

接線ベクトル列を用いたオンライン手書き文字の認識*

藤原 塩 和** 池田 克 夫***
富 永 善 治**** 清 野 武***

Abstract

We are daily using over 2000 characters in Japan. Hence, to build computer utility in our country, it will be inevitable to read directly many characters. It seems that many researches in the area of character recognition are font/type dependent, which may be insufficient to the practical applications in this country.

We studied General Contour Vector Sequence (GCVS) method which we believe applicable to almost every type of characters, and attained fairly good recognition score. This paper describes the principles of the shape analysis of strokes by GCVS method, the analysis of positional relations between strokes by GCVS method, and the recognition procedures.

1. はじめに

計算機の発達と共に、計算機と人間とが対話形式で情報処理を行うシステムが一般化し、人間の意志をできるだけ自然な方法で計算機に伝えることが要求されている。現在のところ大部分は情報問い合わせ端末としてキーボードを用いているが、この方法では文字数が少数に制限されているためにおのずから入力に限度がある。我が国では、多種の文字（数字、ひらがな、カタカナ、漢字、アルファベット）を混用し、その数は日常で用いるものだけで2,000種をこえる。これら日本語で使用するすべての種類の文字を対象とするような処理（たとえば、日本語の文章の編集、校正）には、適当な入力方法が必要である。その一つとして、和文タイプと同様の形式のものが、すでに開発・実用化されている。しかし2,000種以上の文字をキーボードにより入力するためには、オペレータの訓練を必要

とし一般的でない。一般の人が自由に日本語を入力する方法の一つとして手書き文字の認識が望まれている理由がここにある。

手書き文字の認識については、すでに種々試みられてきているが、それらのほとんどは、文字の種類別のものであった。本論文では、多種類の文字を統一的に扱う手法について報告する。

本研究では、すでに書かれているものを対象とするのではなく、手書きを行っている過程から情報を取り込むオンライン方式の認識について考える。

本論文で用いる手法は、人間が文字読み取りの際に重視しているのではないかと考えられる曲線の曲りに注目して形状を接線方向列のグラフに変換し、そのグラフのマッチングを行うものである。マッチングにはダイナミック・プログラミングの考え方を取り入れている。

2. 手書き文字の特徴抽出

手書き文字のように描かれるパターンのばらつきが多い場合に問題となるのは、何を特徴パラメータに選ぶかということである。従来のオンライン方式の認識方法には次のようなものがある。

(A) 文字を手書きする際の筆点運動を直交座標成分 $X(t)$, $Y(t)$ に分解した一次元波形と考え、その波形

* On-line Recognition of Handwritten Character by General Contour Vector Sequence Method by Siokazu FUJIWARA (Nippon Telegraph and Telephone Public Corp.), Katsuo IKEDA, Takeshi KIYONO (Department of Information Science, Faculty of Engineering, Kyoto University) and Yoshiharu TOMINAGA (Mitsubishi Electric Co.)

** 日本電信電話公社

*** 京都大学工学部情報工学教室

**** 三菱電機(株)

のフーリエ係数を用いる¹⁾。

(B) ストロークを8方向で量子化したベクトルのつながりとして近似し、このつながりから11種の基本ストロークに分類する。文字は基本ストロークの組み合わせと考える²⁾。

(C) ストロークを尖点で分割し、それぞれの部分を6等分しその弦を8方向で量子化した直線で符号化する³⁾。

(B)の8方向のベクトルで近似する方法は、漢字、カタカナを対象としているために構成ストロークの形状が比較的単純である。そのため、8方向のベクトルで表現すると、比較的単純なつながりとして表わすことができ基本ストロークへのマッチングが容易である。ひらがな、アルファベットのように、ストロークの形状が複雑なものを含めると、ベクトルのつながりが複雑となり、マッチングが難しくなる。

(C)の6本の弦で近似する方法では、アルファベットを対象としている。アルファベットの場合には、尖点をはっきりしているが、カタカナ、ひらがなの場合には、尖点が必ずしも明確でなく分割があいまいとなる。また8方向で近似するのは個人差を吸収できるという利点はあるが、細部が無視されることがある。

以上のように、これらの方法は対象とする文字種別を限定しその各々の特徴を生かしたものであり、すべての種類の文字に対してただちに有効であるとは言い難い。

さて、改めて文字とは何かを考えてみると「さまざまな形状をした曲線を組み合わせでできている記号」ということができる。従来の方法は「さまざまな形状」ということを捕えるのに焦点がおかれ、形状が同じときに位置関係が補助的に用いられてきた。特に、画数の多い漢字では、位置関係だけで一意的に決定できる文字が非常に多い。本論文では位置関係を中点を結ぶベクトルの方向として捕え、形状と同様に重視した。

