オンライン手書き片仮名文字認識*

寺井秀一** 中田和男**

Abstract

This paper describes two subjects regarding the on-line recognition of hand writing characters.

Firstly, the necessary bandwidth to transmit a hand writing process is studied. It conclues that a hand writing process can be transmitted through 320 bits/sec or less, and it means that seven or eight input terminals of hand writing can be transmitted simultaneouly with a telephone line of 2,400 Baud.

Secondly, an algorithm of the on-line recognition of hand writing Japanese KATAKANA characters is developed mainly based on the extraction of numbers, types and certain characteristics of strokes. Informations on time sequence and relative position among strokes are considered if necessary.

The recognition experiments indicate 95% correct recognition is achieved without any preinstruction or restriction on the way of writing, and a further improvement of the result, almost 100%, is possible with several instructions which do not force any essential restrictions on the way of free hand writing. A recognition time is estimated less than 50 m sec a character with a mini-computer HITAC-10.

1. まえがき

データ通信回線が開放されるに従って計算機をオンラインで使用する各種の情報処理システムが広くゆきわたり，それに伴って情報問い合わせ端末として，簡単でしかも柔軟性に富んだ手書き文字入力の必要性が増してくると思われる．従来，文字の自動読み取りといえば OCR (Optical Character Reader) が代表的であるが，OCR はいったん書かれてしまった，あるいは印刷された文字を認識の対象とするため，計算機の使用形態でいえば集中高速のバッチ処理に適し，情報問い合わせのようなオンライン実時間的な使用には必ずしも適していない．これに対しオンライン的な使用形態を指向した，「オンライン (実時間) 文字認識」と呼ばれるものもいくつか研究[1)~6),8)]されている．オンライン文字認識のアルゴリズムは対象となる文字の形態（曲線を主構成要素とするか直線を主構成要素とするかなど）で決まる場合が多いが，おおよそ次の二つの方法が考えられる．

(1) 文字をストロークに分解し，ストローク固有の属性，ストローク間の相対位置関係などを考慮して認識を行なう．

(2) 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，一次元的な波形とみなして認識する．

(2)の方法では入力波形と標準波形との一次元的なマッチングをとることも考えられるが，入力波形をフーリエ展開したときの係数をみて認識を行なう方法が藤崎[8)]等によって発表されている

(1)に属するものでは次の二つが代表的である．

(a) ストロークを方向の量子化されたセグメントのつながりで近似し，このチェインリンクを用いてあらかじめ各文字の特定セグメントに関して用意された特長群の有無を調べて文字を決定する．

これは花木[2)]や Bernstein[4)] によってとられた手法である．

(b) 文字中でのストロークの大きさ，連続するストローク間の相対位置関係，ストローク固有の属性（曲率や回転方向など）を参照して文字を判定する．

これは Nugent[5)] がアルファベット小文字について

* An On-Line Recognition of Japanese KATAKANA Characters, by Hidekazu Terai, Kazuo Nakata (Central Research Laboratory of Hitachi Ltd.)

** (株)日立製作所中央研究所

とった手法である.

われわれがとった方法は本質的には(1)の(b)に属するものである. すなわち, 文字軌跡を時間の関数として X, Y 直交座標成分に分解し, これらの波形からそのストロークが文字の構成要素としてあらかじめ決めておいた数種の基本ストロークのいずれであるかを決定したのち, 各文字ごとに用意された基本ストロークの Decision Tree をたどって文字の決定を行なっている.

オンライン方式は文字を手書きしている過程から情報を直接取り込んで認識を行なう, いわば動的な認識方法であり, これには次のような特長が考えられる.

(1) 文字のストローク(画)数は認識の際に非常に有力な情報となるが, ペン先が紙面に圧着されているか否かを観測することによって, 文字を構成しているストロークの分離抽出が簡単にできる.

(2) 文字を書く際の筆点の運動を時間の関数として取り込むから, ストロークの特長や筆順情報を容易に抽出することができ, 認識のアルゴリズムが簡単化される. ストロークの特長としては,
(イ) 直線/曲線の区別,
(ロ) 直線の方向,
(ハ) 開曲線/閉曲線の区別,
(ニ) 曲線の回転方向などがある.

(3) オンライン方式では人間と処理装置が実時間で結ばれているため会話形で認識を進めていくことが可能となる. たとえば機械が認識した結果をただちに返送表示することによって誤認識に対する処理を迅速かつ適確に行ない, 認識の信頼度を向上させることができる.

