

細線化及び正規化による手書き文字の認識

(昭和61年11月29日 原稿受付)

電気工学科 稲永 征司

Recognition of Handprinted Characters Using Abstracted Strokes with Thinning and Normalizing Procedure

by Syoji INANAGA

Abstract

Optical character recognition methods are divided into two classes of structural method and pattern matching method. We tried to recognize handprinted characters on the stroke analysis method, one of the structural method. On this method, some strokes are extracted after thinning for input character, and only some feature extraction points on the abstracted strokes are normalized.

An experiment has been made over a set of free handprinted characters(Katakana) written by one hundred persons. It results that the recognition rates are 93.0% for ETL-1 KATAKANA. We point out some problems for recognition of them on this method that should be improved, and also mention some problems that must be resolved to recognize Chinese characters, including about three thousand ideographs designated for daily use, and Japanese characters, forty six Hiragana.

1. まえがき

文字認識の手法はパターン整合法と構造解析的手法に大別できる。前者は文字の変形や類似した文字の判別に弱く、後者は局部的雑音の影響を受け易い欠点がある。構造解析的手法には種々のヒューリスティックな方法^{1), 2), 3)}が行われているが、ストローク抽出による文字認識は抽出された特徴自体に文字の詳細な構造情報を含み、又直観的(目視的)にもわかり易く本質的手法と考えられる。

前回の報告で⁴⁾ストローク抽出による手書き文字の認識実験の結果から次の問題点を指摘した。

- (1) 線素の抽出において細線化処理をしていないので抽出される線素の長さが文字の幅程度に短くなると線素の抽出が難しい。
- (2) 交叉領域において線素と線素を結合してストロークを抽出する場合、線素間の結合を誤り正しいストロークを抽出しないことがある。

(3) 文字の認識を行う場合、入力文字の大きさやストロークの位置ずれ、文字の変形等に対して認識システムが充分に対応できない。

今回の実験では、これらの問題点に対して有効と考えられる対策を施す。(1)の問題点に対しては文字の本質的な構造を損わない程度、入力線図形に細線化を施す。(2)については二つの線素の結合の組み合わせが不適当と判断されるとき線素の組み合わせを変える処置を取る。(3)については入力文字を正規化し文字の位置の変動や変形による認識の不安定を防ぐ。以上の対策の方法については項をあらためて記述する。これらの対策を施したあと前回と同じデータ(片仮名)を使用して認識実験を試みた。その結果、第1認識率の平均93.0%、第5位までの累積認識率の平均98.8%を得た。これは前回の第1位認識率の平均74.5%、第5位までの累積認識率の平均93.4%に比べて著しい改善であるがまだ満足できる数値とはいえない。本稿ではこの結果を分析し更に認識率を高めるための方法を検討し、最後に本手法をひらがなや漢字に適

用する場合に生じると考えられるいくつかの問題点について言及する。

2. 前処理及びストロークの抽出

一般にパターンを認識する過程は入力した図形に対して図1に示すように前処理、特徴抽出、識別を段階的に得てカテゴリー名を判別する。入力文字には文字の生成過程で生じる変動成分(大きさ、太さ、位置ずれ等)が含まれているので、これらの成分を除去するために前処理として細線化と正規化を行う。正規化についてはあとの項で述べる理由で特徴抽出後行うことにする。従って実際の具体的な各処理は図2に示すような過程となる。

2.1. 細線化

電子計算機による線図形認識の前処理として一般に線図形の中心線を幅一画素に細線化する方法が行われている。この方法を文字図形に用いると文字線の縮退や消失、ヒゲ状ノイズの発生、交叉点の二分岐点への分離等の問題を生じる。これらの問題の発生を避けるために必要最

小限度の細線化を行い原図形の形状をできるだけ残すことが望ましい。従来行われている細線化の方法^{5), 6)}は上下左右からそれぞれ一画素(ビット)ずつ細線化を施すために一回の細線化動作で線の幅が二画素分ずつ減少する。この方法では入力図形の線の幅が2-4画素に量子化されている場合、線の幅の割合が急激に減少して先に述べた必要最小限の細線化という条件を満たさない。この条件を満たすために1回の細線化過程で一画素分ずつ幅が減少するアルゴリズム拡大縮小法を新たに開発した。⁷⁾以下この方法について概要を述べる。