画数の少ない文字では、位置関係よりも形状の方が重要な特徴となる。特に一画の文字では、形状だけで決定しなければならない。形状を捕えるのに注意しなければならない点は「手書き」という条件からくる個人差をいかに吸収するかということである。個人差には次の二つのタイプがある。

(1) ストロークの部分的な形状の差

(2) 文字の構成ストロークの形状は、異なっているが、ストロークの組み合わせとしてみると同一の文字となる場合。

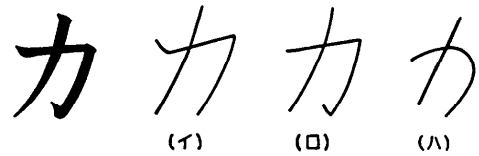


Fig. 1 Examples of minor shape difference

(1)の部分的形状の差というのは Fig. 1 に示すものである。

Fig. 1 (イ)は書き始めに「おさえ」がある。

(ロ)は書き終りに「ハネ」がある。

(ハ)はゆるやかなカーブを描き屈折点が存在しない。

このような千差万別の形状を同一と見なすためには屈折点の個数、 x, y 成分の極値の個数、およびそれらの種類といった局所的な特徴の捕え方でなく、「右へ進み、右まわりに回転し、左斜めに進む曲線」といった概括的な捕え方をする必要がある。その反面、後述する例のように局所的な部分まで調べる必要がある場合も存在する。たとえば「し」と「レ」では尖点の有無が重要な相違点となる。そこで、最初は概括的に形状をとらえ、さらに必要に応じて細部を調べていくといった段階的な方法が有効である。

(2)のストロークの組み合わせが問題となるのは、Fig. 2 に示すような場合である。このような場合、辞書の項目を増す方法も考えられるが、組み合わせが多くなり、項目数が増加するので辞書の検索に不利である。ストローク「L」-「L」-「L」が似た形状であると考えの方が自然である。そこで形状間に類似度という概念を導入し、文字全体のストロークとして比較を行う。

ストロークの形状を概括的に解析するのに本論文で用いた方法は、(B)のベクトルのつながりとして近似する方法、(C)の6本の弦で近似する方法にヒントを得て、(A)の一次元波形を比較する方法を利用したものである。形状を解析し特徴を抽出するためには生データに近い情報を用いる必要がある。この観点からみると(A)の x, y 成分の波形を比較するのは良好な方法といえる。しかし、 x, y 成分の波形は、情報密度の

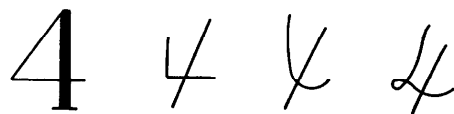


Fig. 2 Examples that decision is made by combination of strokes rather than shape of strokes

小さいものと言える。つまり x, y 成分のグラフの極値は、屈折点を表現してはいるが、単に x の増加、減少だけでは入力波形との対応が悪く、 x, y 両成分のグラフを同時に考慮しなければならない。ストロークの特徴は、 x, y 成分の増減よりも、むしろ増減の変化（ストロークの変曲に対応する）にあると考えられる。この特徴を抽出するために次のような方法を用いた。ストロークの各点における接線の方向を縦軸に、時間を横軸にとったグラフを考える。認識はこの一次元波形のマッチングにより行う。マッチングは、ダイナミック・プログラミングの手法を用いる。詳細は次章で述べる。

3. 形状の解析

本章では、ストロークの形状を概括的に解析するのに接線ベクトル列を用いた方法 (General Contour Vector Sequence: GCVS 法) について述べる。形状を認識する際に概括的に捕えることは重要であるが、局所的な部分を調べる必要がある場合も生じるので、もとの入力情報はできるだけ保存しておく必要がある。

3.1 接線ベクトル

われわれがストロークの形状を観察するときには、「どちらの方向へ曲っているか」ということに注意が払われると考えられる。そこで入力座標点列を接線の方向列に変換する。7. で述べるフィルタリングの前処理がなされた入力点列を $P_0P_1 \dots P_n$ とし、 P_{k-i}, P_{k+i} で作られるベクトルを接線ベクトル (General Contour Vector: GCV) とする。 i は、小さ過ぎると平滑化の効果が少なく、また、大き過ぎると尖点、コーナなどの特徴点を失うおそれがある。本報告書では $i=2$ としたが一ストローク当り $n=50 \sim 200$ 程度であるのでベクトルの長さがストローク長の $5 \sim 10\%$ となっている。

$$GCV_k = \overrightarrow{P_{k-2}P_{k+2}}$$

GCV の方向は Fig. 3 で示す方向量子で符号化する。方向はできるだけ忠実に表現するために 15° 間隔で 24 方向に量子化した。GCV の値域は、方向だけでなく回転角をも情報として含めるために、 $-\infty \sim +\infty$ の値をとるようにした。

