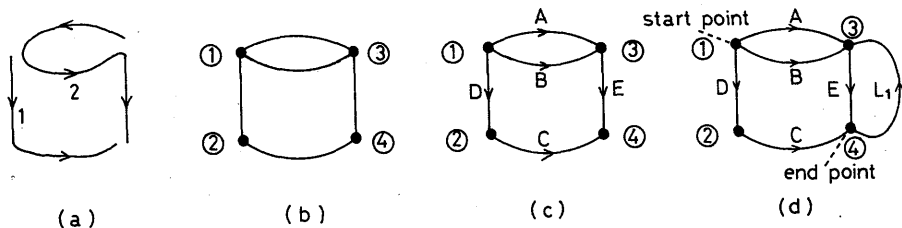


図5 (a)入力ストローク, (b)仮定したシンボル, (c)有向グラフ, (d)探索するグラフ
Fig. 5 - (a) Input strokes, (b) Supposed symbol, (c) Directed graph, (d) Graph in search.

$$\begin{matrix} & A & B & C & D & E \\ \textcircled{1} & \left[\begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 \end{array} \right] \end{matrix}$$

となる。

入力ストロークが「仮定したシンボル」として妥当であるためには、上記の有向グラフ内のすべてのブランチを1回ずつ通過し、かつ入力ストロークと対応した候補端点を始点あるいは終点とする N 本のストロークからなる径路が存在することが条件となる。すなわち処理手順としては、4.1で求めた候補端点の対応条件に制限された N 筆書きの径路を求めるグラフ問題となる。更に問題を単純化するため、引き続くストロークの間のペンアップを1つのブランチとして追加することによりオイラーパス、すなわち1筆書きの解を求める問題に変形する。以下に処理手順を示す。

- (1) 第 n ($1 \leq n \leq N-1$) ストロークの終点に対応したノードから、第 $n+1$ ストロークの始点に対応したノードに向う新しいブランチ L_n をもとの有向グラフに追加する(以下、追加ブランチと呼ぶ)。一例を図5(d)に示す。接続行列は、

$$\begin{matrix} & A & B & C & D & E & L_1 \\ \textcircled{1} & \left[\begin{array}{cccccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & -1 & 1 \end{array} \right] \end{matrix}$$

となる。

- (2) 第1ストロークの始点に対応したノードをパスの始点とし、第 N ストロークの終点に対応したノードをパスの終点とするオイラーパスを求める。解の存在のための条件は、パスの始点と終点を除くすべてのノードの次数が偶数となることである。ノード v の次数は接続行列の第 v 行ベクトルより $\sum_a |d_{va}|$ として計算できるから、ノードの偶奇条件を満足しない場合は、着目した部分図形「仮定したシンボル」ではありえないとし、他の部分図形を着目する処理に移行する。
- (3) 次数に関する条件が満足される場合には、以下に示す条件を満足するようなオイラーパスをすべて求め、候補ストローク列とする。下記の条件は入力ストローク列の順序を考慮するために生じるものである。

- ①パスが追加ブランチ L_n ($n=1, 2, \dots, N-1$) を通

過する順序は L_1, L_2, \dots, L_{N-1} である。

- ②パスが追加ブランチを引き続いて通過することを禁止する。
- ③追加ブランチの逆矢印方向へのパスの通過は禁止するが、その他のブランチでは両方向に通過可能である。

オイラーパスの探索には、プッシュダウンオートマトンの導入によりプログラム化が容易な depth first 法を採用した。探索が成功した場合には、着目した部分図形を「仮定したシンボル」の候補図形として抽出し、探索が失敗した場合には、着目した部分図形は「仮定したシンボル」ではありえないとする。

図5(d)の例に対し、上記の手法でパスを探索すれば、以下に示す6組の候補ストローク列が生成される。

- ① $+D, +C, L_1, -B, +A, +E$
- ② $+D, +C, L_1, -A, +B, +E$
- ③ $+B, +E, L_1, -A, +D, +C$
- ④ $+B, -A, +D, +C, L_1, +E$
- ⑤ $+A, +E, L_1, -B, +D, +C$
- ⑥ $+A, -B, +D, +C, L_1, +E$

但し、“ $+D$ ”はブランチ D の矢印方向への通過を意味し、“ $-B$ ”はブランチ B の逆矢印方向への通過を意味する。上記の候補ストローク列の2組目に入力ストローク(図5(a)参照)との正しい対応が現われていることがわかる。

