

# 接続ルールを考慮した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

ONLINE HAND-SKETCHED FLOWCHART RECOGNITION  
BY CANDIDATE LATTICE METHOD WITH CONNECTION RULES

村瀬 洋 若原 徹 梅田 三千雄

Hiroshi MURASE Toru WAKAHARA Michio UMEMA

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.

This paper describes an online hand-sketches flowchart recognition algorithm. The algorithm consists of three parts. First, candidate symbols are extracted from an input chart by using topological features, and distances from reference patterns are calculated. Second, symbols which minimize the total sum of distances are selected out of candidates as a tentative recognition result. Third, connective relation between selected symbols is examined and illegal symbols are replaced with other candidates which are consistent with connection rules. Applying this algorithm to 100 hand-sketches flowchart data (each contains 5-10 symbols and lines), recognition rate of 95.9% is achieved.

## 1 まえがき

近年の文書処理の発展に伴い、手書き図形・文字の自動消書や、手書き図形による機械との円滑な対話の実現が望まれている。これに関連して、手書き図形の認識が必要となっている。現在までに手書き線図形の認識に関する研究としては、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認識法<sup>(1),(2),(3)</sup>と、タブレット上に手書きした線図形を即座に認識するオンライン型の認識法<sup>(4),(5)</sup>についての報告がある。

本報告では、後者のオンライン型の認識を扱う。オンライン認識の特長は、認識結果を即時に確認でき修正もその場で行える点や、入力と編集とを融合して対話的に行える点にある。しかし、従来のオンライン認識法では、筆記時の制限が強く、素人には使いが困難であった。筆者らは既に、①シンボル(図形要素)毎にセグメンテーション情報をシステムに指示する、②各シンボルを定められた画数と筆順で筆記する、という筆記制限を取り除く認識アルゴリズム

ムとして、候補ラティス法<sup>(6)</sup>を提案した。候補ラティス法は、入力図形中から対象とするシンボルの候補図形を抽出し、候補ラティスと呼ぶ表に登録する段階と、候補ラティスの中から矛盾のないシンボル系列を選択し、それを認識結果とするという2段階から構成された手法である。

本報告では、前回報告<sup>(6)</sup>から認識対象を拡大し、JIS 6270<sup>(7)</sup>で定められたシンボルより構成されるフローチャートに、上述の候補ラティス法を適用し、まず問題点の抽出と分析を行う。次に、それに対処するために、候補ラティス法の改良について述べる。更に、認識結果を整形し出力する自動消書法についても述べる。

前回報告の候補ラティス法の問題点として今回、以下に示す2点が確認された。

- (1) 候補図形抽出の誤り。
  - (2) 包含関係にあるシンボル間での誤読や、形状の類似したシンボル間での誤読。
- 問題点(1)は、入力ストロークの端点とシンボルの候補端点間での対応誤りが原因であった。これに対処するため、対応誤りの可能性のある

シンボルの候補軸点群を見かけ上1つの点として扱う手法を導入した。一方、問題点(2)に対処するため、フローチャートに現れるシンボル間の接続ルールを上位概念として使用した。候補ラティスの規模による認識結果がこの接続ルールに矛盾する場合には、候補ラティスを修正し、その後、再認識しなおす手法を導入した。

上記の2手法を新たに導入した候補ラティス法を用いて、100枚のフローチャートを対象に認識実験を行った結果、95.9%の認識率が得られた。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JIS C6270で定められたシンボルから構成されたものである。但し、シンボル中「Auxiliary operation」は形状的に「Process」の一変形と考えられるので、同一カテゴリ「Process」として扱う。又、「Flow line」はシンボル間を接続するという特殊な意味を持つので別に扱う。従って、ここでの認識対象は28種類のシンボルと直線から構成されるフローチャートである。

認識実験に用いるフローチャートデータを以下の機に収集した。各フローチャートは6~10個程度のシンボルと直線から構成されるものとする。但し、筆記には、①1つのシンボルは連続して置くものの、システムにメッセージを通知する必要がある、②各シンボルの画

