

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

正員 村瀬 洋[†] 正員 若原 徹[†] 正員 梅田三千雄[†]

Online Hand-Sketched Line Figure Recognition by Candidate Lattice Method with Connection Rules

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

あらまし 複数のシンボルと直線から構成される手書き線図形を対象に、自由書式、任意の画数と筆順での筆記を許し、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オンライン線図形認識法について述べる。本認識法は以下のように構成される。①入力図形からストローク端点の位置情報を用いて各シンボルおよび直線に対する候補図形を抽出し、候補ラティスに登録する、②候補ラティスから入力図形との形状相異度を最小とするシンボル系列を探索し“仮の認識結果”として選択する、③上位概念として導入したシンボル間の接続ルールに“仮の認識結果”が矛盾する場合には候補ラティスを修正し再度シンボル系列の探索を行い、この処理を反復することにより接続ルールに矛盾しないシンボル系列を最終認識結果とする。認識実験を行った結果、100枚のフローチャートおよびブロック図に対し96.1%の認識率、40枚の論理回路図に対し93.2%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

日本語文書処理システムの浸透に伴い、図形を含む文書の簡便な入力法の実現が強く望まれている。手書き線図形の認識技術を用いた入力方式は素人に扱いやすいという点で有利である。このため、従来より、既に手書きされた図面をFAXから入力して認識するオフライン型の認識法^{(1)~(3)}、およびタブレット上に手書きしつつある図面を逐次認識するオンライン型の認識法^{(4),(5)}について検討が進められてきた。

筆者らがオンライン型の認識を採用する理由は、誤認識の発見や修正がその場で対話的に行え、また入力と編集の作業を融合しやすく、マンマシンインターフェースの観点からも有望と考えるからである。

先の報告⁽⁶⁾では特に筆記時の制限緩和について検討し、形状の比較的単純な9種類のシンボルから構成される自由書式の手書きフローチャートを対象に、シンボル間のセグメンテーション指示を必要とせず、任意の画数と筆順での筆記を許す認識アルゴリズムとして

候補ラティス法を提案し、有効性を確認した。しかし、この手法をより一般的な線図形の認識に適用するには、①形状の複雑なシンボルに対し、手書き変形のために正しく候補図形が抽出されない、②候補図形は正しく抽出されても、形状の類似した他のシンボルに誤読しやす、③包含関係にあるシンボル間でセグメンテーションの失敗を招きやす、等の問題点が残されていた。

本論文では上記問題点を解決して、認識対象をフローチャート、ブロック図、論理回路図などの一般的な線図形に拡張した、高精度なオンライン線図形認識法について報告する。要点を以下に述べる。

(1) 複雑なシンボルに対しては、候補端点連結処理を導入することにより、手書き変形に安定な候補図形の抽出を可能にする。

(2) 上位概念として接続ルールを導入して、シンボル間の接続関係に矛盾のないシンボル系列を認識結果として決定する。これにより、形状だけでは判別困難な類似シンボル間や、包含関係にあるシンボル間での誤読を救済する。

(3) 認識結果の表示において、シンボル間の接続関係や位置関係を利用した整形処理を導入する。

[†] 電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

2 認識対象

認識対象の一例として、JIS C 6270⁽⁷⁾で定められたシンボルと直線から構成されるフローチャートおよびブロック図をとりあげる。図1に対象とするシンボル群を示す。具体的には、6~12個程度のシンボルとそれらを接続する直線からなる5種類の図形（フローチャート3種類、ブロック図2種類）を設定した。それらを5名の筆記者に各4回ずつ筆記させ、合計100枚の図形データ（総シンボル数780）を収集した。一例を図2に示す。筆記にあたっての指示は以下の3点である。

- ① 各シンボルの画数と筆順は自由である。
- ② 各シンボルの位置と大きさは自由である（自由書式である）。
- ③ 1つのシンボルは引き続くストローク列で筆記するものの、シンボル間のセグメンテーションの指示は不要である。

3 認識アルゴリズム

認識アルゴリズムを図3に示す。候補ラティス法⁽⁶⁾を基本とするが、複雑な形状のシンボルにも対処可能な候補図形抽出法（図中*印）を導入するとともに、上位概念としてシンボル間の接続ルールを導入し、接

