

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

ON-LINE RECOGNITION ALGORITHM FOR HAND-SKETCHED LINE FIGURES

村瀬 洋 若原 徹 梅田三千雄
Hiroshi MURASE Toru WAKAHARA Michio UMEDA

日本電信電話公社 武蔵野電気通信研究所
Musashino Electrical Communication Laboratory, N.T.T.

In this paper, an on-line recognition algorithm for hand-sketched line figures is described. The algorithm consists of two parts. At first part, candidate figures are extracted by using topological relation among end points of input strokes, and the distance between candidate figure and input figure is calculated by DP matching method. At second part, examining the candidate lattice composed of candidate figures and their distances, the combination of candidates to minimize the total sum of distances is selected.

Applying the above algorithm to 60 handwritten flowchart data (these consist of 4-9 figures and lines), recognition rate of 97.6% was achieved.

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会話を実現しようとする動きが強ま、ている。これを達成するためには、理想的には自然紙面上に自由書式で手書きされた線図形を自動認識する技術が必要となる。

現在までに、ファクシミリ(又はスキャパー)から入力された手書き線図形の認識に関する研究の報告は幾つかなされている(文献(1)~(3))。これらの入力方式は、身近なファクシミリを入力装置として利用できる点、既に紙面上に筆記されている線図形の認識が可能である点で優れている。しかし新規に線図形を入力する場合には、一旦紙面上に筆記しその後ファクシミリ装置へ入力するため、2段階の手間となり即時性に欠ける。

これに対し、タブレットから直接入力する方式がある。例えば、CAMやCAMの入力手法の一つとして、タブレット上に手書きされた編集記号を認識する手法がある(文献(4))。ま

たタブレット上に手書きされた線図形を認識する研究の報告もある(文献(5)~(6))。これらタブレットから入力する手法は、即時性の面では有効である。しかし従来のこの種の認識手法では、

- (1) 図形要素を決められた画数・筆順で筆記する。
- (2) 図形要素間のセクメンテーション情報をシステムに指示する。

などの制限が筆記者に課せられた。そのために必ずしも手軽な入力方式とは言えなかった。簡便な入力のためには、任意の画数・筆順での手書きを許し、かつセクメンテーション情報を必要としない認識手法の実現が望まれる。

本報告では、上記制限(1),(2)を同時に取り除くタブレット入力による手書き線図形の認識アルゴリズムを提案する。その際、任意の画数・筆順に対応するため、トップダウン的に候補ストローク列を生成し、標準ストローク列としてマッチングに使用する手法を導入した。また、全体として矛盾のないセクメンテーションを実現するために、候補ラティス法⁽⁷⁾を考案した。今

回は本アルゴリズムをフローチャートの図形部分に適用し、認識実験を行なった。なお今書きフローチャートの自動認識技術は、論文・仕様書等によく出現するフローチャートの自動消書き、さらにはフローチャート入力によるプログラムの自動作成等に応用できる技術である。

以下、2章で認識対象データについて、3章で認識アルゴリズムを、4章で認識実験の結果を述べる。

2 認識対象データ

今回扱うフローチャートは、図1に示す専用図形9種類と直線から構成されるものにする。この専用図形と直線によりほとんどのフローチャートは構成可能である。

実験に使用する今書き図形データとして、専用図形データとフローチャート図形データを収集した。

専用図形データとは図1に示す専用図形を1図形ずつ、任意の函数・算項で今書きさせたデータのことである。この図形データを用いて任意の函数・算項に付与するためのアルゴリズムを検討した。

フローチャート図形データとは、4~9個程度の専用図形および直線から構成される図形を、専用図形を1つ書き終る毎に、筆記者がセクメンテーション情報を指示することなく、任意の順序で今書きした図形データのことである。その際に、各専用図形については任意の函数・算項で書くことを許した。フローチャート図形データの例を図2に示す。この図形データを使用し、自動セクメンテーション手法の検討および認識実験を行なった。