3.2 接線ベクトル列の決定

入力座標点列を前節で述べた GCV に変換し、Fig. 3 の方向量子数で符号化して順番に並べたものを GCVS (GCV Sequence) と呼ぶ。GCVS は次の規則で決定する。

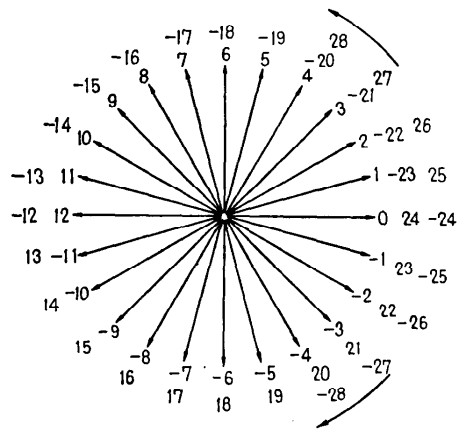


Fig. 3 Code for stroke direction

1. GCVS の初期値は (-11) から (12) の間で選ぶ。
2. その他の GCVS の符号は、隣り合う GCV の差の絶対値が小さくなるように選ぶ。

急激にストロークの方向が変化する場合、たとえば「 $\#$ 」の第 1 ストローク「 $\#$ 」の折り返しの部分では右まわりか左まわりかの区別がつかないので二通りの GCVS を認めることにする。Fig. 4 は「 $\#$ 」について GCVS の方向数を縦軸に、時間を横軸にとって入力情報をプロットしたものである。

GCVS はストロークの形状をほぼ忠実に実現している。たとえば、Fig. 4 の点 A', B' のように不連続な点は、入力図形の尖点に対応する。点 B'' から点 C までの縦軸の変化量は 24 以上であることから入力図

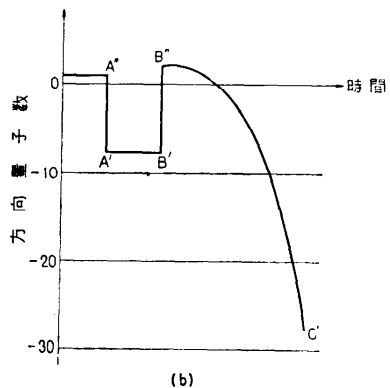
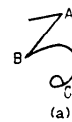


Fig. 4 An example of GCVS curve

形は1回転していること、また B^*C' の後半ほど、傾きが大きいことから、入力図形の最後に小さなループが存在することがわかる。座標点列は、一定距離間隔ごとに入力しているので、横軸は長さを表わしている。以上のように GCVS は、入力図形の尖点、回転方向、回転角、長さ、曲率の情報を含んでいる。

3.3 GCVS の符号化

ストロークの形状の認識はGCVSのマッチングにより行う。横軸はストロークの長さを表わすからGCVSのグラフの横幅は一定でない。そこで長さに関して正規化する。

経験的にはストロークの長さの10%以上の部分に特徴点が現れるようである。そこでGCVSを等間隔に11点をサンプルし、極値の情報(尖点や屈折点に対応する)を含ませるために近傍の極値でおきかえた代表値を用いる。またストロークの長さの5%以下の部分に重要な意味を二重に持つことはないので、このおきかえは一意的に決定できる。すなわち次の規則で、GCVSの11点の値を選択して、ストロークの概形を示すパラメータ(Sampled GCVS: SGCVS)とする。

ストロークのGCVSが $n+1$ 個のGCVから構成されているとする。

$$m = \lceil n/10 \rceil$$

$$a = \lceil (n \bmod m)/2 \rceil$$

とすると

(1) $a+(k-1)m$ ($k=1, 2, \dots, 11$) の11のGCVを選ぶ。 a は端数をストロークの最初と最後の部分にふりわけける効果がある。

(2) $k=1, 2, \dots, 9$ について区間 $[a+km-5m, a+km+5m]$ でGCVSが極値を一つ持つときは、 $a+km$ 番目のGCVを極値の値でおきかえる。極値が複数個存在する場合には、もとの値のままとする。