入力文字は f 画面上で $n \times n$ の正格子に分割され格子点上で図3(a)に示すように黒点(1)と白点(0)に2値化されている。 f 画面上の格子点 (i, j) の2値化値を $f(i, j)$ で表す。今 f 画面に対応して縦、横それぞれ2倍の大きさの F 画面を考え F 画面上の格子点 (I, J) の2値化値を $F(I, J)$ で表す。

(1) f 画面上の黒点 $f(i, j)$ を $F(I, J)$ (但し、 $I=2i-1, 2i, J=2j-1, 2j$) に写像する。(図4(a)→(b))

(2) F 画面上で Hilditch の細線化法⁵⁾を用いて細線化を施す。(図4(b)の斜線の部分を消去)

(3) F 画面上の少なくとも1個の黒点を含む4画素の集合 $F(I, J)$ (但し、 $I=2i, 2i+1, J=2j, 2j+1$) から次に述べる条件で f 画面上の1画素 $f(i, j)$ に写像する。(図4(b)→(a))

i) $\sum F(I, J) \geq 2$ のとき、 $f(i, j) = 1$

ii) $\sum F(I, J) = 1$ のとき

$F(2i, 2j+1) \neq 1$ かつ $f(i, j)$ の4連結数 $\neq 1$ ならば、 $f(i, j) = 1$ 他は $f(i, j) = 0$

但し、 $\sum F(I, J) = F(2i, 2j) + F(2i+1, 2j) + F(2i, 2j+1) + F(2i+1, 2j+1)$

ここで(3)の条件ii)について説明する。 F 画面上から f 画面上に戻るとき $i \rightarrow i+1, j \rightarrow j+1$ と右下に1だけずらしているために右下の斜めの線分は細線化処理後も $F(2i+1, 2j) = 1, F(2i, 2j+1) = 1$ となり $\sum F(I, J) = 0$ とならない。このため $f(i, j)$ に写像するときここで線分の上側境界線を消去している。(図5(a), (b)) 図6(a)に入力図形(b)に細線化後の図形を示している。実際の細線化処理にあたっては文字図形の平均の線の幅 w が $w \leq w_n$ (w_n は閾値) となるまで細線化過程を繰り返す。 w_n の値は実験によって最適の値に決める必要がある。

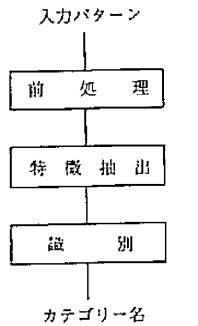


図-1 パターン認識

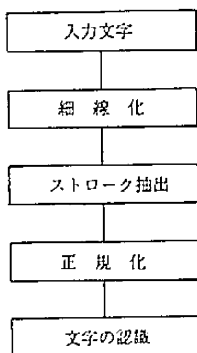


図-2 文字認識の過程

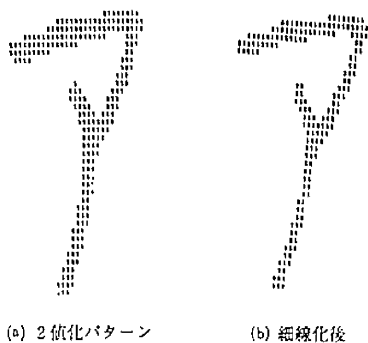


図-3 細線化

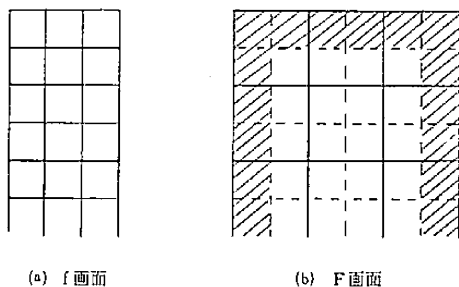


図-4 細線化処理 (横線)