数・筆順は自由である、を指示した。フローチャートデータの収集数を表1に示す。収集したフローチャートデータの一例を図2に示す。

表1 フローチャートデータ

筆記者数	フローチャートの種類	筆記数	合計
5名	5種類	各4回	100枚

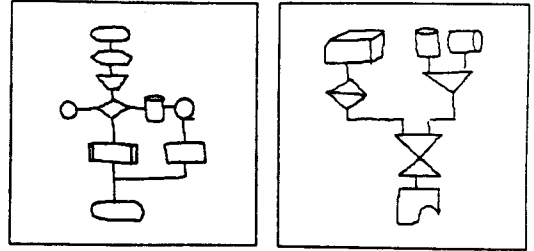


図2 フローチャートデータ例

## 3 認識アルゴリズム

### 3.1 認識アルゴリズムの概要

認識アルゴリズムの概要を図3の流氷図に示す。図中、\*印の箇所が今回改良した処理を、井印の箇所が今回新たに導入した処理を示す。以下、図3の流氷図に沿って説明する。

### 3.2 手書き図形入力

手書き情報の入力にはデータタブレットを使用する。タブレットはパンのアップダウン情報とXY座標の値を一定時間間隔でセンシングする機能を有する。パンのダウンからアップ

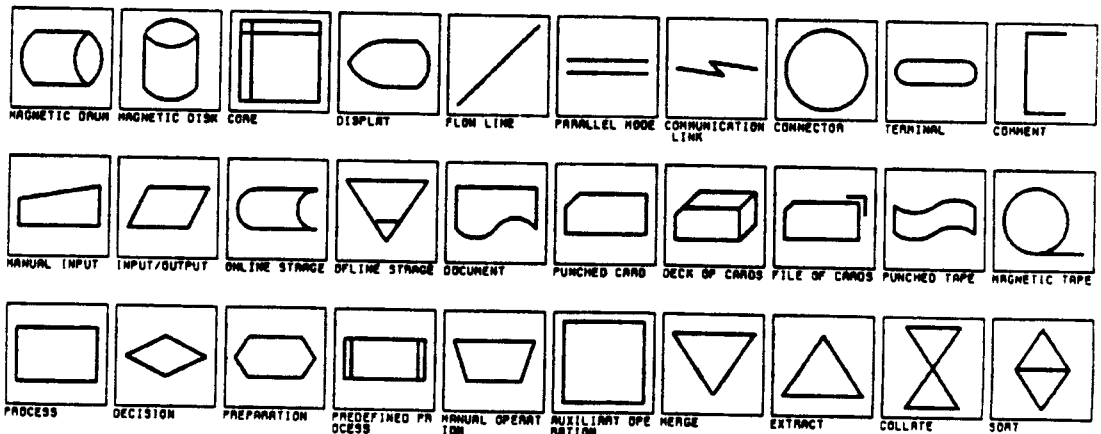


図1 JIS C6270で定められたシンボル

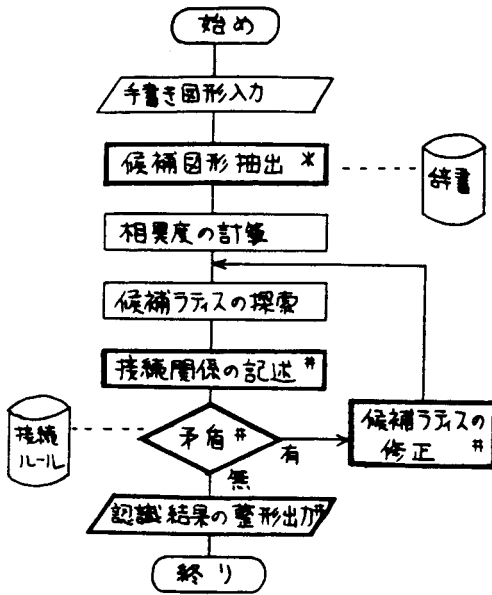


図3 認識アルゴリズム

までの座標値系列をストロークと呼ぶことにする。前処理として、一定時間間隔で得られる座標値系列を、一定距離間隔の座標値系列に変換し、以下の処理に使用する。

### 3.3 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。本処理では、位相構造的な観点（入力ストロークの端点の位置と端点間の接続関係を利用）から、各シンボルに対するすべての候補図形を抽出する。更に、抽出された候補図形に対しては、シンボル上をどのような筆順と画数で書かれるかの候補を生成する。この候補のことを候補ストローク列と呼ぶ。

本節ではまず前回報告<sup>(4)</sup>の候補図形抽出法の概要を述べる。次いで、フローチャートデータに対する抽出実験結果に基づき問題を分析し、これに対する候補図形抽出法の改良について述べる。