Fig. 4の「る」はこの符号化により次のようになる。

$$\text{SGCVS(る)} = 1, 1, 8, 8, 2, 1, -1, -5, -9, -15, -28$$

3.4 ストローク間のGCVSの距離

二つのGCVS G_1, G_2 の距離を定義するための、一番簡単な評価式として

$$\int |G_1(x) - G_2(x)| dx \tag{1}$$

が考えられる。しかし、ストロークの形状に個人差があることを考えると(1)式では評価しきれない場合があり、ダイナミック・プログラミングの考え方⁴⁾を用

い、 $t, v(t)$ とを使って異なった点における G_1 と G_2 の比較を行う。すなわち

$$\min_v \int |G_1(t) - G_2(v(t))| dt \tag{2}$$

(ただし、 v は t の単調増加関数)

を G_1 と G_2 の距離とする。

(2)式で評価するとき次の欠点がある。

イ) 計算時間がかかる。

ロ) (2)式では「最適」のマッチングがとられるために、 G_1 は G_2 の一部分とのみマッチさせられることがある。

(イ)、(ロ)共にすべての場合を調べることにより起因するので適当な条件を加えることにより除くことができるはずである。また、「最適」のマッチングをとる必要はなく、形状を分類するのに十分な距離を定めることができればよいのであり、著しく隔った点を比較することは、かえって悪い結果を生じる。

手書き文字のストロークは接線ベクトルの方向と相対的な長さに関してある程度の変動を伴うのが常である。すなわち、入力されたストロークのGCVSは単なる一本の曲線ではなく、方向と長さ(時間)の双方について一定の幅を持っていると考えるのが妥当である。

しかし、すべての場合についてこれを適用すると、一致する範囲が大き過ぎる。(1)式の評価で問題となるのは屈折点のずれによるミスマッチであるので、屈折点の近傍での位置あわせに上の考え方を用いるのが適当である。

接線ベクトルの方向の変動については ± 1 の差は無視する。また、長さに関してはストローク長の $\pm 20\%$ を限度とした。これは(2)式の $v(t)$ を屈折点に対応するように選び、かつ Fig. 5 のように値域に制限を加えたことに対応する。

アルゴリズムの詳細は次に示すとおりである(Fig.

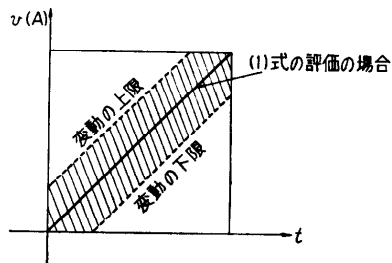


Fig. 5 The domain of $v(t)$ in dynamic programming

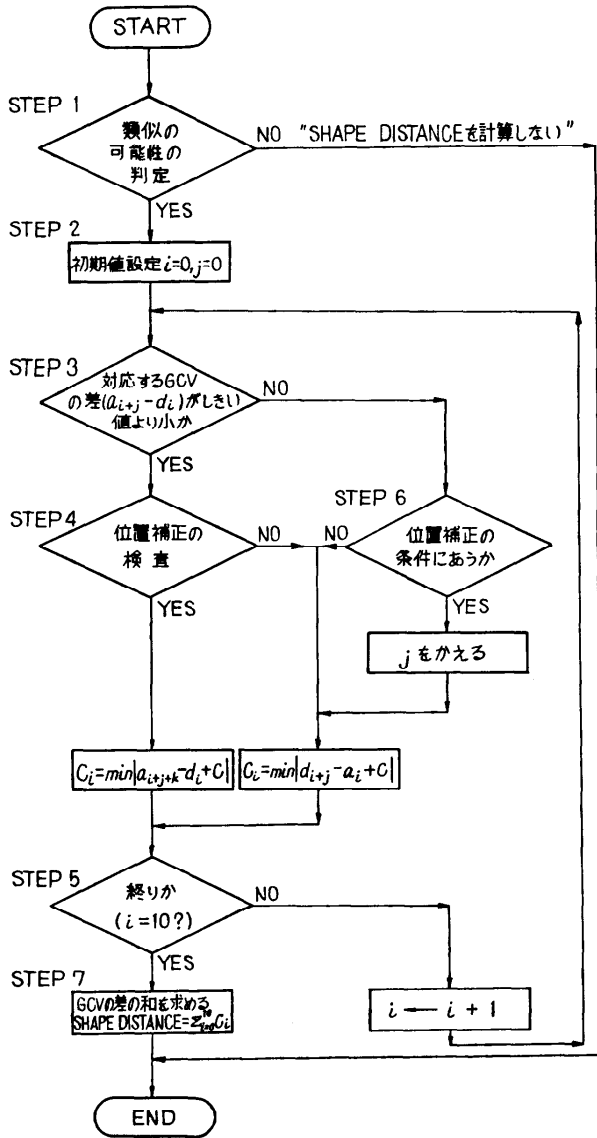


Fig. 6 Calculation of stroke distance

6 参照.