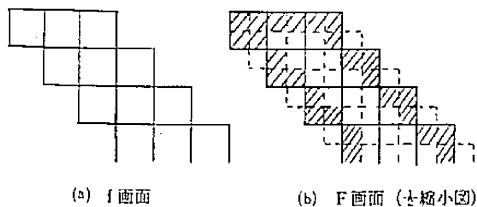


図-5 細線化処理 (斜線)

2.2. ストロークの抽出

ストローク抽出の手順については前回の報告³⁾と重複することにもなるが新たに追加した前処理等の関連で必要最小限記述する。

線図形(文字)は48×48に量子化され又2値化されている。この線図形の曲線を順次追跡して図6(b)に示すように両端を端子番号で表される線素 L_1, L_2, \dots 及び交叉領域 C_1, C_2, \dots を抽出する⁸⁾。これらの抽出された線素が交叉領域で滑らかな結合が可能なとき2つの線素を結合して1つのストロークを構成する。このとき滑らかさを表す指標 S は次式で求められる。⁴⁾

$$S = \frac{e_{12} \cdot (e_1 + e_{12})}{\|e_{21} \otimes e_{12}\|} + \frac{e_{21} \cdot (e_2 + e_{21})}{\|e_{12} \otimes e_{21}\|}$$

ここで図7に示すように e_1, e_2 は結合する2つの線素 L_1, L_2 の端点の単位方向ベクトル、 e_{12}, e_{21} は2つの線素 L_1, L_2 の端点を結ぶ線分の単位方向ベクトル($e_{12} = -e_{21}$)である。

今交叉領域で n 個の線素が交叉しているとき2つの線素の結合の組み合わせは nC_2 個($= n(n-1)/2$)あるがこの組み合わせの中から滑らかさ S の小さい値を優先してできるだけ多くの線素を結合させる⁹⁾(但し、 $S \leq c_0, c_0$ は閾値、又1つの線素は2つ以上の線素と結合しない。)この方式を文字全体の交叉領域に拡張し文字をできるだけ少ないストローク数で構成する。

2.3. 正規化

正規化は入力図形の生成過程で生じる変動成分を除去し、認識(識別)の段階での処理を容易かつ安定に行わせるための前処理と考えられる。手書き文字に生じる種々の変形のうち位置、大きさ、太さ(幅)、傾き等の変動はある程度除去できるが字形、字体における変動(手書き

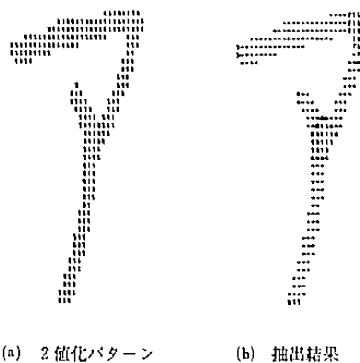


図-6 ストロークの抽出

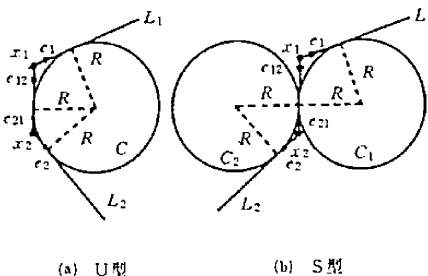


図-7 線素の交叉

変動)を除去するような方法は現在のところ知られていない。一般に高次元のパターン(図形)を実際に直接正規化するとかなりの計算量が必要であり、又それによる新たな歪の問題も生じる。そこで正規化を高次元のパターン空間で行うより低次元の特徴空間で行う方が簡単であり、又原パターンの特徴を損うことが少ないと考えられる。一般の線形特徴抽出の場合ある条件のもとでは特徴空間での正規化は可能であることが明らかにされている¹⁰⁾。従ってここでの正規化は特徴抽出後、すなわちストローク抽出後抽出されたストロークの特徴点のみに対して行うことにする。正規化には種々の方法があるがここでは長田らが一括処理法¹¹⁾で使用した修正 MAX-V法¹²⁾を用いる。以下これについて概要を簡単に述べる。