前回の手法を図4を例に説明する。入力図形内のある連続したストロークの組を図4(a)である場合に、これがシンボル「Mag. disk」の候

補図形として抽出される手順を示す。

- (1) 「mag. disk」の辞書を図4(b)のようなグラフで表現する。
- (2) 入力ストロークの端点を、辞書シンボルの候補端点に対応付ける。但し、候補端点とは、シンボルにおいてストロークの端点が位置し得る箇所のみであり、シンボルデータの事前調査により決定する。対応付けは辞書シンボルの大きさを正規化し、入力ストロークを重ね合せた後に、ストロークの各端点をそれぞれ最近傍の候補端点に対応付ける（図4破線参照）。
- (3) 上記で対応付けられた関係を満足するパス（ペンの動きに対応）をすべて列挙するため、グラフ（図4(b))を探索する。この処理は本質的にはオイラーパスを求めるグラフ問題と等値である。探索が失敗すれば、この入力ストロークは「Mag. disk」に対する候補図形ではないとされる。図4の例では探索は成功する。
- (4) 探索が成功した際のグラフ上でのパスが候補ストローク列となる。この例では、図4(c)に示す2種類の候補ストローク列が生成される。

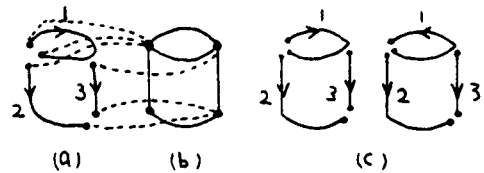


図4 候補ストローク列の生成

実際には入力図形のすべてのストロークの組に対して(1)~(4)の処理を行う。例えば入力図形が20ストロークから構成され、「Mag. disk」の候補図形を抽出する場合には、5画の場合16組、4画の場合17組、...、1画の場合20組のストロークの組、つまり全体で90組のストロークに対して上記の処理を行う。

この手法を今回の認識対象データに適用したところ、候補図形抽出率は95.4%であった。但し、

$$\text{候補図形抽出率} = \frac{\text{正しく候補図形が抽出された数}}{\text{入力図形中に含まれるシンボル数}} \times 100$$

で定義した。

候補図形が正しく抽出されなかった4.6%の原因としては、2つ考えられる。1つは、シンボル内に近接した2つの候補端点が存在する場合に、手書き変形により入力ストロークの端点が近接した候補端点に誤対応し、そのため抽出できなかつたもので4.2%存在した。もう1つの原因は、候補端点は相互に離れているが、入力ストロークが極端に変形したために、候補端点が正しく対応できなかつたもので0.4%存在した。

ここでは特に前者の原因による誤抽出に対応するため、グラフの探索時(処理(3)参照)に近接する候補端点に対応するノードを結合する手法を導入する。これは近接する候補端点を見かけ上1点として扱うことに相当する。結合されたグラフの例を図5に示す。その結果、入力ストロークの端点が、正しい候補端点に近接する候補端点に誤対応した場合でも、正しい候補ストローク列が生成される。つまり正しく候補図形が抽出される。

例えば図6(a)に示す入力ストロークの端点が、図6(b)に示す候補端点に破綻で示すように対応付けられたとしても(第1ストロークの終点が誤対応)、図5(c)のようにグラフを変形して探索することにより、図6(c)に示す正しい候補ストローク列が生成される。

本手法を導入することにより、近接する候補端点内でのストローク端点の誤対応は許容されることになり、候補図形抽出率は99.6%に改善された(表2参照)。

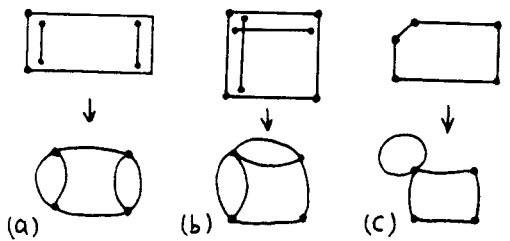


図5 結合されたグラフ

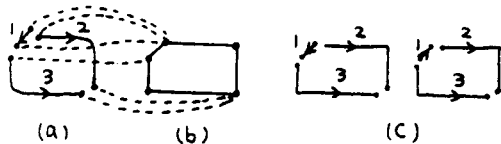
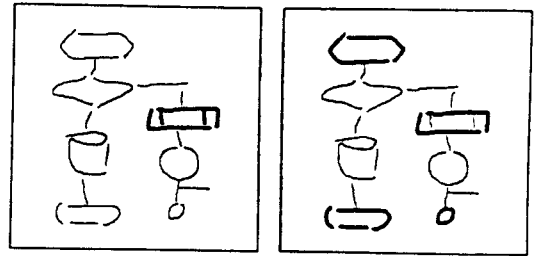