入力の手書き文字 A, 比較する辞書の手書き文字 D の GCVS が次のように符号化されているとする.

$$SGCVS(A) = a_1 a_2 \dots a_{10}$$

$$SGCVS(D) = d_0 d_1 \dots d_{10}$$

(段階1) 類似性の判定

二つの形状の類似する可能性があるかどうかを調べる. すなわち; (3)式を満たす i が K_2 個以上存在するとき, 段階2以下の手順で二つのSGCVSの距離を

求める. K_2 個未満のとき, 二つのストロークは類似していないとする ($K_i, i=1, 2, \dots, 6$ は定数とする).

$$|a_i - d_i| \leq K_1 \quad (3)$$

(段階2) 初期値設定

初期値を設定する. すなわち; $i=0$ (位置), $j=0$ (位置の補正) とする.

(段階3) GCV の差の計算

対応する点におけるGCVを比較する. すなわち; GCVの差が小さいとき ((4)式を満たすとき) 段階4で最適マッチングをもとめ, 大きいときは段階6の位置ずれ処理へ行く.

$$SA = \min_c |a_{i+j} - d_i + c| \leq K_3 \quad (4)$$

ただし, $c: -1, 0, 1$ (方向差 ± 1 を無視する)

(段階4) 位置補正の検査

位置補正を行っている場合に適性の検査を行う. 位置補正を復元するためにはよりよいマッチングを条件とする ((5)式).

$j=0$ のときは $c_i = SA$ として段階5へ行く.

$j < 0$ のときは $K = 1$

$j \geq 0$ のときは $K = -1$ とする.

$$|a_{i+j+k} - d_i| + K \leq |a_{i+j} - d_i| \quad (5)$$

(5)式が成立するときには

$$c_i = \min_c |a_{i+j+k} - d_i + c|$$

成立しないときには

$$c_i = \min_c |a_{i+j} - d_i + c|$$

として段階5へ行く.

(段階5) 終了条件判定

終了条件を判定する. すなわち; $i=10$ のときは段階7(終了処理)へ行く. $i \neq 10$ のときは $i+1 \rightarrow i$ として段階3(GCVの差の計算)へ行く.

(段階6) 位置補正

近傍と比較してよい条件(辞書または入力に曲率の大きい点が見られたこと)を判定する. (6)式のいずれをも満たさず位置補正を行わないときには,

$$c_i = \min_c |a_{i+j} - d_i + c|$$

として段階5へ行く.

$$\left. \begin{aligned} |d_i - d_{i+1}| \geq K_5, & \quad |d_{i-1} - d_i| \geq K_5 \\ |a_{i+j} - a_{i+j+1}| \geq K_5, & \quad |a_{i+j-1} - a_{i+j}| \geq K_5 \end{aligned} \right\} \quad (6)$$

(6)式のいずれか一つでも満たすときには最適マッチ

ングを行う点を求める ((7)式を満たす l).

$$|a_{i+l} - d_i| < K_3 \quad (7)$$

(ただし, l は $-2, -1, 0, 1, 2$ のいずれかで $|l|$ の小さいものを選ぶ. $|l|$ が等しいときには $l > 0$ のものを選ぶ.)

$j=l$ として

$$c_i = \min_c |a_{i+j} - d_i + c|$$

とする. 段階5へ行く.

ただし $i+j < 0$ のときには $i+j=0$

$i+j > 10$ のときには $i+j=10$ とする.

最適マッチングが得られないとき ((7)式を満たさないとき)

$$c_i = SA$$

として段階5へ行く.

(段階7) 終了処理

GCV の差を求める. すなわち; 段階2~段階6で求めた c_i ($i=0, \dots, 10$) について,

$$\text{SHAPE DISTANCE}(A, D) = \sum_{i=0}^{10} c_i$$

を求める.

$$\text{SHAPE DISTANCE}(A, D) \leq K_6 \quad (8)$$

のときストロークの形状が同じであると判定する.