このようにオンライン文字認識は OCR にないいろいろな特色をもっており, これらをうまく活用することによって種々の興味ある利用形態を考えることができる. 1例としてデータ通信システムの端末に組み込んで認識処理を中央の計算機で集中的に行ない, 端末は文字の入力と認識結果の表示のみを受けもつシステムを考えてみる. このとき当然中央の計算機を多くの手書き入力端末が多重使用することになるが, その際, 電話線1回線でどの程度多重化ができるかを明らかにしておくことが第1に必要である. 以上の観点から, ここではまず手書き文字の筆点情報を伝送する際に必要な伝送周波数帯域幅について考察を加え, ついで手書き片仮名文字の認識アルゴリズムと認識実験の結果について述べる. 筆点情報の伝送に必要な周波数帯域幅としては, 1端末で 40 Hz 以下でよいことがわかり, 手書き片仮名の認識については, なんら制限を加えないで平均 95% 以上, わずかの注意を与えればほとんど 100% 認識でき, 認識に必要な処理時間も HITAC-10 のようなミニコンで, 1端末当り約 50 msec 以内であることがわかった.

2. 手書き文字の筆点運動の観測と伝送帯域幅の推定

われわれが用いている, 手書き情報の入力装置は, RAND Tablet[7] で, 小型計算機 HITAC-10 (コア 16K語) に接続されている. Tablet の機構上, 計算機に取り込まれるデータは定時間間隔サンプリングのデータではなく, 座標データ列は実際の手書き過程に対応した時間情報を正しくになっていない. 以後のデータ解析には座標データを時間の関数として利用することを考えたので, HITAC-10 に内蔵されている 1 kHz のタイマを座標データ取り込みのときにいっしょに読み X_i, Y_i, T_i の形で1点の座標データとした. なおデータ取り込み口で平滑化を行なう. すなわちある時点 t_0 で読まれたデータを $X(t_0)$, $Y(t_0)$, $T(t_0)$, その直前の有効データを X_i, Y_i, T_i とする. このとき $2 \leqq |X(t_0)-X_i| \leqq 4$ あるいは $2 \leqq |Y(t_0)-Y_i| \leqq 4$ の少なくとも一方が成立したとき $i+1$ 番目の有効データを,

$$X_{i+1}=\frac{X(t_0)+X_i}{2},\quad Y_{i+1}=\frac{Y(t_0)+Y_i}{2},$$

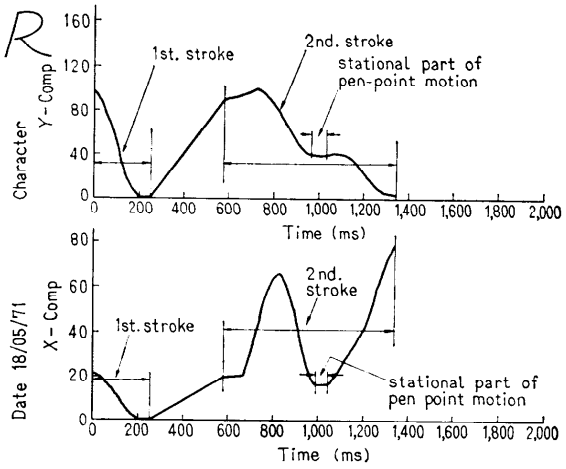


図 2.1 An example of X, Y component of strokes of character "R"

$$T_{i+1}=\frac{T(t_0)+T_i}{2}$$

とする.

図 2.1 はこうして得た座標データを 1 kHz サンプリングの波形 $X(t)$, $Y(t)$ に分解して筆点の動きを時間の関数としてプロットしたものである（ストロークの間は直線で連結してある）. 図 2.1 より筆点の速度は書き始めはゆっくりで, しだいに加速され, ストロークの途中ではほぼ定速運動をし, 最後には減速しながら書き終わることがわかる. また筆点の運動方向の接線ベクトルが不連続に変化する点（たとえば図 2.2 の○印, 以後 "尖点" と呼ぶ）ではかなり大きな筆点

図 2.2 examples of "corner"
"corner"=a stational part of pen-point motion

の時間的停留が観測される. この停留は方向変化が鋭いほど長くなる傾向がある.

図 2.1 に示した "R" の尖点付近の X, Y, T, ΔT を表 2.1 に示す.