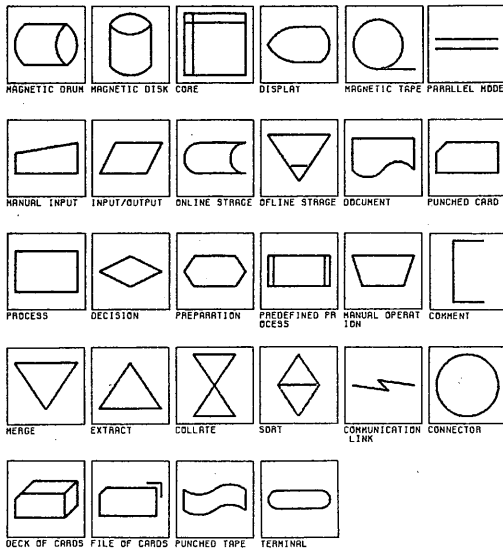


図1 シンボル (JIS C 6270)
Fig.1-Symbols (JIS C 6270).

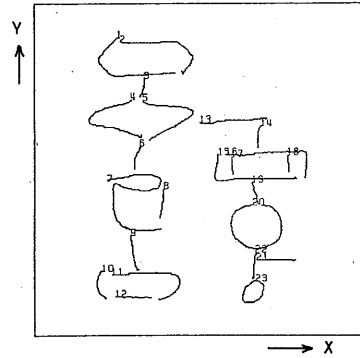


図2 入力図形の一例
Fig.2-Example of input pattern.

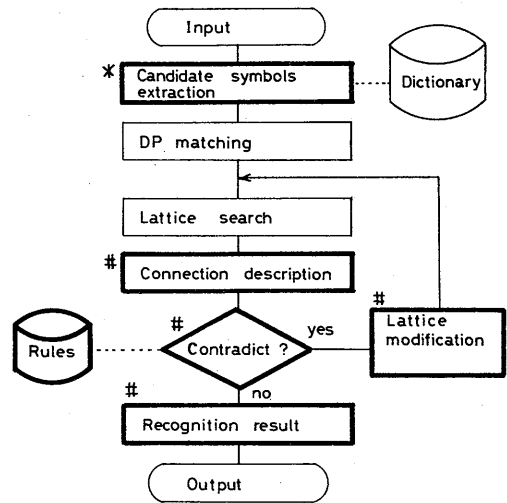


図3 認識アルゴリズム
Fig.3-Recognition algorithm.

続ルールを満足するように候補ラティスの探索と修正とを反復（図中#印）することにより認識を行う。

3.1 手書き線図形入力

手書き線図形はデータタブレットから入力する。使用したタブレットはペンのアップダウン情報とXY座標値を一定時間（10 msec）間隔で検出する。ペンのダウンからアップまでの座標値系列をストロークと呼ぶ。各ストロークには入力順にストローク番号を付与し、以下の処理に使用する。

3.2 候補図形の抽出

図1に示した各シンボルの存在を順次仮定し（以下「仮定したシンボル」と呼ぶ）、「仮定したシンボル」と成り得る入力図形中のストローク列を候補図形とし

て抽出する。処理は、まず入力図形中の1組の引き続くストローク列(以下、入力ストローク列と呼ぶ)に着目し、これが「仮定したシンボル」であるための条件をストロークの端点の位置と接続関係を用いて調べ、条件を満足すればこの入力ストローク列を候補図形とする。この処理を順次他の入力ストローク列に対し適用することにより候補図形の抽出は終了する。「仮定したシンボル」であるための条件を調べるには、シンボルの形状が単純の場合には文献(6)の候補図形抽出法の適用が可能であるが、複雑な場合にはストロークの端点の対応付けが不安定になり適用できない。そこで、候補端点の連結処理を導入し、安定な候補図形抽出を実現する。

入力ストローク列が図4(a)の場合を例に、これがシンボル「Punched card」の候補図形として抽出される手順を説明する。

(1) 各シンボルの辞書を、候補端点(シンボル上でストロークの端点となり得る点)をノードとし、候補端点間の接続関係をブランチとするグラフで表現する。合せて標準形状を座標値で格納しておく。「Punched card」のグラフ表現を図4(b)に示す。

(2) 辞書中のシンボルの座標表現を拡大縮小し、入力ストローク列に重ね合わせる。次いで入力ストローク列の各端点をそれぞれ近傍の候補端点に対応付ける(図4(a), (b)間の破線参照)。近傍に候補端点がない