収集したデータの数を表1に示す。

3 認識アルゴリズム

3.1 認識アルゴリズムの概要

認識処理の概要を図3のフローチャートに示す。処理は大きく2つの段階から構成した。

第1段階では、入力図形中から候補図形を抽出する。その際に、函数・算項に関する制限はないことを前提に処理を行なう。本段階では入力図形中のすべての部分図形を処理の対象と

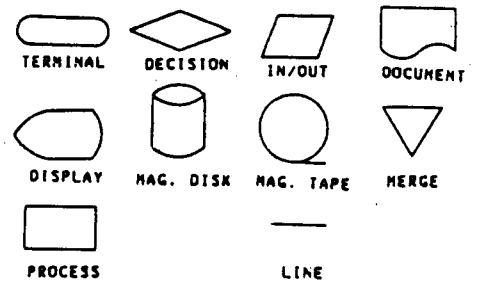


図1 専用図形と直線

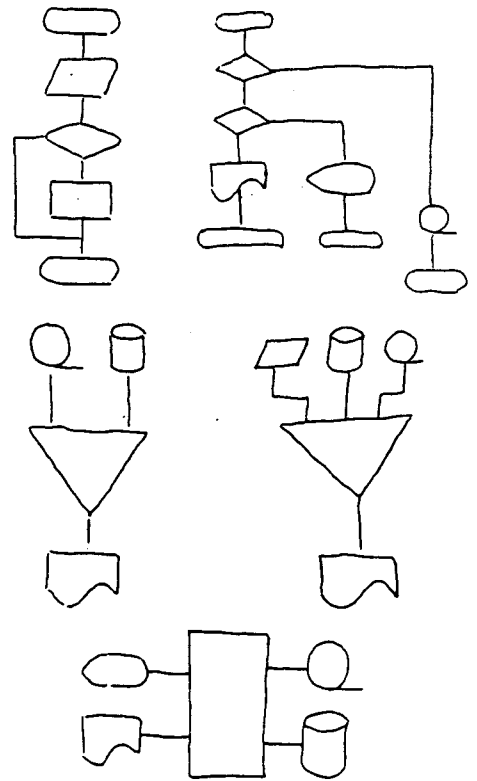


図2 フローチャート図形データ例

表1 収集データ

種類	筆記者数	記入数	データ数
専用図形データ	20名	9種類を各4回	720
フローチャート図形データ	3名	5種類を各4回	60

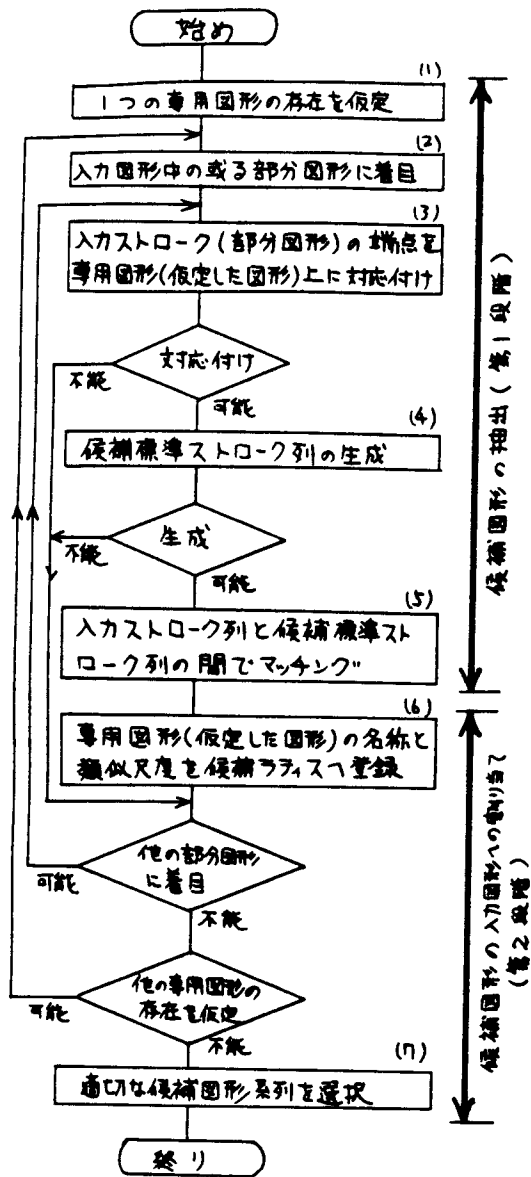


図3 認識アルゴリズム

するため、処理量の少ない簡単な特徴を利用することが有効である。そこで入力図形中のストロークの端点とその接続関係のみを用いて、候補図形を抽出する。図3中で第1段階は(1)~(4)までである。処理の詳細を3.3, 3.4, 3.5にて述べる。