図8に示すように文字パターンの長軸方向及び短軸方向の偏差 σ_1, σ_2 を求め偏差の定数倍 $\beta \cdot \sigma_1, \beta \cdot \sigma_2$ のうち、外側の点からできる矩形で図形を切り出す。このままでは長軸が水平又は垂直方向に近づいた場合切り出された図形は細くなり過ぎるので外接矩形の短辺の長さ b を次の式で修正する。

$$\frac{b'}{a} = \left(\frac{b}{a}\right)^{\frac{1}{n_a}} \quad (\text{但し, } a > b, 1 < n_a)$$

修正後矩形の長辺 a の長さを標準図形(正方形)の1辺の長さに等しくなるように図形全体を拡大又は縮小する。実際にはこの項の初めの方で述べたように入力図形からストローク抽出を行ったのち抽出された特徴点(端点, 交叉領域の中心点)から文字(ストローク)を構成し上に述べた方法で特徴点のみ正規化を行う。

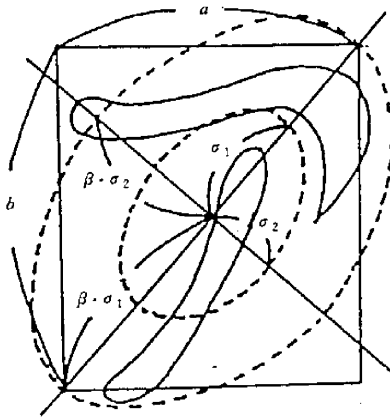


図-8 MAX-V法

3. 文字の認識

この項も前回の報告と重複することが多いので必要最小限記述する。前項までに述べたように入力図形(文字)からストロークを抽出し正規化されたストロークの集合 $\{L_i\}$ で入力文字を特徴づけることができる。次に未知入力 X が N 個のカテゴリーからなる標準文字集合 $\{R_i\}$ ($i = 1, 2, \dots, N$) のどのカテゴリーに属するかを決めなければならない。未知の入力文字 X と標準文字集合 $\{R_i\}$ 間の識別関数として次の項で定義する類似度を用いる。この類似度を用いて未知入力文字 X と標準文字集合 $\{R_i\}$ ($i = 1, 2, \dots, N$) との文字間類似度 $S_C(X, R_i)$ を求め、その最小値 $S_C(X, R) = \min. |S(X, R_1), S(X, R_2), \dots|$ を満たす標準文字 R を未知入力文字カテゴリーと決定する。

3.1. 類似度

線分間類似度

線分 $AB, A'B'$ を $t : (1-t)$ に内分する点をそれぞれ P, P' とすると線分間類似度を次のように定義する。¹³⁾

$$S_t = \int_0^1 |PP'|^2 dt = |AA'^2 + BB'^2 + AA' \cdot BB'| / 3$$

ストローク間類似度

$$S_s = \sum_l \frac{l_{li}}{\sum_l l_{li}} \cdot S_{li}$$

l_{li} : ストローク l の i 番目の線分の長さ

S_{li} : ストローク l の i 番目の線分とストローク l' の i 番目の線分との線分間類似度

$\sum_l l_{li}$: ストローク l の長さ

自己類似度

$$S_c = \sum_l \frac{l_{li}}{\sum_l l_{li}} \cdot S_{ci}$$

S_{ci} : ストローク l の i 番目の線分とその線分の重心間の類似度

文字間類似度

$$S_c = \min. |S_{c1}, S_{c2}, \dots|$$

$$S_{cjk} = \sum_l \left[\left(\frac{l_{li}}{\sum_l l_{li}} + \frac{l_{lj}}{\sum_l l_{lj}} \right) \cdot S_{st} / 2 \right] + \sum_k S_{ck} \quad (\text{但し, } i \leq j, k=1, j-i)$$

l_i : 入力文字(又は標準文字)の i 番目のストローク

- l_j : 標準文字(入力文字)の j 番目のストローク
 S_{st} : ストローク l_i とストローク l_j 間のストローク間類似度
 S_{ck} : 標準文字(入力文字)の K 番目のストロークの自己類似度
 S_{cn} : 文字間のストロークを対応させた組み合わせのうち n 番目の組み合わせの文字間類似度