次に筆点運動の波形 $X(t)$, $Y(t)$ を伝送すべき信号波形とみて FFT (高速フーリエ変換) によるスペクトル分析を行なったときのスペクトルを図 2.3 に示す. これからわかるようにスペクトルの主要成分は 10 Hz 以下に集中している. 一般に尖点は走査による波形では高周波成分を発生するところであるが, 上述のように筆点はこの近くで時間的に非常にゆっくりした運動をするから, その時間波形はなだらかであり高い周波数成分を発生しない. これは情報伝送の観点からみれば非常に大きな利点である. これを確認するために次

表 2.1 (X, Y, T, ΔT) of the pen-point motion near the corner of character "R"

N	X	Y	T(msec)	ΔT(msec)	
⋮	⋮	⋮	⋮	⋮	
215	17	38	979	4	
216	16	38	987	8	
217	16	39	1047	60	←"corner" of "R"
218	17	39	1051	4	
219	17	40	1057	6	
⋮	⋮	⋮	⋮	⋮	

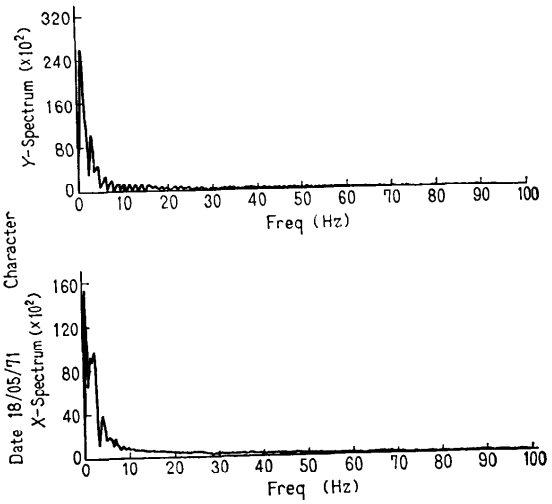


図 2.3 Frequency spectrum of X, Y component of strokes of "R"

のような実験をした. まず, 取り込んだ座標データを 1 kHz のサンプリングデータになおし, これを DA 変換して X, Y アナグロ信号を得, 低域ろ波器に通して X-Y CRT 上に表示する. その際, ろ波器の遮断周波数を変えて文字の受けるひずみを調べた. 図 2.4 はその 1 例である. これにより 8～12 Hz まで遮断周波数を下げても再生文字パタンにはたいしたひずみを生じないことがわかる. この結果から普通の大きさの文字を普通の速さで書くときの手書き文字伝送に必要な情報伝送速度を求めることができる.

条件:

（1） ディジタル伝送
（2） サンプリング周波数—20 Hz
（3） 1 文字の大きさ—1 inch×1 inch
（4） メッシュ分解能—0.01 inch
（5） ペン up/down—1 ビット

ゆえに, 1 文字は 100×100 メッシュで表現されるこ

(a) original pattern　(b) cut off freq. 12 Hz　(c) cut off freq. 8.5 Hz

図 2.4 Examples of regenerated pattern "R" by high frequency cut off

となり，1点の座標は X, Y それぞれ7ビットで指定されることになる．これにペンの up/down を示す1ビットを加えて，1点の情報をそれぞれ8ビットで構成する．1秒間に 20 サンプルを伝送する必要があるから結局，1端末当りの必要情報伝送速度は，

$$8 \text{ビット} \times 2 \times 20 = 320 \text{ ビット/秒}$$

となる．

同じ文字を単純な水平スキャンによって 100×100 メッシュの白黒ドットパタン (0, 1) パタン) になおして伝送する場合，1文字当りの伝送時間を4秒（1文字を書くのに要する平均時間と同程度）と仮定して平均，

$$100 \times 100/4 = 2500 \text{ ビット/秒}$$

となる．つまり 2400 ボーのデータ回線を用いればオンライン方式では7端末までの多重伝送が可能となる．なお，分解能をいま 0.01 inch/mesh としたが，実験によって 0.04 inch/mesh まで粗くしてもさしつかえないことがわかった．この場合には他の条件を同じとして各座標は (5+1)=6 ビットで表現できる．しかし X, Y 座標の分解能を下げると筆点情報からストロークの直線性・曲線性，直線のときはその方向，曲線のときはその回転方向などの諸特長を抽出する精度がおちるから，X, Y 座標の分解能を十分にとりながらしかも筆点運動の特性によって情報を圧縮して伝送できるという特長を利用すべきである．さらに座標値をそのまま送らず，各サンプル点の座標値の差だけ送ったり，端末の出口でストロークごとに符号化してしまえばさらに情報圧縮を達成することができる．