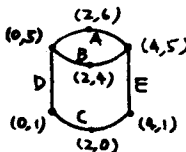
第2段階では、前段階で抽出された複数の候補図形の中から適切な候補図形系列を産出し入力図形に割り当てる。この処理の実現のため、

候補図形の名称とその図形に対する相異度(入力図形と候補図形の間の距離値)とを合せて登録する候補ラリス表現を導入する。候補ラリスから最適経路を探索することにより、入力図形全体として矛盾のない候補図形の割り当てが可能になる。図3中(6),(7)がこの処理に相当する。処理の詳細は3.7にて述べる。

3.2 辞書の記述

対象図形の種類の多様性を考慮し、専用図形に関して計算機に付与する辞書はできるだけ簡単な記述で表現する。ここでは図形を直線と円弧の組合せで記述する。直線(line)は始点と終点の、円弧(arc)は始点と終点と中間点の座標値で表現される。

専用図形"mag.dish" (0,5) (2,6) (4,5) (2,4) (0,1) (2,0) (4,1) に関する辞書の記述例を図4に示す。



ラベル	名称	始点	終点	中間点
A	ARC	(0,5)	(4,5)	(2,6)
B	ARC	(0,5)	(4,3)	(2,4)
C	ARC	(0,1)	(4,1)	(2,0)
D	LINE	(0,1)	(0,5)	
E	LINE	(4,1)	(4,5)	

図4 辞書の記述例 (mag.dish)

3.3 入力ストロークの端点の対応付け

制限を与えずに筆記させた専用図形データを分析すると、筆順および画数にはほとんど規則性は見られないが、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られるという結果が得られた。これらの点を候補端点と定義する。候補端点の例を図5に示す。

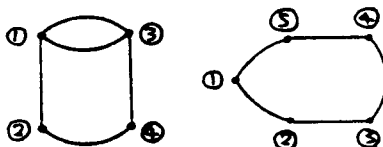


図5 候補端点の例

入力図形中で着目している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した場合、もし入力ストロークが、存在を仮定

して専用図形（以下これを「仮定した図形」と呼ぶ）であるならば、入力ストロークの端点は「仮定した図形」上の候補端点のどれかに対応する。そこでまず、「仮定した図形」を入力図形と同じ大きさで正規化し（縦方向、横方向それぞれ独立にその最大幅を正規化する）、その後に入力ストロークの各端点をそれぞれ最 Nearby の候補端点に対応付ける。例えば図 6 (a) に示す入力ストローク例に対し、図 6 (b) の "mag. disk" が仮定された場合には、図 6 (c) に示す対応結果が得られる。

手書き変形のため端点の位置は変動する。しかし、その変動は一定範囲内にある。処理の効率化を図るため、入力ストロークの端点が「仮定した図形」のどの候補端点からも一定範囲内に位置しない場合は、「仮定した図形」ではありえないとして、他の専用図形の存在を仮定する処理に移る。この端点位置の変動許容範囲については、専用図形データ（表 1 参照）を分析し、余裕ある値に設定した。

3.4 候補標準ストローク列の生成

対応付けられた候補端点とその接続関係を用い、「仮定した図形」が妥当であるか否かを位相構造的な観点から調べる。妥当ならば、「

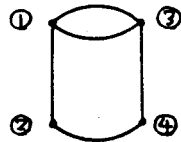
仮定した図形」上での予想されるペンの動きを、候補標準ストローク列としてすべて列挙する。