$K_1 \sim K_6$ は次の値を用いた. $K_1=3(45^\circ)$, $K_2=3$ (ストローク長の30%), $K_3=7(105^\circ)$, $K_4=3$ (よりよいマッチング), $K_5=6$ (曲率大: 90°), $K_6=20$ (一点当たり 30°).

本章で述べた方法を GCVS 法と呼ぶことにする. GCVS 法により, どの程度個人差を吸収できたかを Fig. 7 に示す. なお図で×印のついているのは(8)式を満たさなかったものである.

3.5 複雑度

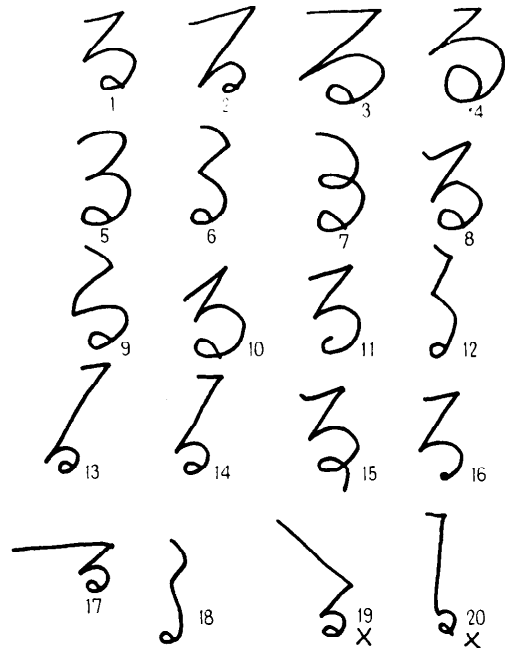
ストロークの形状を GCVS 法で比較する際に辞書のすべての形状と比較するのは無駄が多い. GCVS 法で比較する前にある程度範囲を限定できれば都合がよい. ストロークの複雑さを与える指標としてストロークの回転量が考えられる. GCVS を符号化したものは, GCVS のグラフの極値をすべて含むので, この回転量は次式で求められる.

ストローク A の GCVS を符号化したものが a_0, a_1, \dots, a_{10} のとき

$$\text{COMPLEXITY}(A) = \sum_{i=0}^9 |a_{i+1} - a_i| \quad (9)$$

3.6 形状の類似度

GCVS 法では全く別の形状となるが, 個人差とい



(1)を辞書にしている。(19),(20)はそれぞれ斜め, 垂直の線が長すぎて方向も異なったためマッチングがとれなかった.

Fig. 7 Examples of patterns which are taken for as the same stroke

う立場からみる場合, 似た形状と見なした方がよいときがある. たとえば, 「4」-「み」の第一ストローク「L」と「L」である. これらは GCVS 法による SHAPE DISTANCE は非常に大きい, 文字として書かれるときにはあたかも同じ形状のように用いられる. 形状が似ているということを数値化するのにここでは二次元座標上の点としてストロークの形状を符号化しておく. すなわち, 類似すると見なすべき形状に対しては近傍の点となるように座標の値を視察によって与え, これをストローク A の形状といい, $A(x_a, y_a)$ と表わす. ストローク A, B の形状が $A(x_a, y_a), B(x_b, y_b)$ のとき類似度は次式で表わす.

$$\text{SIMILARITY}(A, B) = |x_a - x_b| + |y_a - y_b|$$

ただし $\text{SIMILARITY}(A, B) \neq 0$ で $(x_a, y_a) = (0, 0)$ のとき $\text{SIMILARITY} = \infty$ とする.

4. 位置関係の解析

本論文では, 連続して描かれる2つのストロークの中点を結んでできるベクトルの方向によって位置関係を示す. 方向は Fig. 3 で示した24方向の方向量子で近似する. 位置関係にも個人差があることを考慮して従来4または8方向で近似することが行われていた

が、方向の差を正確に測定するためには量子化誤差を小さくした方がよいので 24 方向を用いる。

このベクトルで位置関係を表わす方法は次の二つの大きな欠点がある。

(イ) 方向が大きいくずれることがある。

中間間の距離が小さいとき、または 2 つのストロークが交差するときには方向が大きいくずれることがある。そのために 2 つの方向が近いかどうかを判定するには注意を要する。ここでは次式で評価する。二つの方向を V_1, V_2 とすると

$$PSA = \text{MODULO}(|V_1 - V_2|, 12)$$

$$\text{POSITION DISTANCE} = W * PSA$$

W は中間間の距離、ストロークの交差関係、PSA の値により定まる重みで、実際には辞書を用いて文字ごとに与える。

(ロ) 位置関係が筆順に依存する。

連続して描かれるストロークの midpoint を結ぶベクトルを考えているので、筆順が異なれば、当然位置関係も異なってくる。この欠点に対しては異なった筆順の書き方だけ辞書を用意しておく。