3. 手書き片仮名文字のオンライン認識

3.1 基本ストローク

われわれは手書き文字認識の手法として，構造解析的な手法をとる．すなわち文字の構成要素としてのストロークを抽出し，これの数，順序，相対的な位置関係などを調べて識別を行なう．最初に以下に用いる用語について定義をしておく．

定義 1 ストローク

文字を手書きする過程において筆点が紙面に圧着されて文字を書き始めてから，次に筆点が紙面を離れるまでに描かれる線分

定義 2 基本ストローク

文字を構成するうえに必要なストローク

定義 3 尖点

文字を手書きする過程において筆点の接線ベクトルが不連続に変化するストローク上の特定の点

定義 4 単純ストローク

基本ストロークでそのなかに尖点をもたないストローク

定義 5 複合ストローク

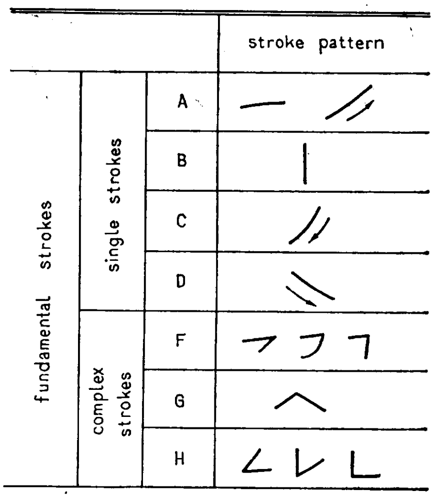
基本ストロークでそのなかに尖点をもつストローク

定義 6 セグメント

尖点において区切られる1本の線分．単純ストロークは一つのセグメントからできている

片仮名文字について採用した基本ストロークを表 3.1 に示す．認識の第1段階は入力データからストロ

表 3.1 Fundamental strokes of KATAKANA characters



ークを分離抽出し，それが基本ストロークのいずれであるかを決定することになる．基本ストロークへの分離のしかたには種々な方法が考えられるが，われわれはストロークの gross features をとらえることによってできるだけ簡単にこれを行なうようにした．

基本ストロークのうち，単純ストロークは書き始めと書き終わりの座標がわかれば容易に表 3.1 の A～D のいずれであるかを知ることができる．また複合ストロークは尖点において2本のセグメントに分割できれば同様にして分類することができる．そこで第1にやるべきことは次の二つである．

(1) 入力文字から抽出されたストロークが単純ストロークであるか複合ストロークであるかを調べること．

(2) 単純ストロークあるいは複合ストロークのセグメントの方向を調べること．

3.2 ストロークの抽出と分類

RAND Tablet では筆点と紙面との圧着状態を筆圧情報として up/down の1ビットで抽出できる．そこで文字を入力している途中で up/down の情報をみることによりストローク数を正しく知り，ストローク区間を抽出することができる．分離抽出されたストロークが基本ストロークのいずれであるかの決定は次の方略に従って行なう．

（1） 単純・複合ストロークの判別

単純ストロークか複合ストロークかの判別は以下の3項目を段階的に調べて行なう．

（a） ストロークの長さによる判別

ストローク S_i の長さ（サンプル点の数）を L_i，文字全体のストローク長(1文字のサンプル点の総数)を L_e とする．このとき，

$$L_i \leqq \alpha \times L_e \quad (\alpha=0.25) \qquad (1)$$

ならば単純ストロークであるとする．$L_i > \alpha \times L_e$ のときは結論を出さず，(b)の判定に持ち込む．

（b） ストロークの X および Y 成分の変化波形

単純ストロークの X 波形，Y 波形は図 3.1 のように単調に変化し，途中に極大点あるいは極小点をもたない．一方，複合ストロークでは図 3.2 のように2本のセグメントの結合点（尖点）において X あるいは Y が極値をとる．それゆえストローク座標列中に極点が存在するか否かを調べて判別を行なう．なおストロークの書き始めおよび書き終わりの“はね”による影響をなくするために，書き始めと書き終わりのデータを $\beta \times L_i$ $(\beta=0.15)$ だけ除いた残りについて上の判定を行なう．

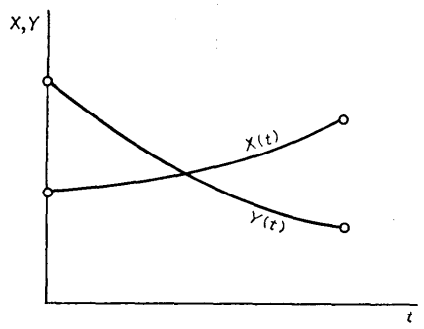


図 3.1 $X(t)$, $Y(t)$ of sigle stroke; stroke “レ”