上記処理を実現するため、まず候補端点をノードとする有向グラフを用いて、「仮定した図形」を表現しておく（例えば、"mag. disk" の有向グラフ表現は図 6 (d) となる）。次に、3.3 で求められた端点の対応条件（図 6 (c) 参照）を満たし、グラフ内のすべてのアラシクを通過するパスの経路の組をすべて求める。但し P は入力ストローク数である。これらの経路の組が候補標準ストローク列となる。具体的には次の手順に従う。

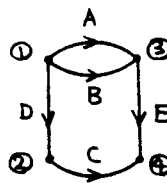
- (1) 第 k ($1 \leq k \leq P-1$) ストロークの終点に対応したノードから、第 $k+1$ ストロークの始点に対応したノードに向う新しいアラシクをもとの有向グラフに追加する（図 6 (e) 参照）。この追加により (3) で述べる条件のもとでオイラーパスを求める問題に変形できる。その際、パスの始点は第 1 ストロークの始点に対応したノードであり、パスの終点は第 P ストロークの終点に対応したノードである（図 6 (e) 参照）。
- (2) パスの始点と終点を除くすべてのノードの次数が偶数であることが解の存在のための必要



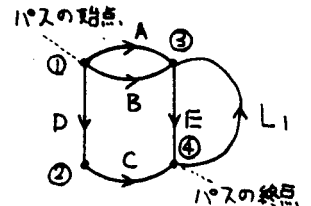
(a) 入力ストローク



(b) 仮定した図形



(d) 有向グラフ



(e) 探索するグラフ

ストロークの番号	始点	終点
1	①	④
2	③	④

(c) 入力ストロークの端点と候補端点との対応

1:	+D	+C	**	-B	+A	+E
2:	+D	+C	**	-A	+B	+E
3:	+B	+E	**	-A	+D	+C
4:	+B	-A	+D	+C	**	+E
5:	+A	+E	**	-B	+D	+C
6:	+A	-B	+D	+C	**	+E

(f) 候補標準ストローク列の生成例

但し $\left\{ \begin{array}{l} + \text{ 矢印方向へ通過} \\ - \text{ 逆矢印方向へ通過} \\ ** \text{ ペンのUP (アラシク } L_1 \text{ に対応)} \end{array} \right.$

図 6 候補標準ストローク列の生成過程

条件となる。この条件を満足しない場合は、入力図形は「仮定した図形」ではありえないとし、他の専用図形の存在を仮定する処理へ移行する。満足する場合は次の処理を行う。

(3)条件付きでオイラーパスを求める。その際必要となる条件を以下に示す。

①ブランチ L_p ($1 \leq p \leq P-1$) の逆矢印方向への通過は禁止するが、その他のブランチは両方向に通過可能である。

② L_p を通過する時は L_1, L_2, \dots, L_{P-1} である。

③ L_{p+1} を L_p に連続して通過することは禁止する。

パスの探索には Depth First 法による tree search を用いた。探索が失敗した場合には、入力図形は「仮定した図形」ではないとし、他の専用図形の存在を仮定する処理へ移行する。

本手法によれば、図6の例の場合には図6(f)に示す6個の候補標準ストローク列が生成される。図6(a)の入カストローク例に対応した候補が図6(f)の2番目に生成されていることがわかる。この例では生成数は6個であったが、表1に掲げた専用図形データの各図形に対し正しい図形の存在を仮定し、候補標準ストローク列を生成する実験を行なったところ、生成数の平均値は1.6個であった。次段階におけるマッチングの回数は候補標準ストローク列の生成数に比例する。つまり本手法により、函数・筆順が固定の場合に比較し、処理量を1.6倍に増加しただけで任意の函数・筆順を許容することができた。

候補標準ストローク列は記号列で求まるので、辞書に含まれる専用図形の記述(図2参照)を用い、記号列を座標値系列に変換し、次段階の処理で使用する。

3.5 ストローク間のマッチング

候補標準ストローク列の生成段階では、ストロークの端点の位置と接続関係だけを適用し、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、候補標準ストローク列と入カストローク列間で形状のマッチングを行ない、ストローク形状の相異度を訂正する。

ここでは点近似されたストロークに対する種類のマッチング手法を適用し、専用図形データを対象に認識実験を行なうことにより相異尺度を評価した。

(1)手法1

入カストロークと候補標準ストロークを各々N点で点近似し、両者のストローク間で単純にN点を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総和を相異度とする(図7-(a)の説明図を参照)。