3.2. 標準文字

前項の文字間類似度を用いて未知入力文字 X の識別を行うには標準文字を決めなければならない。標準文字は1つの文字カテゴリーに対して1個とするのが好ましいが、ストローク抽出や文字の認識の都合上複数のサブカテゴリー M_i をもつものもある。標準文字の作製は各カテゴリーごとに n 個のサンプルデータから抽出され正規化された特徴点を統計的に処理し平均化した値を標準文字の特徴点とする。

4. 実験

実験は電子技術総合研究所の自由手書き文字データベースの片仮名 ETL-1 を冒頭から100人分サンプルとして用いた。使用言語は FORTRAN 77, 使用システムは九州大学大型計算機センターの FACOM M-382 である。

4.1. 諸定数の決定

2.1の項で述べた文字の幅の閾値 w_a , 2.2項の滑らかさ S の閾値 C_a , 2.3項の定数 β , n_a は実験によって最適な値を求める必要がある。このうち滑らかさ S の閾値 C_a は前回の実験と同じ値0.8を使用する。又、文字の正規化をするために必要な定数 β , n_a は一括処理法¹¹⁾で使用している値をそのまま使用する。文字の幅の閾値 w_a は前回の実験でストローク抽出の正解率の良くなかった片仮名のカ(73%), ネ(51%), モ(73%)の3文字について $w_a = 3.1, 2.8, 2.5$ の値に対してストロークの抽出実験を行った。 $w_a = 3.1$ ではストロークの抽出が不十分なものがあり $w_a = 2.5$ では1分岐点が2分岐点に分離する等の細線化による弊害を生じた。 $w_a = 2.8$ ではストロークの抽出の正解率はカ(94%), ネ(98%), モ(92%)と概ね良好であったので $w_a = 2.8$ とした。

4.2. 認識実験

標準文字は各カテゴリーごとにサンプルデータ100個に対してストロークの抽出を行い目視検査によって正しくストロークを抽出したデータについてのみ統計処理を

行って作製した。入力文字と標準文字集合 $\{R_i\}$ から求めた文字間類似度 $S_c(X, R)$ が定数 S_a より大きいとき交叉領域での線索の結合が不相当と判断し、結合の組み合わせを変えて再び認識を行う。今回はこの組み合わせの変更は3回を限度とした。認識実験は各カテゴリーごとにサンプルデータ100個を用いて $S_a = 20.0, 10.0, 5.0$ の値に対して実施し、それぞれ第1位認識率の平均値92.3%, 92.6%, 93.0% 第5位までの累積認識率の平均値98.1%, 98.5%, 98.8%であった。表1に $S_a = 5.0$ における各カテゴリーの認識率を示す。

5. 結論と検討

本実験では入力文字を細線化したのちストロークを抽出し、抽出した特徴点を正規化して手書き文字の認識を試みた。認識実験は前項でみたように認識率93%と前回の74.5%と比べると著しく向上しているが満足できる結果でない。正しく認識できなかった原因を次の2つの場合に分けて検討してみる。

(i) ストロークの抽出が正しく行われている場合

- (1) 3.1項で定義した類似度はストロークの位置ずれや変形等の構造的変化に対して弱く不安定である。例えば表1の「フ」の誤認識文字はすべて「ク」である。
- (2) 入力文字が標準文字に比べて著しく文字のバランスを欠いたとき正規化によって返って類似度を悪くしている。例えば図9に示す「カ」。
- (3) 決定される文字カテゴリーよりも他の文字カテゴリーに類似している。例えばシのサンプルデータの中には我々が見てもシよりもツと読めるデータがある。

(ii) ストロークの抽出が正しく行われていない場合

- (1) 交叉領域での線索の結合、分離が正しく行われていない。例えば図10の「コ」のように縦の線索と下の横の線索が結合し1つのストロークを構成して「フ」と誤認識されている。
- (2) 2つ以上のストロークが重なり合い矩形状又は大きな1つの交叉領域となってストロークとして抽出不可能なことがある。
- (3) 2つのストロークが斜めに交叉しているとき1つの分岐点を2つの分岐点に分離して抽出してしまうことがある。特に細線化処理によって増加する。