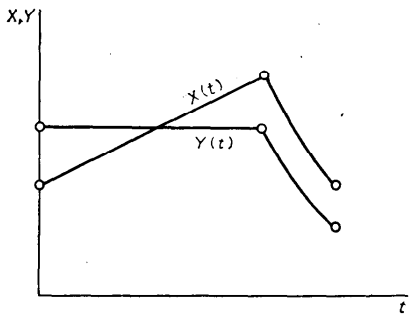


図 3.2 $X(t)$, $Y(t)$ of complex stroke; stroke “ア”

（c） 時間軸上における停留点の有無

図 3.3 に示すような“なまった複合ストローク”では（b）の方法では正しい結果を得ることができない．すでに述べたように時間軸上の筆点の停留はストローク上の尖点で生じるから，時間座標を調べることにより正しい判別を行なうことができる．停留を与える閾値を ΔT_θ，各サンプル点の時間差を ΔT_i，その最大値を ΔT_{max} とすると

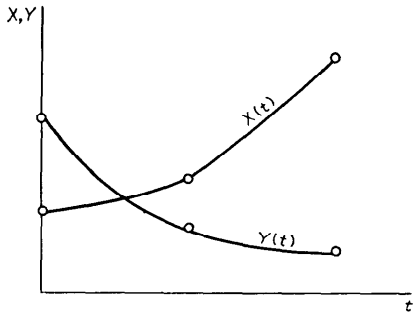


図 3.3 $X(t)$, $Y(t)$ of distorted complex stroke; stroke “レ”

$$\Delta T_{max} \geqq \Delta T_\theta \qquad (2)$$

なら複合ストロークとする．ΔT_θ は式（3）より求める．

$$\Delta T_\theta = \frac{T_E - T_S - \Delta T_{max}}{\text{ストロークサンプル点数}} \times \beta \qquad (3)$$

（T_S, T_E はストロークの始点，終点の時間座標）

β は 5～8 が経験的にみて適当である．

（2） 変形ストロークの処理

正しく書かれたすべての複合ストロークは（a），（b），（c）の方法によって判別可能であるが，ときにはこれだけでなお正しい結果を得ることができない場合がある．たとえば図 3.4 の文字を“ヤ”とみるべきか“セ”とみるべきかはストローク“レ”を D ストローク，H ストロークのどちらに判定するかによっ

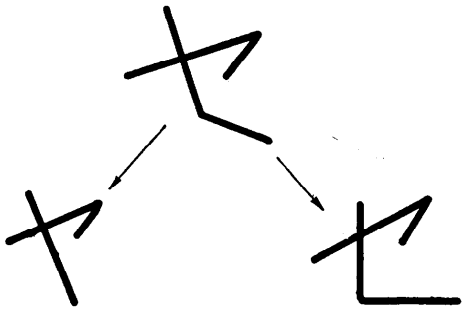


図 3.4 Examples of ambiguous characters

て決まる．このようなあいまいなストロークは F および H ストロークにみられ，いずれも式（2）が成立しないものである．このようなストロークは一応単純ストローク（D）に入れておき，次のストローク決定段階で D ストロークについてのみ曲率に相当した量，式（4）の q，r を計算し最終判定に供する．すなわち図 3.5 で，

$$q=|\overline{AM}|-|\overline{SM}|/\delta \quad (\delta=3)$$
$$r=X_M-X_A \qquad (4)$$

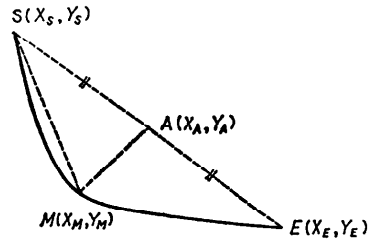


図 3.5 Quantative estimation of "curvature" of a cursive stroke

図 3.6 An example of misjudged stroke

また図 3.6 のようなストロークは（b）の方法で複合ストローク（G）となる．しかしこれは本来単純ストロークであるべきものが変形したと考えるのが妥当である．ここでは一応複合ストロークと判別されるが，ストロークコードの決定段階でその妥当性が調べられ，もし複合ストローク論理式（5）に合わなければ改めて単純ストロークとしての決定を行なう．