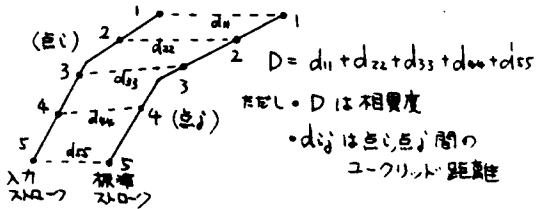


図7-(a) 説明図(手法1)

(2)手法2

入カストロークと候補標準ストロークを各々N点で点近似する。その点の対応付けにDPマッチングを利用する。つまり適当に点の対応をすすることにより各々の距離値の総和を最小とする点の対応付けを行なう。但し各点間の距離にユークリッド距離を用い、距離値の総和を相異度とする(図7-(b)の説明図を参照)。

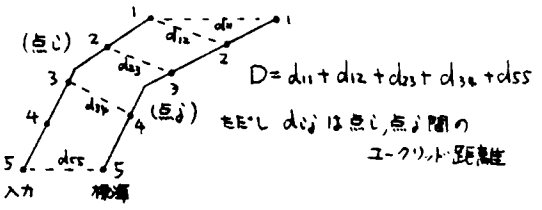


図7-(b) 説明図(手法2, 手法3)

(3)手法3

手法2と同様にDPマッチングにより点間の対応をとる。但し、各々の点間の距離にはユークリッド距離と、その点での接線方向に関する距離の加重和をとる。各点での距離値の総和を相異度とする。図7-(b)の例において $d_{ij} = (\text{点}i, j \text{間のユークリッド距離} + \text{傾角} \times (\text{点}i \text{での傾き角} - \text{点}j \text{での傾き角}))$ としたものである。

各手法を用いて専用図形データを認識した際の認識率を表2に示す。

表2 専用図形データの認識率

手法	認識率
1	95.0%
2	96.9%
3	97.3%

実験の結果、手返りの点間のユークリッド距離とその点での接線方向に関する距離の和をもとにDPマッチングする手法で最も高い認識率が得られたので、この手法を本アルゴリズムに採用することにした。なお誤検パターンの一例を図8に示すが人間が見ても判別が困難な図形である。

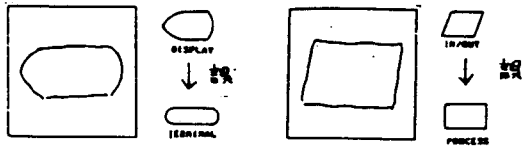
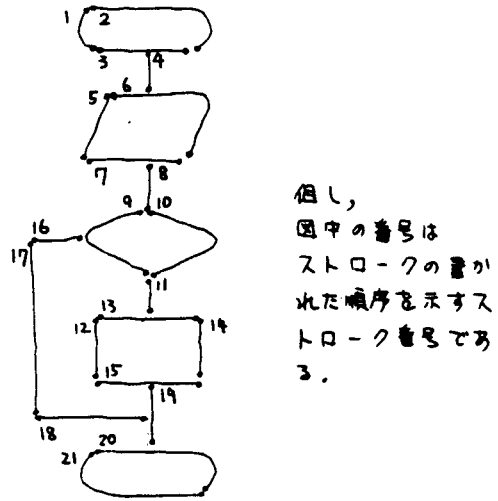


図8 誤検パターンの一例

3.6 候補図形の抽出

図2で示したようなフローチャート図形データが本報告での最終的な認識対象である。図形データごとに、図形要素の書かれた順序、図形要素内での画数や筆順は多様である。その一例として図9に21ストロークで書かれたフローチャートを示し、以下この図形を例に説明する。

3.3 ~ 3.5で示したアルゴリズムにより入力図形から候補図形を抽出し、各候補図形について相異度を計算した。仮に画数や筆順についてあらゆる組合せを考慮し、総合的な手法を用いると膨大な数の相異度の計算が必要であるのに対し、本手法では選抜された候補機率ストローク列の数が相異度を計算すればよい。図9に示すフローチャートの例では、総合的な手法を用いると約41万回の相異度の計算が必要であるのに対し、本手法では67回の相異度の計算だけで必要とした。今回は相異度の計算にDPマッチングを基本とする手法を用いたが、このように処理量の多い相異度を導入する