場合には、候補図形抽出のための条件は不成立となる。この例では、対応付けは成立する。

(3) シンボル形状が複雑で、シンボル中に近似した候補端点群が存在する場合には、(2)の候補端点の対応付けが誤っている可能性がある(図4(a)の第1ストローク終点が誤対応)。そこで、辞書中のグラフで、近接した候補端点のノードを連結する(候補端点連結処理)。図5(a)~(c)のグラフに候補端点連結処理がなされた例を(a')~(c')に示す。

(4) (2)で決定した端点の対応関係を満足し、すべてのブランチを通過するパスをグラフ(図5(a'))上で探索する。この処理は基本的にオイラーパスを求めるグラフ問題と等価⁽⁶⁾である。探索に失敗すれば、入力ストローク列は候補図形とならない。この例では探索に成功し、入力ストローク列は「Punched card」の候補図形となる。

(5) 探索に成功した場合には、グラフ上での探索パスに対応する座標値系列を候補ストローク列と呼び、次節のストローク間のDPマッチングに使用する。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。

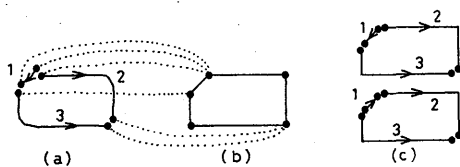


図4 候補ストローク列の生成
Fig.4-Generation of candidate strokes.

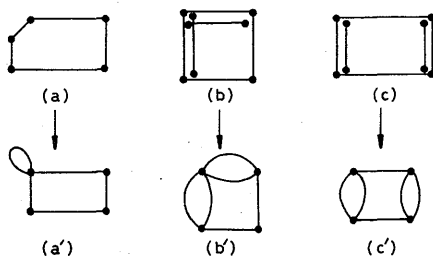


図5 候補端点の連結
Fig.5-Candidate endpoints linking.

図6に候補図形抽出例を示す。図6(a)はシンボル「Predefined process」の候補図形、図6(b)はシンボル「Process」の候補図形である。

2.で述べた線図形データに対し、文献(6)の手法と本節で提案の候補端点連結処理を導入した手法との間で、候補図形抽出能力の比較実験を行った。候補図形

表1 候補図形抽出率

文献(6)の手法	本手法
95.4%	99.6%

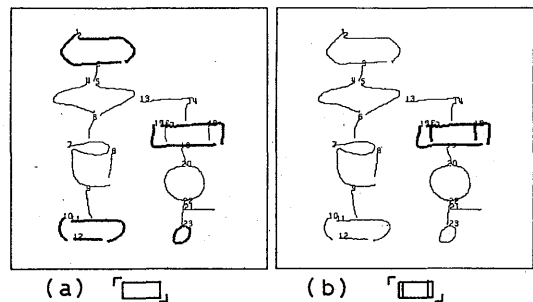


図6 候補図形の例
Fig.6-Candidate figures for symbols.

抽出率を表1に示す。候補図形抽出率は、入力図形に含まれるシンボル数に対する、候補図形に含まれる正解シンボル数の割合で定義した。

3.3 DP マッチングによる相異度の計算

候補図形の抽出段階では、入力ストロークの端点位置とその接続関係のみを利用し、入力ストローク列の細かい形状については考慮していない。そこで、ストローク間のDP マッチング⁽⁶⁾による形状の相異度を導入する。3.2で生成された候補ストローク列の1ストロークの座標値系列を{ p_1, p_2, \dots, p_M } (M は点数)とし、入力ストローク列のこれに対応するストロークの座標値系列を{ q_1, q_2, \dots, q_N } (N は点数)とすれば、ストローク間の距離 d は、DP マッチングにより、

$$d = \frac{1}{M} \min_u \left\{ \sum_{i=1}^M e(p_i, q_{u(i)}) \right\} \quad (1)$$

で計算される。但し、 $e(p_i, q_j)$ は点 p_i と点 q_j の間のユークリッド距離と、その点での接線方向の方向差の加重和とする。 $u(i)$ は座標点間の対応関係を表し、

$$\begin{cases} u(1) = 1 & (2) \\ u(M) = N & (3) \\ u(i) = j \text{ のとき } u(i+1) = \{j \text{ or } j+1 \text{ or } j+2\} & (4) \end{cases}$$