（3） ストロークコードの決定

単純ストロークセグメントおよび複合ストロークセグメントの傾斜許容範囲を図 3.7，3.8 に示す．単純

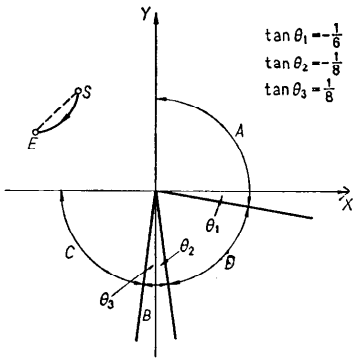


図 3.7 Allowable domain for each single stroke

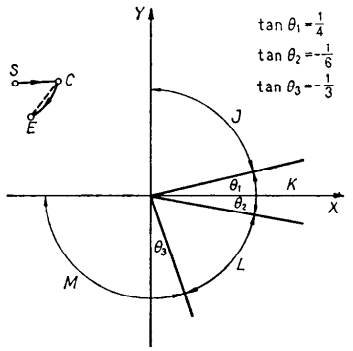


図 3.8 Allowable domain for each complex stroke

ストロークと判定されたものについては始点 S と終点 E を結ぶベクトル $\vec{SE}$ が図 3.7 のどの領域にはいるかを調べる．複合ストロークと判定されたものでは二つのセグメントの結合点 C が求まっているから $\vec{SC}$，$\vec{CE}$ についておのおのが図 3.8 のどの領域にはいるかを調べ式（5）によって複合ストロークを決定する．

$$F=((K \text{ or } J)\blacktriangleright M) \text{ or } (D \text{ and } q\geqq 0 \text{ and } r>0)$$
$$G=J\blacktriangleright L \qquad (5)$$
$$H=(M\blacktriangleright(K \text{ or } J)) \text{ or } (D \text{ and } q\geqq 0 \text{ and } r<0)$$

{▶は左辺（第1セグメント）の後に右辺（第2セグメント）が続くことを表わす}

以上述べたストロークコード決定の総合的なフローチャートを図 3.9 に示す．

（4） 濁点・半濁点の処理

片仮名文字には濁点（゛）および半濁点（゜）がつく場合がある．このときこれらの濁点あるいは半濁点を分離検出し残りについて文字としての認識を行なえば

よい．濁点は人によってさまざまな書き方があり，これを安定に抽出するには（可能ではあるが）複雑な処理が必要である．一方，半濁点はループを形成しこれは安定して抽出することができる．また濁点のない文字のうちで最多画数のものは「ホ，ネ」の4画2字である．以上のことを勘案し，図 3.10 に示す濁点半濁点処理ルールを作った．ここで問題となるのはループの検出方法であるがそれは次のとおりである．

（5） ループの検出

まず図 3.11 に示すようにストロークを4等分し，各分点の座標を $P_i(X_i, Y_i)$ $(i=1, 2, 3, 4, 5)$ とする．次に $X_{i+1}-X_i$，$Y_{i+1}-Y_i$ $(i=1, 2, 3, 4)$ を計算し，

$$\begin{aligned} &X_{i+1}-X_i>0 \text{ なら } dX_i=1 \\ &X_{i+1}-X_i\leqq 0 \text{ なら } dX_i=0 \\ &Y_{i+1}-Y_i>0 \text{ なら } dY_i=1 \\ &Y_{i+1}-Y_i\leqq 0 \text{ なら } dY_i=0 \end{aligned} \qquad (6)$$