但し、図中の番号はストロークの書かれた順序を示すストローク番号である。

図9 フローチャート図形の一例

際、特に有効である。

3.7 候補ラティス

3.6で抽出した候補図形を入力ストローク全体に対して最適に割り当てたため、候補図形を表現形式のラティスで表現する。図9に示すフローチャートの入力例に対するラティス表現の例を図10に示す。ラティス中の各ブロックの中には、候補図形の名前とその候補図形であるとしたときの相異度が記入されている。

候補ラティスによれば候補図形系列の選択の仕方に対応して、第1ストロークから最終ストロークまでの間に経路が考えられる。多数の経路の中で以下に示す評価式の値を最小にする経路を探索する。探索の際には最短経路を探索するアルゴリズムを使用した。2種類の評価式(S_1, S_2)を用い、比較検討を行った。

(1)評価式1

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

(2)評価式2

$$S_2 = \frac{\sum_{\text{候補図形} \in \text{系列}} (\text{相異度})}{(\text{候補図形のストローク数})}$$

評価式1を用いた場合、相異度の単純な算術和を最小にする経路が解となる。この評価式

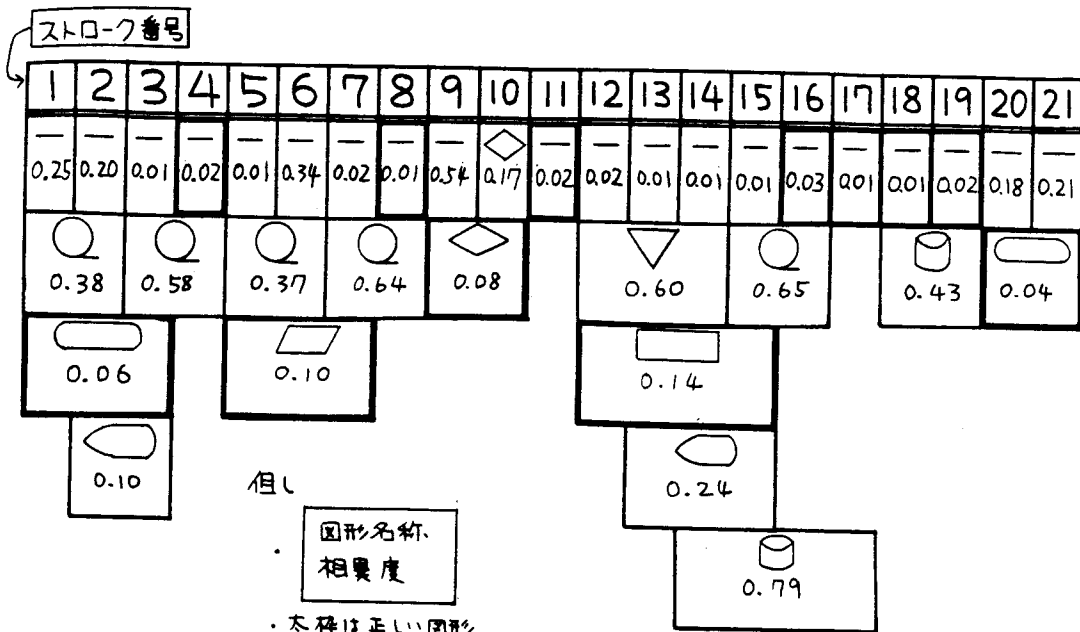


図 10 候補ラティスの例 (入力図形は図9)