候補ストローク列は、上記のように「仮定したシンボル」上でのブランチの系列として求まる。各ブランチは、辞書(3.)によれば直線又は円弧で記述されている。そこで、各ブランチに対し直線や円弧の座標値系列を生成し、各ストローク毎に一定の点数で近似する。この座標値系列を用いて、5.では候補ストローク列と入力ストロークとの相異度を計算する。

4.3 候補図形の抽出例

図2で示した手書きフローチャートから、“decision”記号に対する候補図形を抽出した例を示す。この場合には、“decision”記号を「仮定したシンボル」とする。まずフローチャートのある部分図形(例えば、ストローク番号13, 14から構成される部分図形)に着目し、入力ストロークの端点を候補端点に対応付ける処理(4.1参照)と、対応付け結果を基に有向グラフ(「仮定したシンボル」)上でパスを探索する処理(4.2参照)とを施し、これらの処理で矛盾が生じない場合には、着目した部分図形を候補図形とする。同様の処理を順次各部分図形に対しても行い、すべての候補図

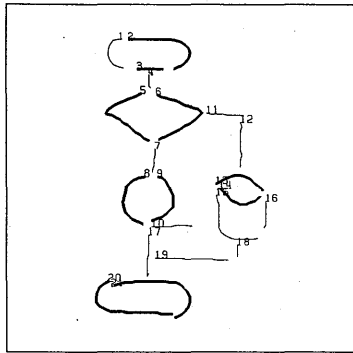


図6 “decision”シンボルに対する候補図形
Fig. 6-Candidate figures for symbol “decision”.

形を抽出する。この例では図6に示す5箇所、候補図形が抽出される。

5. 相異度の計算

候補図形の抽出段階では、位相構造（入力ストロークの端点の位置と、端点の接続関係）のみを利用し、入力ストロークの細かい形状については考慮していない。そこで次段階として、4.2で生成した候補ストローク列と入力ストロークとの間で形状のマッチングを行い、形状の相異度を計算する。

候補ストローク列と入力ストロークとの形状の相異度を表現するのに適した尺度として、次に述べる3種類の相異尺度を検討した。相異度は候補ストローク列と入力ストロークとの間で対応するストローク毎の相異度の和として定義する。従って、1ストローク対の相異度について述べる。入力ストロークと候補ストローク列との間で、対応したストロークのある1組が各々M点から構成され、両者の座標値系列が、 $\{(x_m, y_m)_{m=1, M}\}, \{(x'_m, y'_m)_{m=1, M}\}$ であるとする。但し、Mはストロークの近似点数である。

(1) 相異度1: d_1

入力ストロークと候補ストロークの各々を構成する座標点を単純に1対1に対応付ける。対応付けられた点間のユークリッド距離の和を相異度とする。つまり相異度 d_1^2 は、

$$d_1^2 = \sum_{m=1}^M \{(x_m - x'_m)^2 + (y_m - y'_m)^2\} \quad (1)$$

である。

(2) 相異度2: d_2

相異度1においてストロークを構成する座標点の対応付けにDPマッチングを採用する。つまり、相異度

表3 シンボルの認識率

	相異度1	相異度2	相異度3
認識率	95.0%	96.9%	97.3%

d_2^2 は、

$$d_2^2 = \min_u \left[\sum_{m=1}^M \{(x_m - x'_{u(m)})^2 + (y_m - y'_{u(m)})^2\} \right] \quad (2)$$

である。ここで $u(m)$ は座標点間の対応関係を表わし、 $u(m)$ には以下の制約条件を加える。

$$u(1) = 1 \quad (3)$$

$$u(M) = M \quad (4)$$

$$u(i) = j \text{ のとき } u(i+1) = \{j \text{ or } j+1 \text{ or } j+2\} \quad (5)$$

(3) 相異度3: d_3

点間のユークリッド距離に加え、各点での接線方向の差を評価したDPマッチングを採用する。つまり相異度 d_3^2 は、

$$d_3^2 = \min_u \left[\sum_{m=1}^M \{(x_m - x'_{u(m)})^2 + (y_m - y'_{u(m)})^2 + \alpha \cdot h(m, u(m))\} \right] \quad (6)$$

である。ここで $h(i, j)$ は (x_i, y_i) での接線方向と、 (x'_j, y'_j) での接線方向の差を表わし、

$$h(i, j) = \left| \tan^{-1} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) - \tan^{-1} \left(\frac{y'_{j+1} - y'_j}{x'_{j+1} - x'_j} \right) \right| \quad (7)$$

である。 α は $h(i, j)$ の係数であり、定数である。

3種類の相異度をそれぞれ用いて、予備実験データ(3で使ったデータ)を認識した際の認識率を表3に示す。実験の結果、点間のユークリッド距離とその点での接線方向の差の評価値の和をもとにDPマッチングを行う相異度3の手法で、高い認識率が得られた。以上の検討結果に基づき、本認識アルゴリズムでは相異度3を採用することにした。