以上が誤認識を生じる主な原因である。今回の認識実

表-1 標準文字数及び認識率

文字種	Mi	第1位 認識率 (%)	第2位 認識率 (%)	第5位ま での累積 認識率 (%)	文字種	Mi	第1位 認識率 (%)	第2位 認識率 (%)	第5位ま での累積 認識率 (%)	文字種	Mi	第1位 認識率 (%)	第2位 認識率 (%)	第5位ま での累積 認識率 (%)
ア	2	83	10	100	チ	1	83	7	98	ム	1	99	0	100
イ	1	99	1	100	ツ	1	84	5	97	メ	1	98	0	98
ウ	1	100	0	100	テ	1	91	8	100	モ	1	89	6	100
エ	1	96	3	100	ト	1	100	0	100	ヤ	1	89	0	95
オ	1	97	1	99	ナ	1	94	0	96	ユ	1	93	4	99
カ	1	88	4	96	ニ	1	99	0	100	ヨ	1	99	0	99
キ	1	93	0	97	ヌ	1	91	6	98	ラ	1	98	2	100
ク	2	90	7	100	ネ	1	99	0	99	リ	1	97	2	100
ケ	1	92	4	99	ノ	1	99	0	99	ル	1	99	1	100
コ	1	95	0	96	ハ	1	98	2	100	レ	1	97	1	100
サ	1	94	1	99	ヒ	1	96	2	99	ロ	1	99	0	99
シ	1	75	11	97	フ	1	82	18	100	ワ	1	94	3	99
ス	1	99	0	100	ヘ	1	91	7	100	ヲ	1	80	13	98
セ	1	87	6	96	ホ	1	100	0	100	ン	1	81	10	99
ソ	1	84	12	98	マ	1	96	3	100		計	平均		平均
タ	2	90	2	96	ミ	1	100	0	100		49	93.0%		98.8%

$S_a=5.0$ Mi : サブカテゴリの種類数

表-2 認識実験における誤認識文字の例

自由手書き片仮名(ETL-1, 100個/1文字)	
文字	誤認識文字(個数)(2個以上誤ったもの)
ア	タ(10) フ(4) ク(3)
シ	ラ(8) ン(4) ツ(3) ヲ(3)
ソ	リ(5) ン(4) ツ(3) ク(2)
ツ	ソ(4) ク(4) ワ(3) シ(2)
ン	サ(10) シ(3) ニ(3)

験で得た認識率より高い認識率を得るためにはこれらの原因を考慮して改善しなければならない。以後対策を検討してみる。

(i)の(1), (2) 現在使用して識別関数としての文字間類似度はパターン整合的性質であり、これにストロークの方向や曲率等の位相構造的特徴を取り入れた識別を行えば解決できる問題である。

(i)の(3) 文字パターンの形状からだけでは識別は困難であり文字の前後の文脈を用いて言語处理的¹⁰⁾

に解決しなければならない。

(ii)の(1) 2.1項で滑らかさSは0.8と固定した閾値で線素の結合分離を一意的に判断しているがファジィ集合を用いてSの閾値を場合によっては変化させる。

(ii)の(2) 矩形や交叉領域の形状を抽出し、その形状から文字の重なりあった状態を推論し複数のストロークの集まりと仮定し類似度を求める。

(ii)の(3) 2つの分岐点が互いに近距離で存在するとき、その形状によっては1つの分岐点に併合する。

以上が今回の認識実験に対する問題点及び改善すべき方向や対策についての検討である。今回の実験で用いた細線化や正規化の処理によって認識率は改善できた反面この処理によって(i)の(2), (ii)の(3)等の問題が生じた。単に問題点を分析しそれに対する個々の対策を施すだけでは認識システムはますます複雑化し処理に要する時間も増加して効率の低下を招く。システム全体にわたる総合的対策が望まれる。