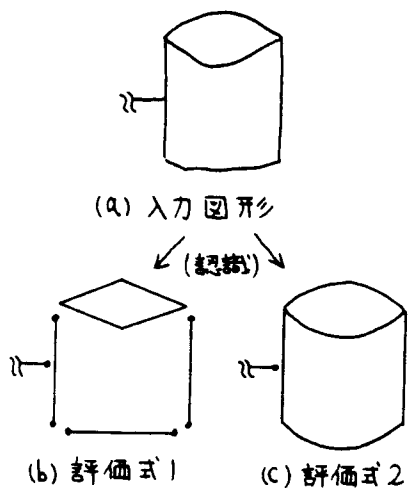


図 11 救済できた誤割り当ての例

で認識を行なったところ、図 11 (b) に示す例 ("mag. disk" を "decision" + 3本の "line" に割り当て) のような割り当て失敗の例が多数発生した。

一方、評価式2では相異度をストローク数で重み付けてから筆算和をとる。そのためスト

ローク数の多い候補図形程、優先的に割り当てがなされる。その結果、図 11 に示したような割り当て失敗の大部分を救済することができた。この手法を最大候補図形優先割り当て法と呼ぶことにする。本アルゴリズムではこの手法を採用する。

今回提案したアルゴリズムにより図9のフローチャートを認識した結果を図12に示す。

ストローク番号	図形名称
1 -> 3	TERMINAL
4 -> 4	LINE
5 -> 7	IN/OUT
8 -> 8	LINE
9 -> 10	DECISION
11 -> 11	LINE
12 -> 15	PROCESS
16 -> 16	LINE
17 -> 17	LINE
18 -> 18	LINE
19 -> 19	LINE
20 -> 21	TERMINAL

図 12 認識結果の例 (入力図形は図9)

5. 認識結果

図2に示したような、フローチャート図形データ(表1参照)60枚を対象に認識実験を行なった。認識率は最大部分図形優先割り当て法を使用しない場合には88.6%であるのに対し、本手法では97.6%であった。誤認識率2.4%の内訳は、専用図形か他の専用図形に誤認識された例2.0%、候補図形の入力図形への割り当て(セクメンテーション)が失敗した例0.4%である。

但し、

認識率 = $\frac{\text{正しく認識された専用図形数}}{\text{入力図形中に含まれる専用図形数}}$ で定義した。

6. あとがき

トップダウン的な候補標準ストローク列の生成手法を導入したことにより、マッチングの処理量を増加することなく(3.4参照)、任意の画数・筆順を許容することが可能となると同時に、フローチャート図形全体の認識においても、不必要なマッチング処理をすることなく(3.6参照)、候補図形を抽出できる見通しが得られた。

一方、候補ラティス法を提案し、最大部分図形優先割り当て法を導入したことにより、精度の高い図形割り当て(セクメンテーションの自動化)が実現できた。

各専用図形は任意の画数・筆順で書かれ、かつ専用図形(直線を含む)間にセクメンテーション情報のないフローチャート図形60枚を対象に97.6%の認識率が得られた。

今後の研究課題としては、(1)個々の専用図形の認識精度を更に向上させる、(2)何度か重ね書きした線分により構成された線図形(ラフスケッチの際によく用いる)の認識を可能にする、(3)文字や図形が複雑に混在する線図形や複雑のより複雑な図形の認識に本手法を拡張することなどがあげられる。

<謝辞>

日頃御指導頂く、畔柳基礎研究部長、猪本統祐役、増田第八研究室長に感謝します。また日頃御討論頂く第八研究室の諸氏ならびに、データ収集に御協力頂いた諸氏に感謝します。

<文献>

- (1) 佐藤, 禎上: "手書き図面処理の一方法" 情報処理学会, 第22回全国大会, 1981
- (2) 吉田, 柳井, 長田, 織田: "手書き図面の自動入力/処理装置", 情報処理学会, vol.22, No.4, 1981
- (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回画像工学会コンファレンス, 8-6, 1979
- (4) "ターンキー型CAD/CAMシステムの研究", 図形と画像, vol.2, No.3, 1981
- (5) Gabriel F. Groner: "Real Time Recognition of Hand Printed Text", FJCC, 1966
- (6) Wen C. Lin: "Machine Recognition and Plotting of Hand-Sketched Line Figures", IEEE, Trans. of SMC-8, 1978
- (7) 鹿野, 巧田: "全話音声の機械認識における言語処理", 信学論(カ), J61-D, 4, 1978