5. その他の特徴パラメータ

GCVS は、はね、おさえ、尖点といった局所性を軽視して概括的な形状を認識するために、この方法による形状解析だけでは、非常によく似た形状を区別できない場合がある。しかしストロークの分類という点では十分満足のいくものである。区別できない例を次に示す。

(イ) 屈折点の有無を認識する必要のあるもの

$$\text{しーレ U-V}$$

(ロ) 長さの情報を必要とするもの

$$\text{のーんーhーp, bーf, aーdー9}$$

(ハ) その他

$$\text{3ーろ, ろーn, 2ーZ, 1ーノ}$$

これらの形状を区別するためには、他の補助的な情報が必要である。

5.1 尖点の検出

GCVS が V_i ($i=1 \dots n$) のとき

$$c = |V_{i-3} - V_{i+3}| > \alpha \quad (10)$$

を満たすとき i の点を尖点とする。通常、上式を満たす i は連続して現れるが、その場合には、 c の最大となる i の点を尖点とする。

5.2 ストロークの長さ

「土」-「士」の文字の差は第 2, 第 3 ストロークの長

さの差である。こうした文字間の区別には長さに関する情報が必要である。ストロークは一定距離間隔でサンプリングした点列によって入力しているので入力点数が長さを与える。

5.3 ストロークの始点・終点の座標

「力」-「刀」, 「土」-「工」のような文字の区別には、第 1, 第 2 ストロークの始点の位置関係が決め手となるので、ストロークの始点・終点の座標を記憶しておく。

6. 認識方法

辞書としてストロークの形状解析に用いる形状辞書と文字認識のために用いる形状および位置関係を示した面数ごとの文字辞書を用いる。

文字の認識は次に示す手順で行う。

(1) オンラインによるデータ入力

ランドタブレットにスタイラスペンを用いて、文字を書くことにより、ペン先が触れたタブレットの座標が一定時間間隔で得られる。この座標点列に、フィルタリング(次章で述べる)の前処理を行う。

(2) 位置関係の解析および面数と位置関係による文字の一次選択

連続して描かれるストロークの位置関係を符号化して、面数により辞書を選び、位置関係の一致する文字を選び出す。

(3) ストロークの形状決定と形状による二次選択

ストロークの形状は、文字の構成ストロークすべてについて解析する必要はなく、(2)で選ばれた文字を一意的に決定するのに必要なストロークについて GCVS 法により形状辞書を用いて形状解析を行った後、文字辞書の中のストロークに対して形状の類似度を判定してストローク形状の一致するものを認識結果の文字とする。

(4) 同じ位置関係や形状を持つ文字の区別

二次選択で一意的に定まらない文字(たとえば「I」-「工」-「土」-「士」)については、ストロークの長さ、始点・終点の座標など個々の特徴を調べて区別を行う。

7. 前処理

7.1 ウィンドウフィルタリング

ランドタブレットからの座標をすべて入力している、同じ座標値のものが多いので、一定距離離れた点のみを入力する。その方法は、最初の入力点 P_0 を入

かし、 P_1 を原点として Fig. 8 の領域を定義する。領域 A に入った点は P_0 に近すぎるので無視し、領

誤認識した入力パターン	認識結果	誤認識し理由
ノ	1	ノの曲りが少ない。
Q	a	分離ルーチン(a, O)のプログラムの不備 現在のところaとOの判定は $k_2/k_1 < 0.4$ のときaとしている。 $\overline{A} \overline{C} \overline{B} \overline{D}$ OがOのように書かれたため。
Q	2	Qの最初の部分が入力されず"Q"となつてをため。
ρ	n	Pとnの分離ルーチンの不備。 $k_1/k_2 > 0.5$ のときPとしている。 $\overline{P} \overline{1} \overline{k_1} \overline{k_2}$ $k_2/k_1 = 0.436$ となった。
ル	f	"ル"の分離ルーチンの不備。 $a/b < 0.8$ のときfとしている。 $\overline{a} \overline{b}$
2	Z	"2, 正"の分離ルーチンの不備。
し	1	"し"の曲りが少ないため。
q	a	SHAPE DISTANCE 2 によりqの後半が無視された。
ク	7	分離ルーチンの不備。 "7"のストロークで左向きに方向変化があるかどうかで判定している"7"と"⑦"の部分 が直線に近いときは"7"と判定される。
う	ラ	"う"が水平であり、かつ"う"が"ラ"より "ラ"に近い形であるため。
子	子	"子"が"了"と認識された。
丁	T	"丁"のハネが小さかったため"丁"が "し"と認識された。
コ	コ	ユとコの分離ルーチンの不備 $\frac{CB}{AC} > \frac{1}{3}$ をユの判定の基準 $\overline{ユ} \overline{C} \overline{B}$ としている。
れ	わ	"れ"の最後の曲りが小さかったためと判定された。