図6 候補ストローク列の生成

表2 候補図形抽出率

前手法	本手法
95.4%	99.6%

本手法により候補図形を抽出した例を図7に示す。図中太線が候補図形である。図7(a)に「□」の候補図形を、図7(b)に「□」の候補図形を示す。



(a) 「□」

(b) 「□」

図7. 候補図形の抽出例

### 3.4 相異度の訂正(DPマッチング法)

候補図形の抽出段階では、位相構造特徴のみを利用し、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、入力ストロークと3.3章で生成した候補ストローク列との間で形状のマッチングを行い、相異度を計算する。以下に訂正方法を述べる。

候補ストロークの記述(直線または円弧の系列)を具体的な座標値系列に変換した後、対応する入力ストロークとの間で座標値系列のDPマッチングを行い、1ストローク分の相異度を計算する。各ストロークの座標値系列を  $\{(x_m, y_m)_{m=1, M}\}$ ,  $\{(x'_n, y'_n)_{n=1, N}\}$  (但し  $M, N$  は近似点数である) とすれば、相異度  $d$  は

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

である。但し、 $u(m)$  は座標点間の対応関係を表わし、次の関係を満たす。

$$\begin{cases} u(1) = 1 \\ u(M) = N \\ u(i) = j \text{ のとき } u(i+1) = \{j \text{ or } j+1 \text{ or } j+2\} \end{cases}$$

又  $h(i, j)$  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す関数で、 $\alpha$  はその係数

である。

入力ストロークと候補図形との相関度は、その図形を構成する各ストロークの相関度の総和で定義する。

### 3.5 候補ラティスの作成

候補ラティスは、入力図形中の各ストローク毎の候補図形あるいは候補図形の一部として抽出されたものを、表形式に表現したものである。以下に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(3.3参照)。候補図形はストロークの番号列で得られるので、候補ラティスの対応するストローク番号列の位置に、候補図形のシンボル名を登録する。その候補図形に対する入力ストロークとの相関度の計算を行い(3.4参照)、相関度も合せて登録する。他のシンボルに対してはこの処理を繰返す。又、すべてのストロークは直線の候補図形でもあるので、これも合せて登録する。一例として、図8に示す入力図形に対する候補ラティス(一部分)を図9に示す。

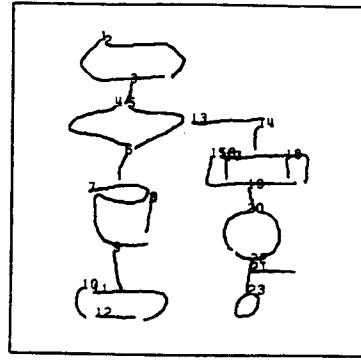


図8 入力図形

ストローク番号	15	16	17	18	19	20	21	22	23
	—	—	—	—	—	—	—	—	—
	0.187	0.170	0.025	0.012	0.019	0.610	0.008	0.025	0.707
	0.344					Q			0.097
	0.070				0.132				0.101
	0.076				0.038				0.140
					0.113				

図9 候補ラティス

### 3.6 候補ラティスの探索

入力図形中に含まれるシンボルをセグメンテーションし、各シンボルの名称を識別すること(入力図形の認識)は、候補ラティスの中からすべてのストロークをおおむね最適な図形系列を選択することにより実現できる。一般には、候補ラティスの中から図形系列を選択する場合、図形系列の組合せは多数存在する。そこで相関度を用いた評価関数を導入し、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。但し、評価関数Sは、

$$S = \sum_{\text{候補図形} \in \text{図形系列}} (\text{相関度}) / (\text{ストローク数})$$

で定義する。一例として図9の候補ラティスを探索すると、太枠の候補図形が選択される。図10は、選択された候補図形を、入力図形内の対応するストロークの位置と太枠に合せて表示した例である。この例では破線内に示される4箇所が誤認識されている。前回報告の手法ではここで認識処理を終了する。