を求めて2ビットの2進符号．

$$D_i=dX_i dY_i \quad (i=1, 2, 3, 4) \qquad (7)$$

を作る．そしてすぐ次の符号とのハミング距離，

$$H_i=D_i\tilde{H}D_{i+1} \quad (i=1, 2, 3) \qquad (8)$$

（$\tilde{H}$ はハミング距離を計算する演算記号とする）

の和，

$$H_S=\sum_{i=1}^{3} H_i \qquad (9)$$

を求める．こうして求めた H_S を用い式 (10) によりループの存否を決定する．

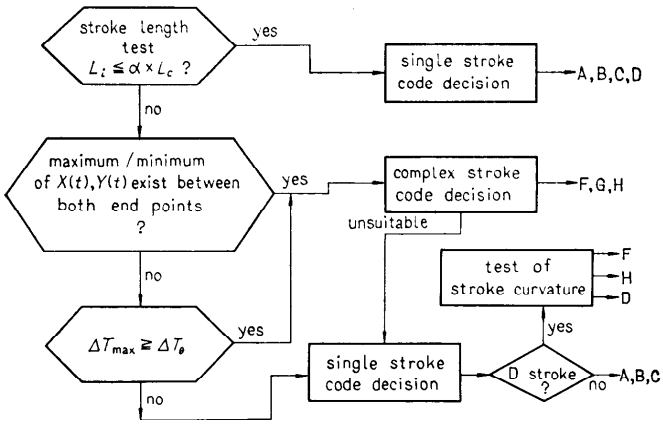


図 3.9 Flow chart of stroke classification

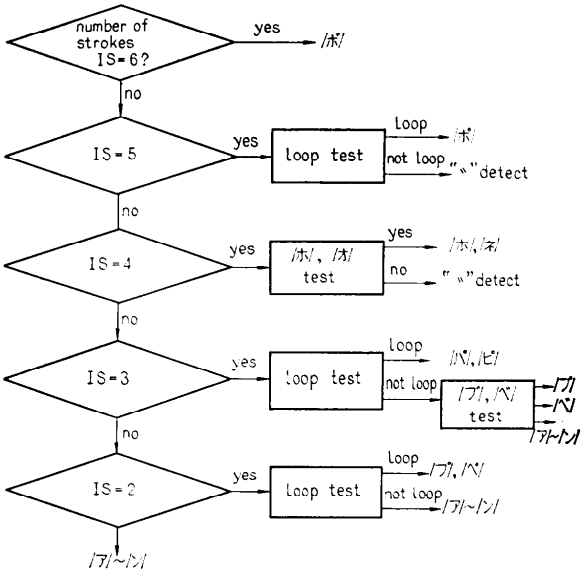
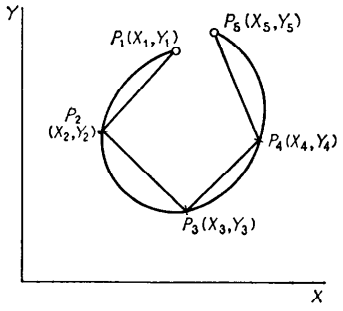


図 3.10 Flow chart to detect "゛" and "゜"



	Binary Code	Hamming Distance
D_1	0 0	1
D_2	1 0	1
D_3	1 1	1
D_4	0 1	H S 3

図 3.11 A method of loop detection

$H_S \geqq 3$ のとき；ループ存在

$H_S = 2$ のとき；

$$|\overline{P_1P_5}| = \sqrt{(X_1-X_5)^2+(Y_1-Y_5)^2} \leqq 4$$

ならループ存在 (10)

それ以外のとき；ループ存在せず

4. 文字識別ルール

片仮名文字（濁点・半濁点を除く）をストローク数によって分類すると表 4.1 のようになる．文字のストローク数は，すでに述べたように入力段階で容易に抽出することができるから，これを用いて最初のクラスタリングを行なえば非常に有効である．いまの場合クラスタの数は4であり，そのうち最大のものは2画文字からなるもので 27 字ある．

表 4.1 Classification of KATAKANA characters by the number of strokes

number of strokes	KATAKANA characters									
1	ノ	フ	ヘ	レ						
2	ア	イ	カ	ク	コ	ス	セ	ソ	ト	ナ
	ニ	ヌ	ハ	ヒ	マ	ム	メ	ヤ	ユ	ラ
	リ	ル	ワ	ン	(ケ)*)	(チ)*)	(ヲ)*)			
3	ウ	エ	オ	キ	ケ	サ	シ	タ	チ	ツ
	テ	ミ	モ	ヨ	ロ	ヲ				
4	ホ	ネ								

*) (ケ), (チ), (ヲ) indicate to be written with 2 strokes

文字識別に際して用いた情報は，（1）ストローク数，（2）ストロークコードの順列（すなわち筆順），（3）ストローク間の位置関係，である．大部分の片仮名文字は筆順が一定しているが，2通り以上の筆順がみられるものもいくつか認めた．ほとんどの文字はストローク数と筆順だけで正しく識別することができるが，ストローク数およびストロークコード列が同じで，ストローク間の位置関係までをみないと識別できないものもある．（アとカ），（マとヤとスとヌ），（オとサ），（コとユ）がそれらである．たとえば（アとカ）の場合，第1ストロークの第1セグメントと第2ストロークの交差関係を調べる．いまの場合は図 4.1 に示すように線分 $\overline{AB}$ と $\overline{AR}$ の勾配を比較することでわかる．