6. 候補ラティス

6.1 候補ラティスの生成

候補ラティスは、入力図形中の各ストロークがどんな候補図形あるいは候補図形の一部として抽出されたかを、表形式に表現したものである。以下、候補ラティスの生成手順を示す。まず1つのシンボルに対し、入力図形中から候補図形を抽出する(4.参照)。候補図形はストローク番号列で得られるので、候補ラティス内の対応するストローク番号列の位置に、候補図形の名称(シンボル名)を登録する。更に、その候補図形に対する相異度の計算(5.参照)を行い、相異度も

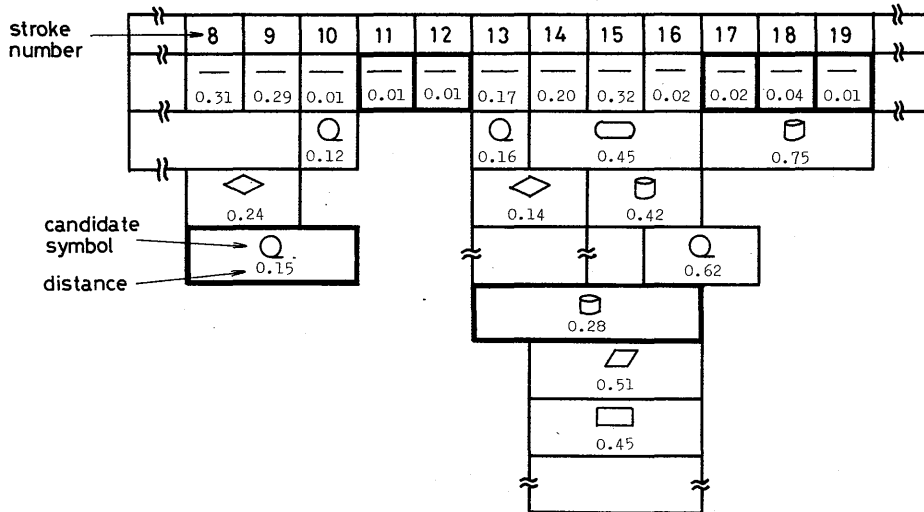


図7 候補ラティス
Fig. 7 - Candidate lattice.

合せて登録する。他のシンボルに対してもこの処理を繰り返す。直線は1本のストロークで記述されるから、入力図形内のすべてのストロークの各々について“直線”が候補図形として登録される。1例として、図2に示す入力図形に対して生成された候補ラティスの一部分を図7に示す。

候補ラティスによれば、候補図形の名称とその図形に対する相異度の値が、入力図形のストローク番号に対応して参照できる。

6.2 最適図形系列の探索

候補ラティス表現を用いて、入力図形全体をおおひ最適図形系列を決定し、入力図形中のシンボルのセグメンテーションと認識を同時に実現する。そのための手順を以下に説明する。まず候補ラティス中から候補図形を順次選択し、入力図形の第1ストロークから最終ストロークへ向う図形系列を抽出する。しかし、一般にそれらの図形系列は複数存在し、一意に定まらない。そこで次に示す評価関数を導入し、その評価関数の値を最小とする図形系列を最適な図形系列として決定する。評価関数としては2種を提案し、比較検討を行った。

(1) 評価関数1: S_1

$$S_1 = \sum_{\text{候補図形} \in \text{系列}} (\text{候補図形の相異度}) \quad (8)$$

(2) 評価関数2: S_2

$$S_1 = \sum_{\text{候補図形} \in \text{系列}} \frac{(\text{候補図形の相異度})}{(\text{候補図形のストローク数})} \quad (9)$$

評価関数1を採用した場合には、相異度の単純な算術和を最小とする径路が解となる。

一方、評価関数2を採用した場合には、相異度をストローク数で除してから算術和を計算するため、ストローク数の多い図形程、優先的に選択される。この手法を最大部分図形優先割り当て法と呼ぶことにする。

一例として、評価関数2を採用し図7の候補ラティスを探索すれば、太枠の図形系列(図7)が選択され、図8に示す認識結果が得られる。図8にはシンボル名とともに、入力図形より抽出した各シンボルの中心座標(LOCATION)と、各シンボルの縦横の大きさ(SIZE)を、図2の座標系に従って示した。

7. 認識実験

4., 5., 6.で述べた認識アルゴリズムを用いて、手書きフローチャートの認識実験を行った。認識対象は図2に一例を示す5~9個程度のシンボルと直線より構成されたフローチャートである。表4に実験に使用したフローチャートの収集データ数を示す。