の関係を制約条件にする。

候補図形の相異度は、その図形を構成する各ストロークの距離の総和で定義する。

3.4 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がどの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。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。候補図形はストロークの番号列で得られる。そこで、候補ラティス内の対応するストローク番号の位置に、候補図形の名称(シンボル名)とその相異度を登録する。他のシンボルに対し

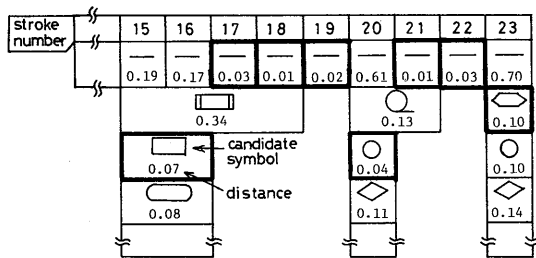


図7 候補ラティス
Fig.7-Candidate lattice.

てもこの処理を繰返す。又、すべてのストロークは直線の候補図形でもあるため、これも合せて登録する。一例として、図7に入力図形(図2)に対する候補ラティス(一部分)を示す。

3.5 候補ラティスの仮の探索

入力図形に含まれるシンボルのセグメンテーションおよび、シンボル名の識別を行うこと(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覆う最適な図形系列(シンボルおよび直線からなる系列)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選択可能な図形系列の組合せは多数存在する。そこで、第一段階として文献(6)のラティス探索法により、評価関数 S

$$S = \sum_{\text{候補図形} \in \text{図形系列}} \frac{\text{相異度}}{\text{ストローク数}} \quad (5)$$

を導入し、 S を最小とする図形系列を“仮の認識結果”とする。一例として、図7の候補ラティスを探索すると、太枠の候補図形が“仮の認識結果”となる。図8は“仮の認識結果”を、入力図形(図2)の、対応するストロークの位置と大きさに合せて表示した例である。この例では、破線内に示した4箇所で誤認識が生じている。

上記の手法のみで、今回の認識対象データを認識したところ、認識率は89.8%であった。但し、認識率は、

$$\text{認識率} = \frac{\text{正しく認識されたシンボル数}}{\text{入力図形中に含まれるシンボル数}} \times 100 \quad (6)$$

で定義した。

誤読例を分析すると、セグメンテーションの失敗とシンボル名の誤読に大別される。以下に分析内容を述べる。

- (1) セグメンテーションの失敗は2.3%存在した。

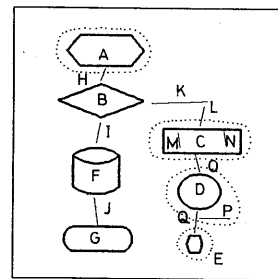


図8 仮の認識結果(1)
Fig.8-Tentative recognition result (1).

表2 シンボル間の包含関係の一例

包含するシンボル名	包含されるシンボル名
「Predefined process」	「Process」「Line」
「Sort」	「Decision」「Merge」他
「Magnetic tape」	「Connector」「Line」

代表的な失敗には、シンボル「Core」を「Process」と2本の直線に誤読したもの0.9%、「Sort」を「Decision」と直線に誤読したもの0.3%等があげられる。原因は、一方のシンボルの部分図形が他のシンボルであるというシンボル相互間の包含関係にある。図1に示したシンボルの包含関係の一部を表2に示す。包含関係にあるシンボルの認識では、両者の評価関数の値は類似するため、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。代表的な失敗には、「Terminal」を「Preparation」に誤読した例2.3%、「Connector」を「Preparation」に誤読した例2.3%等があげられる。原因はシンボル間の形状の類似にある。誤読例は、人間でも個々の手書きシンボルを単独に提示されたら誤読するような例である。

上記の2種類の誤りが発生する理由は、この段階ではシンボル個々の形状情報しか利用していないことによる。そこで、次段階として、シンボル間の接続関係やシンボルの持つ意味などの大局情報を上位概念として接続ルールの形で利用し、「仮の認識結果」を修正する手法を導入する。

3.6 接続関係の記述

認識結果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。そこで、相互の接続関係を以下のように記述する。

(1) 各シンボルや直線は、接続する対象を示す接続ポインタ(C.POINTER)を持つ。

(2) 各シンボルや直線は、原パターンを参照するためのストローク番号(ST.NO.)を持つ。

上記の記述で、(1)は接続ルールの適用の際に必要であり、(2)は認識結果を整形して出力する際に必要となる。図9は、図8に示した「仮の認識結果」の接続関係を記述した例である。図中IDはシンボルや直線に仮に付した記号で、図8中の記号A~Qに対応している。図9によれば、例えば記号Bのシンボル