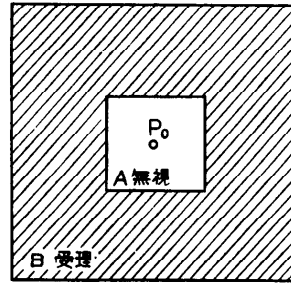


Fig. 8 Window for input data

域 C に出た点は雑音とみなす。次の入力点 P_1 は領域 B に入った点とする。 P_1 を原点として同様の操作を繰り返す。

7.2 平滑化

ウィンドウフィルタリングにより得られる入力点列 P_1, P_2, \dots, P_n は必ずしもなめらかな曲線とならないので、次の式で計算した値を改めて入力点とする。

$$P_n' = (P_{n-1} + P_n + P_{n+1} + P_{n+2}) / 4$$

8. 実験結果

以上述べた方法で7人の被験者に対して 1, 2, 3, 12 画の文字について実験を行った結果は次のとおりである。なお前処理は FACOM-R で、認識処理は HITAC 8350 の計算機を使用した。

		認識率	時間
1画	61種	97%	300 msec
2画	83種	98%	450 msec
3画	60種	99%	600 msec
12画	171種	99.5%	1000 msec

実験文字数は各文字について7字ずつである。誤認した文字例を Fig. 9 に示す。

一	子	千	第1ストロークが右下りに書かれた。
三	ツ	川	第1,2ストロークが"し"と認識された。
画	寸	さ	"し"のハネが小さく"し"と認識され"一"、 "し"、"へ"の組合せよりさとなった。 長さも判定に入れていないのが問題。
	子	千	"し"の曲りが小さかったため。

Fig. 9 Examples of ill-recognized characters

9. あとがき

GCVS法により各種の文字を統一的に処理できることを示した。特に、ストロークの形状を認識する際に、手書きに起因する変動分を除去し辞書の項目を少なくするために、制限付きのダイナミック・プログラミングの手法が有効であった。形状の類似度を判定するパラメータとして二次元座標の点と定めたことも文字の辞書の項目を少なくするのに役立っているように考えられる。ストロークの形状の解析と従来余り取り上げられなかったストロークの位置関係の比重も文字の画数によって大きく変化することがわかった。12画以外の4画以上の文字についての実験など、実用性をたしかめる実験が残されているが、本手法は相当広範囲の応用が期待されると考えている。

誤認した文字は、どちらかという人間でも単独の文字として読み誤まる可能性のあるものであった。

認識率をさらに向上させるためには文脈の導入などを図る必要もあるが、本論文では、一応別個の問題として、触れなかった。

末筆ながら、本研究に協力された情報工学教室の各位に感謝いたします。

参考文献

- 1) 荒川弘照, 増田 功: 手書き平仮名文字のオンライン認識, 情報処理学会大会予稿集 No. 33 (1974)
- 2) 寺井秀一, 中田和男: オンライン手書き片仮名文字認識, 情報処理 Vol. 13, No. 3, pp. 145~153 (1972).
- 3) G. M. Miller: ON-LINE RECOGNITION OF HAND-GENERATED SYMBOLS, Proc. AFIPS Vol. 35, pp. 399~412 (FJCC 1969).
- 4) 磯道義典, 小川辰男: 動的計画法によるパターンマッチング, 情報処理 Vol. 16, No. 1, pp. 15~22 (1975).
- 5) T. T. Kim & D. C. Evans: SYNTAX DIRECTED ON-LINE RECOGNITION OF CURSIVE WRITING, Technical Report 4-8, Univ. of UTAH (July 1968).
- 6) M. I. Bernstein: HAND-PRINTED INPUT FOR ON-LINE SYSTEMS, SDC Technical Memorandum (April 14, 1968).
- 7) G. F. Groner: REAL-TIME RECOGNITION OF HANDPRINTED TEXT, The RAND Corporation (May 1968).
- 8) 富永善治: 特徴ストロークによる手書き文字のパターン解析, 京都大学修士論文 (1973).
(昭和50年4月7日受付)
(昭和50年8月6日再受付)