認識率を表5に示す。認識率は、評価関数1を採用

表4 フローチャートデータ

筆者数	記入数	データ数
6名	5種類を各4回	120

STROKE	NAME	LOCATION	SIZE
1,2,3	TERMINAL	(39,94)	(29,10)
4	LINE		
5,6	DECISION	(40,71)	(34,17)
7	LINE		
8,9,10	MAG. TAPE	(42,41)	(24,18)
11	LINE		
12	LINE		
13,14,15,16	MAG. DISK	(71,39)	(17,23)
17	LINE		
18	LINE		
19	LINE		
20,21	TERMINAL	(36, 6)	(33,13)

図8 認識結果
Fig. 8 - Recognition result.

した場合には88.6%であるのに対し、評価関数2すなわち最大部分図形優先割り当て法を採用した場合には97.9%であった。前者の場合、誤認識11.4%の内訳は、シンボルの名称を誤ったものが1.8%、セグメンテーションが失敗したものが9.6%であった。セグメンテーションの失敗とは、例えば図9に示すように、“mag. disk”を1個の“decision”と3本の“直線”に誤った場合を指す。評価関数2を採用した場合の誤認識2.1%の内訳は、シンボルの名称を誤ったものが1.8%、セグメンテーションが失敗したものが0.3%であった。つまり、最大部分図形を優先的に割り当てることにより、セグメンテーション精度が大きく向上したことになり、評価関数2の有効性を確認できた。

ここで、認識率は、

$$\text{認識率} = \frac{\text{正しく認識されたシンボル数}}{\text{入力図形中に含まれるシンボル数}} \times 100 \quad (10)$$

で定義した。

8. 考 察

ミニコン(1 MIPS)上でのシミュレーション実験によれば、各段階での計算時間(言語はFORTRAN)は、図2の例の場合、第1段階(候補図形の抽出)に25秒、第2段階(相異度の計算)に70秒、第3段階(候補ラティスの探索)に1秒程度かかる。ここでは、処理時間の大半を占める第1、第2段階において、不要な候補図形を棄却し、処理量を低減する手法について考察する。但し、候補図形を過剰に棄却することは正しい候補図形をも棄却することになり、認識率の低下を招く。最適な棄却パラメータを選定するため、以下の2点を検討する。

①候補図形抽出(第1段階)での棄却(4.1参照)

4.1で述べたように、入力ストロークの端点と最近傍

表5 認識率

	評価関数1	評価関数2
認識率	88.6%	97.9%

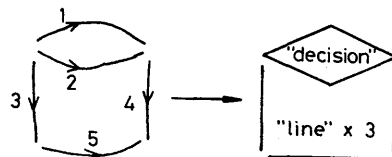


図9 セグメンテーションの失敗例
Fig. 9 - Example of segmentation error.

の候補端点との距離が、あるしきい値以内でない場合には、この候補図形を棄却する。そのしきい値を最適化するため、表4に示す120枚のフローチャートデータを用い、端点間の距離のしきい値をパラメータとして、認識率と平均候補図形数を求めた(図10(a))。但し、平均候補図形数は、入力図形中のシンボル1個に対し抽出された候補図形数の平均値である。端点間の距離は、着目した部分図形の大きさ(図形を枠む長方形の長辺の長さ)を1.0にしたときの値である。図10(a)より、認識率を下げないしきい値の範囲から、0.35を選択する。その結果、平均候補図形数は8.3個から6個に減少する。この処理により、棄却される候補図形については抽出処理を途中で打ち切ることができる。候補図形数が減るため、第2、第3段階の処理量も減少する。

②相異度計算(第2段階)での棄却(5参照)

相異度の十分大きい候補図形は候補ラティスに登録する必要はない。そこで、あるしきい値以上の相異度を持つ候補図形を棄却する。①と同様に、相異度のしきい値をパラメータとし、認識率と平均候補図形数を求めた結果が、図10(b)である。認識率を下げないしきい値の範囲から0.1を選択した。平均候補図形数は6個から2.8個に減少する。この処理により、相異度がしきい値を越えた時点で相異度計算を打ち切ることができるため、候補図形数が減り、第3段階の処理量も減少する。

以上説明した①、②の処理により、認識率を下げることなく、本方式の全処理量(計算時間)を約50%低減することが可能になった。

本方式では、処理量は入力図形中に含まれるシンボル数に比例するが、オンライン入力では筆記時間もシンボル数に比例するため、入力と同時に認識処理をすることにより、処理量の増大は問題にならない。