最後に本手法をひらがなや漢字に適用する場合の諸問

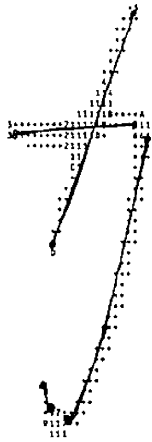


図-9 片仮名「カ」。

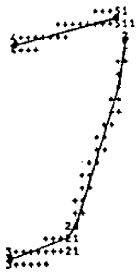


図-10 片仮名「コ」。

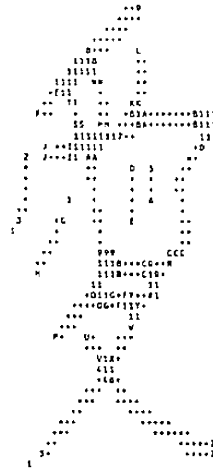


図-11 漢字「愛」。

題について考えてみる。ひらがなは片仮名に比べて曲線の部分が多いが基本的には類似している。しかし漢字の場合に特徴的なことはカテゴリーの多さと個々の文字の複雑さ(ストローク数の多さ)である。例えば図11の漢字「愛」の場合、ストローク数は15本であり3.1項で述べた文字間類似度を求めるとき入力文字と標準文字のストロークの組み合わせは15の階乗 $=1.3 \times 10^{12}$ となり、これらの組み合わせを全て計算することは実用上不可能である。もっと効率的に処理できる新しいアルゴリズム¹⁵⁾が必要となる。又2.2項で交叉領域での線素の結合の組み合わせも多くなり、ここでもっと効率的に処理できるアルゴリズムが要求される。これらは漢字の複雑さから派生した問題であるがカテゴリーの多さから派生する問題もある。未知の文字と認識する場合、多くのカテゴリーの

中からいかにやく効率的にカテゴリーを決定するかは重要な問題である。これらのことは今後の課題である。

謝 辞

電総研手書き文字データ・ベータ利用を御承認頂いた情報処理システム技術研究組合各位の皆様には謝意を表します。

参 考 文 献

- 1) 梅田, 安田: 「メッシュ・ベリフェラルパターンによる手書き漢字の分類」, 信学技報, PRL 79-26, 1979-7
- 2) 内藤, 淀川: 「手書き漢字のストローク密度関数による大分類」, 信学技報, PRL 79-3, 1979-1
- 3) 赤松, 小森: 「構造集積特徴による手書き漢字の類別実験」, 信学技報, PRL 80-25, 1980-7
- 4) 稲永: 「ストローク抽出による手書き文字の認識」, 九工大研究報告 pp.15-25 (85)
- 5) Hilditch, C. J: 「Linear skeleton from square cupboards」, In Machine Intelligence 6 (1969)
- 6) 田村: 「図形の細線化についての比較研究」, 情報学会IP研究(昭50-06)
- 7) 稲永: 「拡大縮小法による線図形の細線化」, 電気四学会九州支部大会論文集, pp.194 (昭60)
- 8) K. OSADA, T. HITAKA: 「Software for Recognition of Hand-Printed Japanese Characters」, Proc. Ints Comp. Symp. 80 (1980)
- 9) 稲永, 日高: 「線図形におけるストローク抽出アルゴリズム」, 電気四学会九州支部大会論文集, pp. 806 (昭57)
- 10) 甘利: 「パターン信号の特徴空間における正規化理論」, 信学誌, 49, 7, pp.1342-1350 (昭41-07)

- 11) 長田, 日高: '一括処理法による手書き文字の認識', 信学論(D), J66D, 2, pp.151-158 (昭58-02)
- 12) 長田, 日高: '一括処理法による手書き文字認識', 信学技報, PRL80-74 (1981-01)
- 13) 豊福, 稲永, 日高: 'ストローク抽出による手書き文字の認識', 電気四学会九州支部大会論文集, pp.472 (昭58)
- 14) 新谷, 目黒, 梅田: '認識情報及び単語, 文節情報を利用した文字認識後処理', 信学論(D), pp.1348-1355 (昭59-11)
- 15) 稲永: '手書き文字における文字間ストローク対応アルゴリズム', 電気四学会九州支部大会論文集, pp.392 (昭61)