「Decision」は、入力図形(図2)のストローク番号の4, 5から構成され、記号H, I, Kの直線に接続されている。

ID	SYMBOL NAME	ST.NO.	C.POINTER
A	PREPARATION	1,2	H
B	DECISION	4,5	H,I,K
C	PROCESS	15,16	L,M,N,O
D	CONNECTOR	20	O,Q
E	PREPARATION	23	Q
F	MAG. DISK	7,8	I,J
G	TERMINAL	10,11,12	J
H	LINE	3	A,B
I	LINE	6	B,F
J	LINE	9	F,G
K	LINE	13	B,L
L	LINE	14	K,C
M	LINE	17	C,C
N	LINE	18	C,C
O	LINE	19	C,D
P	LINE	21	Q
Q	LINE	22	D,P,E

図9 接続関係の記述例
Fig.9-Connection description.

3.7 接続ルール

フローチャートなどのような複数のシンボルから構成される図形には、意味の上からは存在し得ないシンボル間の接続関係がある。例えば、「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、このようなシンボル間の関係を接続ルールとして登録しておく。「仮の認識結果」が接続ルールに反する場合に候補ラティスを修正し、再度認識しなおす。接続ルールを満足するまでこの処理を反復実行して矛盾のない認識結果を決定する。接続ルールは、性質の異なる各種ルールの記述部 B_i (i はルールの種類を示す)と、そのルールが作用するシンボル名称部 S_{ij} (j はシンボルの種類を示す)という構成で表現する。

2に示した認識対象に対して登録した接続ルールを表3に示す。この例では B_i と S_{ij} の組合せにより52種類の接続ルールが存在することになる。

接続関係の記述が、上記接続ルール52個のうち1個でも満足しなければ、そのシンボルに対応する候補

表3 接続ルール

ルールの記述部	作用するシンボル名称部
B_1 直線はシンボル内部には存在しない。	S_{1j} すべてのシンボル $j=1\sim 28$
B_2 シンボルに接続する直線数は1である。	S_{2j} 「Terminal」 「Connector」 $j=1\sim 2$
B_3 シンボルに接続する直線数は2以上である。	S_{3j} 「Preparation」 「Decision」 等 $j=1\sim 6$
B_4 極端に小さいシンボルは存在しない。	S_{4j} 「Terminal」 「Display」 等 $j=1\sim 16$

図形を候補ラティスから削除し、候補ラティスの探索処理を再度実行する。接続関係に矛盾がなくなるまで上記処理を反復実行し、矛盾が解消した段階での候補ラティスの探索結果を最終的な認識結果とする。

図9に示した記述例にこの接続ルールを適用すると、4箇所のシンボルで接続ルールに矛盾が生じる。そのシンボルを図9中に*印で示す。そこで、矛盾したシンボルに対応した候補図形を図7に示した候補ラティスから削除(候補ラティスの修正)し、再度候補ラティスを探索する。この例の場合には、候補ラティスの修正と探索の処理を2回反復実行した後に認識結果が得られる。1回反復した後の仮の認識結果を図10(a)に示す。この段階で接続ルールに矛盾する箇所を破線で示す。図10(b)は、2回反復した後の最終認識結果である。

3.8 認識結果の整形出力

認識結果の各シンボルと直線を入力図形中の対応するストロークの位置と大きさに合わせてそのまま表示すると、不自然な表示となる。そこで、認識結果の表示においてシンボル間の接続関係の記述に基づく以下の

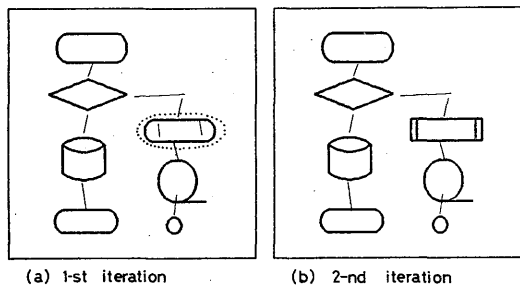


図10 仮の認識結果(2)
Fig.10-Tentative recognition result (2).