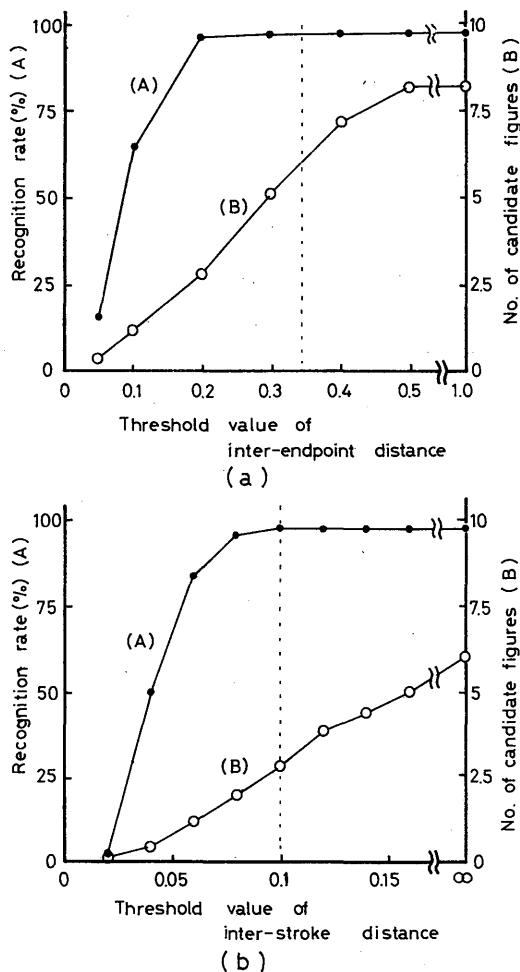


図10 候補図形抽出の改良

Fig. 10 - Improvement of candidate extraction.

9. むすび

自由書式で手書きされた線図形、特にフローチャート図形のオンライン認識を目標に、認識アルゴリズムを提案し、その性能を実験的に検討した。結果を以下に示す。

- (1) 各シンボルの形状の構造を有向グラフで表現し、有向グラフのパスを探索することにより、任意の画数と筆順に対処することが可能になった。
- (2) ストローク間の形状の相異度を計算する際、座標点間の距離に加えて接線方向の差をも考慮したDPマッチング法を提案し、この手法が優れてい

ることを実験的に示した。

- (3) 入力図形に、全体として最適なシンボル及び直線の系列を割り当てる手法として、候補ラティス法を考察し、精度の高いセグメンテーションと、各シンボルの認識を同時に実現した。120枚のフローチャート図形を対象に認識実験を行ったところ、97.9%の認識率が得られた。

本論文で提案した候補ラティス法は汎用性が高く、広く手書き線図形のオンライン認識に応用可能である。そこで、今後は本手法の認識対象をフローチャート図形に限定せず、一般線図形へ拡大してゆく予定である。その際、認識対象シンボル数の増大に伴い、候補図形数がそれに比例して増加し、処理量の増大や認識の低下が予想される。更に、今回は問題にならなかった入力図形中のシンボル数の増大も認識対象の拡大に伴い問題になる可能性がある。これらの点は今後の検討課題とする。

謝辞 日頃御指導頂く畔柳基礎研究部長、橋本統括役、増田第二研究室長に深謝します。研究を進める上で有益な議論をして頂いた基二室の諸氏に感謝致します。

文 献

- (1) 佐藤, 棟上: "手書図面処理の一方法", 昭56情報全大(22回)。
 - (2) 吉田, 榊井, 長田, 織田: "手書き図面の自動入力/処理装置", 情報処理, 22, 4, pp.300-306 (昭56-04)。
 - (3) 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(昭54)。
 - (4) Suen, C. Y. and Radhakrishnan, T.: "Recognition of Hand-drawn Flowchart", IJ CPR, pp.424-428(1976)。
 - (5) Lin, W. C. and Pun, J. H.: "Machine Recognition and Plotting of Hand-sketched Line Figures", IEEE Trans., SMC-8, pp.52-57(1978)。
 - (6) Kato, Iwase, Yoshida and Tanahashi: "Interactive Hand-drawn Diagram Input System", Conf. on PRIP, pp.544-549(1982)。
 - (7) 村瀬, 若原, 梅田: "タブレット入力による手書き線図形認識アルゴリズムの検討", 信学技報, PRL81-69(1981)。
 - (8) 鹿野, 好田: "合話音声の機械認識における言語処理", 信学論(D), J61-D, 4, pp.253-260 (昭53-04)。
- (昭和57年9月6日受付, 58年1月10日再受付)