これまで述べた手法にて、今日の認識対象データを認識したところ、認識率は89.2%で

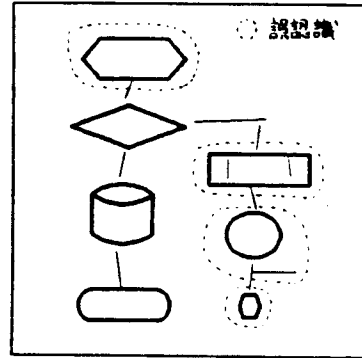


図10 第1回目の認識結果

あった。但し、認識率は

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

で定義した。

誤識した原因を分析すると以下の2点が見られる。

- セグメンテーションの失敗は2.9%存在した。主な原因は、シンボル相互間の包含関係にある。包含関係とは、「□」↔「□」+2本の直線、「Q」↔「O」+直線、等のように一方の

シンボルの部分図形が他方のシンボルであるような関係のことである。この関係にあるシンボルの認識では、認識が正解の場合でも両者の評価関数の値は類似していた。つまり誤読する危険が高い。

- (2) シンボル名の誤読は7.9%存在した。原因はシンボル間の形状の類似にある。人間でも単独に切り出された手書きシンボルを提示されたら、誤読するような例である。

ID	SYMBOL-NAME	STROKE-NO.	CONNECTION-POINTER
1	PREPARATION	1,2	501(S)
2	DECISION	4,5	501(E),502(S),504(S)
3	MAGNETIC DISK	7,8	502(E)
4	TERMINAL	10,11,12	503(S)
5	PROCESS	15,16	505(E),506(SE),507(SE),508(S)
6	CONNECTER	20	508(E),510(S)
7	PREPARATION	23	510(E)
501	LINE	3	1
502	LINE	6	2
503	LINE	9	3
504	LINE	13	2,505(S)
505	LINE	14	504(E)
506	LINE	17	5
507	LINE	18	5
508	LINE	19	5
509	LINE	21	510(M)
510	LINE	22	6

図11 接続関係の記述

人間がこれらの図形を認識できる理由は、人間は、シンボル個々の形状情報のみならず、図形の全体情報（例えば、シンボル間の接続関係や、図形の持つ意味）を利用しているためである。ここでは上記問題点に対処するため、新たにシンボル間の接続関係を上位概念として利用する手法を導入する。以下、節3.7、3.8で詳細に述べる。

### 3.7 接続関係の記述

フローチャートは複数のシンボルとそのシンボル間を接続する直線により構成されている。この接続関係を記述することは、図形の認識に接続ルール等の上位概念を適用する際、あるいは認識結果を整形し出力する際に必要となる。それらの目的のために、ここでは以下に示す記述法を導入する。

- (1) 各シンボル又は直線は、識別番号 (ID) と、入力のパターンを参照するためのストローク番号 (STROKE-NO.) と、接続する対象を示すポインタ (CONNECTION-POINTER) とを持つ。
- (2) 直線が直線に接続する場合には、直線のどの箇所 (始点 (S), 中間点 (M), 終点 (E)) に接続するかの情報もポインタに含む。
- (3) シンボルが直線に接続する場合には、直線のどの箇所 (始点 (S), 終点 (E)) に接続するかの情報もポインタに含む。

上述の記述法で、認識結果 (図10) の接続関係を記述した例が図11である。

### 3.8 接続ルールと候補ラティスの修正

フローチャートには存在し得ないシンボルの接続関係がある。例えば、シンボル「Terminal」がフローチャートの分岐点に存在したり、直線

が2つのシンボルを接続する役割りを果たしていない、等はフローチャートとして明らかに矛盾している。そこで、認識結果がこのような矛盾した結果となった場合、誤認識であるとし、処理にフィードバックをかけた再認識することが望ましい。ここでは接続ルールを導入し、認識結果が接続ルールに反する場合には、候補ラティスを修正し、再度認識しなおし、接続ルールを満足するまでこの処理を反復実行する手法を導入する。

以下に、今回実際に採用した接続ルールと、接続ルールに反した場合の候補ラティスの修正法を述べる。このルールは節3.6で分析した誤読原因を考慮して決定したものである。