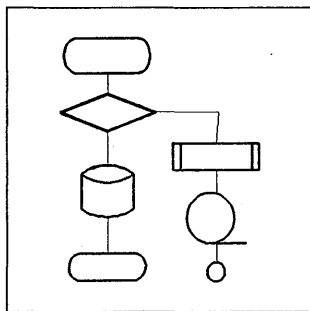


図11 認識結果の出力
Fig.11-Output of recognition result.

整形処理を導入する。

- (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は、シンボルのX方向の位置を一致させる。
- (2) 水平(垂直)に近い直線は、水平(垂直)に正規化する。
- (3) 接続関係の記述に従い、直線の端点位置等を補正する。

この処理に従って、図10(b)の認識結果を整形して出力した例を図11に示す。

4 認識実験結果

3.で述べた認識アルゴリズムを用いて、手書き線図形の認識実験を行った。2.で述べた100枚のフローチャートとブロック図に対する認識率を表4に示す。認識率は式(6)(3.5参照)で定義した。接続ルールの導入により96.1%の認識率が得られた。

5 考察

5.1 接続ルール導入の効果

接続ルール無しで認識した際の誤読は、原因別以下にI~IVに分類可能である。I:シンボル間の包含関係によるセグメンテーションの誤り、II:フローチャートの開始点や終了点に存在するシンボルをそこには存在し得ないシンボルに誤読、III:フローチャートの分岐点や中間点に存在するシンボルをそこには存在し得ないシンボルに誤読、IV:上記以外の誤読、の4種類である。3.7で述べた接続ルール $B_1 \sim B_4$ は、それぞれ上記I~IVの誤読の原因を排除するためのものである。接続ルール導入の効果を表5に示す。各接続ルールの適用により対応する誤読が減少し、各接続ルールの有効性が確認された。

接続ルール $B_1 \sim B_4$ を導入しても修正不能であった

表4 認識率

接続ルール無し	接続ルール有り
89.8%	96.1%

表5 接続ルール導入の効果

接続ルール	無	B_1	B_1, B_2	$B_1 \sim B_3$	$B_1 \sim B_4$
誤読 I	2.3%	0.5	0.4	0.4	0.4
誤読 II	5.1	5.1	0	0	0
誤読 III	0.6	0.6	0.6	0	0
誤読 IV	2.2	2.2	4.6	4.9	3.5
認識率	89.8	91.6	94.4	94.7	96.1

3.9%の誤読には、セグメンテーションの誤り0.4% (誤読I), 「Connector」を「Display」に誤読したもの0.5% (誤読IV), 「Connector」を「Terminal」に誤読したもの0.4% (誤読IV), 等があった。これらに対しては、更に他の接続ルールを導入する必要がある。

5.2 候補ラティスの修正と探索の反復実行回数

本手法では、認識した結果が接続ルールに矛盾しなくなるまで候補ラティスの修正と探索を反復実行する。上記実験での反復実行回数の頻度分布を調査した。結果を図12に示す。また候補ラティスの修正と探索の反復実行を途中で打ち切った時の認識率を図12に重ねて示す。反復実行回数が0とは、接続ルールを適用しないことを意味している。この例では、40%のデータが1回以上の反復実行を受け、それにより認識率は6%程度向上することがわかる。反復実行回数が4回で認識率は飽和しているため、ここでは処理の発散の危険性を考慮し、候補ラティスの修正と探索の反復実行は4回で打ち切ることとする。

5.3 認識時間

ミニコン(1MIPS)上でFORTRAN77を用いシミュレーション実験を行った際の、ストローク数と認識

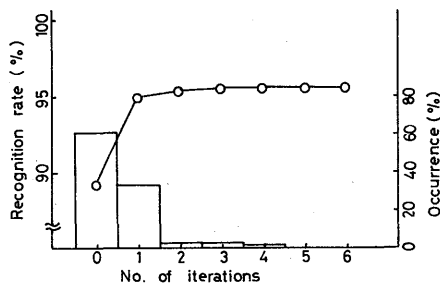


図12 反復実行回数とその出現頻度、認識率
Fig.12-Occurrence and recognition rate vs. no. of iterations.

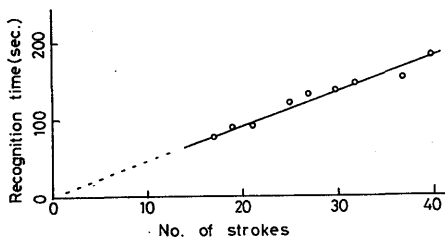


図13 ストローク数と認識時間
Fig.13-Recognition time vs. no. of strokes.