- ① シンボル (「Connector」「Terminal」) は、処理の分岐点または中間点 (2つ以上の直線に連結する箇所) には存在し得ない。
- ② シンボル (「Decision」「Preparation」「Predefined process」「Collate」「Sort」「Communication line」) は処理の端点 (1つの直線に連結する箇所) には存在し得ない。
- ③ 1本の直線の両端が同一シンボルに接続することはない。これは直線がシンボルの内部に存在する場合に発生する。  
入力図形がこの接続ルールの条件を1つでも満足しなければ、その候補図形を候補ラティスから削除し、処理はフィードバックされる。接続ルールに対し矛盾がなくなった段階で、その際の候補ラティスの探索結果を最終的な認識結果とする。

この接続ルールを図11に示す記述例に適用すると、5箇所条件を満足しない。その箇所を図11中に①~⑤で示す。但し、①~⑤は接続ルールの番号に対応する。候補ラティスから不要な候補図形を削除した例を図12に示す。この例

の場合には、処理はフィードバックされる。第2回目の認識結果、第3回目（最終段階）の認識結果を図13(a), (b)に示す。

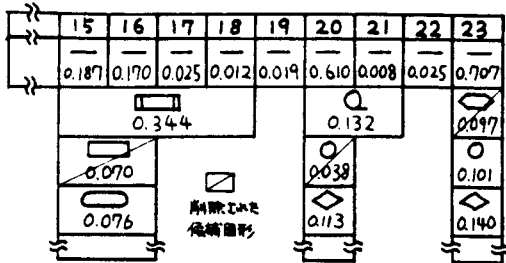
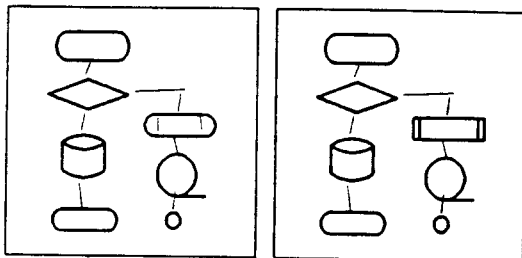


図12 修正された候補ラティス



(a) 第2回目認識結果 (b) 最終(第3回目)認識結果

図13 中間認識結果

### 3.9 認識結果の整形出力

入力図形のレイアウトに従い、認識結果を整形して出力する手法について述べる。認識結果を、入力図形内の対応するストロークの位置と大まかに合せて表示すると、図13に示すような不自然な表示となる。そこで以下に示す整形処理を施す。

- (1) あるシンボルのX方向の位置が他のシンボルのそれに類似している場合には、シンボルをX方向に移動し、その位置を一致させる。
- (2) 3.7章で示した接続関係の記述に従い、直線の位置を補正する。
- (3) 水平または垂直に近い直線は、水平または垂直位置に回転する。

この処理に従って、図13(b)に示す認識結果を整形した例が図14である。

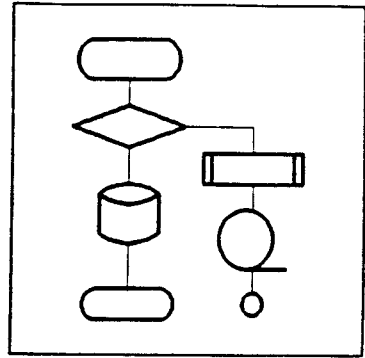


図14 認識結果の整形出力例

## 4. 認識結果

100枚のフローチャートデータ(2章)を対象に認識実験を行った。認識結果を表3に示す。

表3 認識結果

接続ルール	認識率	セクメント誤り	シンボル誤り
無	89.2%	2.9%	7.9%
有	95.9%	0.6%	3.5%

接続ルールを導入したことにより、セクメント誤りの誤り(シンボルの包含関係が原因)、シンボル誤り(シンボル形状の類似)に対し、大幅な改善がみられた。4.1%の誤謬の例には、今回の接続ルールでは修正できなかったもの(「O」→「□」0.5%、「O」→「◇」0.4%...)、候補図形として抽出されなが、たもの(0.4%)等がある。

## 5. 考察

(1) 候補図形抽出段階に、近接した候補端点を一つのノードと見なす処理により、候補端点の対応付けが誤った図形や、それ以外の不用な図形をも候補図形として抽出する。従って、相異度の計算量が増大し、全体の計算時間は増加することが予想される。そこで計算時間の増加割合を調査した。本処理を導入しない際の計算時間を1とすると、導入した際の計算時間は1.3(100枚のフローチャートに対する結果の平均値)であった。計算時間を減少させる手法としては、候補ストローク列を探索すると同時に、候補ストロークと入力ストロークとの相異度を計算し、相異度が極端に増加した場合には探索を打ち切る手法等が考えられる。

(2) 接続ルールの利用により候補ラティスを修正し、認識処理にフィードバックをかける手法では、処理の発散が心配される。そこで、実際のデータでフィードバックの回数分布を調査した。その結果を図15に示す。3回程度でほとんどの図形は収束することになる。実際のアルゴリズムでは安全を取り、処理の発散の危険性を考慮し、フィードバックの回数を5回に制限している。

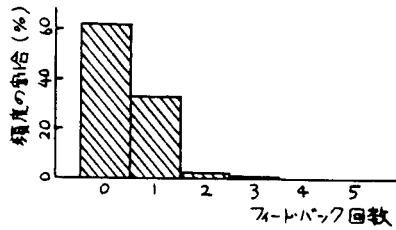


図15 フィードバック回数の分布

このような接続ルールを導入する際に考慮すべきと思われる問題点を2点述べる。

① 正しい候補図形が、その図形に近接した箇所での誤読図形の影響を受け、削除されてしまう危険がある。

② 接続ルールの過剰な導入は、図形の表現の自由度を制限することになる。

これらの点に対処するためには、接続ルールをむやみに導入するのではなく、誤読に対処するための最小限の接続ルールのみを導入することが重要である。今回の実験では、接続ルールを適用しない場合の認識率は比較的高く、接続ルールを緩く設定することが可能であり、問題は生じなかった。

(3) 入力ストローク数の増加に伴い計量時間が異常に増加することから心配される。そこで、ストローク数と認識時間の関係を調査した(図16)。認識時間はほぼストローク数に比例している。

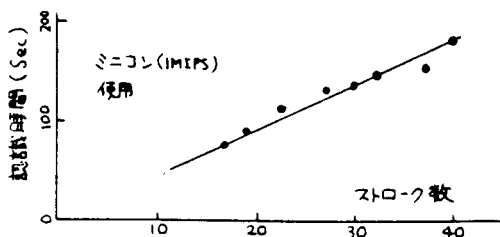


図16 ストローク数と認識時間

つまり、図形中に含まれるシンボル数が増大しても、計量時間は極端に増加しないことを確認した。

## 6. おまわり

本報告では、候補ラティス法を手書きフローチャートに適用した際の問題点の分析と、その問題点を解決するための接続ルール等を導入した候補ラティス法について述べた。以下にまとめる。

(1) 候補端点の対応付け誤りによる候補図形抽出の失敗に対しては、近接する候補端点を見かけ上一点として扱う処理を導入した。その結果、候補図形抽出率は95.4%から99.6%に向上した。

(2) シンボル間の包含関係(例えば「□」は「□」と2本の直線である)による誤読、シンボル間の形状類似による誤読に対しては、接続ルールを導入し、処理にフィードバックをかける手法を導入した。その結果、認識率は89.2%から95.9%に向上した。

(3) 認識結果を整形して表示する処理を導入し、自動描画の初期段階を達成した。

今後の課題には、接続ルールを最適化すること、本手法をフローチャート以外の図形に拡張した際の検討、文字の混入した図形の認識法の確立等がある。

謝辞 日頃御指導いただく野柳基礎研究部長 橋本統治役、増田繁二研究室長、ならびに研究室室員に感謝します。

## 文献

- (1) 佐藤, 磯上: "手書き図面処理の手法", 昭56, 情報全大
- (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, 22, 4 (8856-4)
- (3) C.Y.Suen, T.Radhakrishnan: "Recognition of Hand-drawn Flowchart", IJOPR, pp424-428 (1976)
- (4) W.C.Lin, J.H.Pun: "Machine Recognition and Plotting of Hand-drawn Line Figures", IEEE, SMC-8 (1978)
- (5) Kato, Iwase, Yoshida, Tananashi: "Interactive Hand-drawn Diagram Input System", PRIP pp544-549 (1982)
- (6) 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 (8859)
- (7) JIS: "情報処理流し図記号, C 6 270"