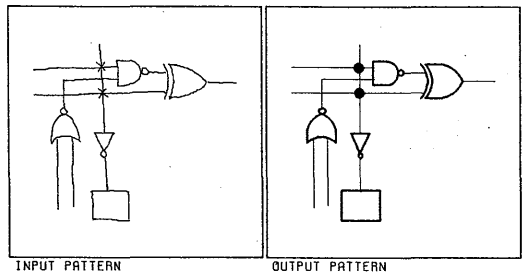


図14 論理回路図の認識
Fig.14-Recognition of logic circuit diagram.

時間の関係を図13に示す。これより、本手法による認識時間は1ストローク当たり5秒程度で、ストロークの増加にはほぼ比例し、発散的には増大しないことがわかる。更に、処理時間の約80%はDPマッチングに費やされているので、この箇所をハードウェア化することにより認識時間の短縮が容易である。一方、接続ルールによる候補ラティスの修正と探索の反復実行は簡単な記号処理により実現可能なため、処理時間上は問題にならない。

5.4 論理回路図への応用

本手法はシンボルと直線から構成される図形ならば、シンボルの辞書と接続ルールのテーブルを用意するだけで認識可能である。ここでは認識対象の拡張性を確認する目的で、論理回路図に本手法を適用してみた。辞書にはシンボル「AND」「OR」「NAND」「NOR」「EXOR」「NOT」のそれぞれを90度ずつ回転した各4種類と「CONNECT」「BLOCK」との合計26種類を登録し、接続ルールには「NOT」は1入力1出力である。など7種類のルールを登録した。2で述べたと同様の筆記条件で40枚(総シンボル数205個)のデータを収集し、認識実験を行った。接続ルール無しで90.2%、接続ルールを導入して93.2%の認識率が得られた。認識したデータの一例を図14に示す。

6 むすび

本論文では、接続ルールを上位概念として導入した候補ラティス法を提案し、フローチャート、ブロック図、論理回路図などの線図形を例に、本手法の能力を検討した。以下にまとめる。

(1) 複雑なシンボルに対しても候補図形を安定に抽出するために、候補端点連結処理を導入した候補図形抽出法を提案した。その結果、候補図形抽出率は95.4

%から99.6%に向上した。

(2) 包含関係にあるシンボル間でのセグメンテーションの失敗や、類似形状のシンボル間での誤読に対処するため、上位概念として接続ルールを導入した。入力図形を仮に認識し、“仮の認識結果”が接続ルールに矛盾する場合には、候補ラティスを修正し候補ラティスの探索処理を反復実行する手法を提案した。その結果、フローチャートやブロック図に対する認識率は89.5%から96.1%に向上した。これにより、意味を持つ線図形の認識において接続ルール等の上位概念の有用性が確認できた。

(3) 論理回路図に対し93.2%の認識率が得られた。本手法は、複数のシンボルと直線から構成される線図形の範囲ならば、シンボルの辞書と接続ルールのテーブルを用意するだけで、容易に他の図形に拡張して適用できることを確認した。

(4) 認識結果の出力表示においてシンボル間の接続関係や位置関係を利用した整形処理を導入した。

以上、本手法により自由書式で手書きされた線図形を高精度にオンライン認識することが可能となった。今後の課題には、接続ルールをより最適化して認識精

度を更に向上させること、文字の混在した図形を認識することがあげられる。

謝辞 日頃御指導頂く畔柳基礎研究部長、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第二研究室の諸氏に感謝致します。

文 献

- (1) Sato and Tojo: "Recognition and Understanding of Hand-drawn Diagram", ICPR, pp. 674-677 (1982).
- (2) 恒川, 吉野, 森, 岡崎: "手書設計図面読取装置", コンピュータビジョン研究会, 25-1 (1983).
- (3) 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 (1981).
- (4) Lin, W. C. and Pun, J. H.: "Machine Recognition and Plotting of Hand-sketched Line Figures", IEEE Trans., SMC-8 (1978).
- (5) Kato, Iwase, Yoshida and Tanahashi: "Interactive Hand-drawn Diagram Input System", PRIP, pp. 544-549 (1982).
- (6) 村瀬, 若原, 梅田: "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", 信学論(D), J66-D, 6, pp. 675-682 (1983).
- (7) JIS C 6270: "情報処理流れ図記号".
(昭和58年9月12日受付)