文字判定の辞書はそれぞれの文字を認識するための

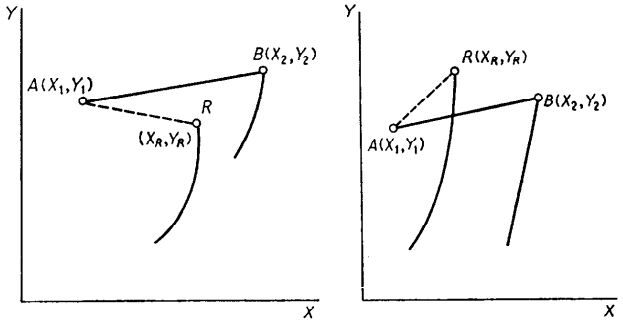


図 4.1 Check of the relative position of the first stroke to the second one in the discrimination of "ア" from "カ"

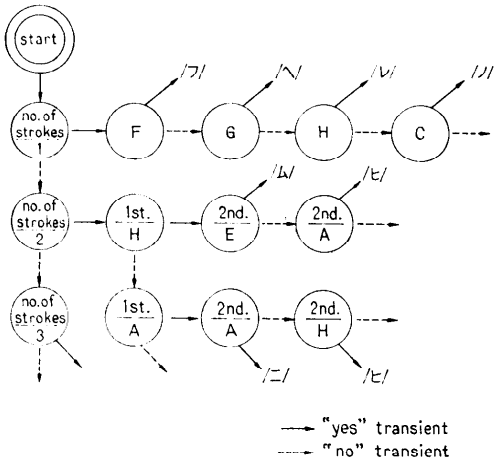


図 4.2 Examples of Decision Tree for KATAKANA characters

Decision Tree である．この Tree は最初，人間が各文字を書く際最も普通であると筆者らが判断したルールをそのまま取り入れて作り，逐次，現実の変形に対する許容度をあげるよう修正拡張していった．Decision Tree の例を図 4.2 に示す．

5. 認識実験と結果の検討

認識プログラムは HITAC-10 のアセンブラ語で約 2500 ステップ，1文字の認識に要する時間は 1 cm×1 cm 程度の文字で 50 msec 弱である（データ取り込みの段階で同時に特長抽出を行なえば時間は文字の大きさに無関係に 15 msec/文字，となるが，ペン先の up/doun の際のチャタリングノイズなどを除去するため，いったんデータを取り込んだのち，ノイズ除去をしてから特長抽出をしている。この前処理の時間が文

表 5.1 Analysis of causes of errors

causes of errors	examples	
	input	output
number of strokes (unexpected way of writing)	コ	フ
length of strokes (too short to detect)	ラ タ	フ ク
mis-classification of stroke	イ ハ	ハ ヤ
the dictionary does not contain their stroke sequence (unexpected sequence of writing)	メ	ソ
stroke distortion	リ	ソ
mis-judge for relative position of strokes (critical stroke relation)	ス ヌ コ ユ	ヌ ス ユ コ

字の大きさに依存する).

8人のまったく新しい別の被験者で,文字の大きさ,書き方にまったく予備知識を与えないで試したところサンプル文字総数 576 字で,

正解	547 字	95%
誤り	17 字	3%
リジェクト	12 字	2%

の結果を得た.

誤りの主な内容とその原因を表 5.1 に示す.

これらのうち辞書を拡張すれば改善されるものもあるが, "ス"⇄"ヌ", "コ"⇄"ユ" のように一義的にどちらかとは決められない場合もある. これらは単語, あるいは文章中でのコンテキストを考慮してはじめてどちらであるか判明するのであるが, これをアルゴリズムで行なうにはあまりにも複雑である. このような文字は, 書く際に辞書で決められた閾値を正しく満たすような書き方をするのが現在最も得策である. あらかじめ注意あるいは予備知識を与えることはこの程度であればそれほど大きな制約とはならない.

6. むすび

オンライン手書き文字認識の手はじめに片仮名文字の認識を試み,特別の注意を与えないで平均95%,注意を与えれば100% に近い認識ができた. この手法は文字の位置のずれ, 大きさには関係ないが現在はストロークを決定する際に Tablet 面上に固定された X, Y 直交座標系に従っているため, 文字の傾きに対する許容度は少ない. またこの手法はアルファベットの大文字や漢字の認識にも拡張することができる.

また手書き筆点運動波形の解析から, 文字の手書き過程の情報を伝送するときに必要な情報量は 320 ビット/秒以下であることが確認できた. これを利用して経済的な端末文字入力システムを作ることができる. また HITAC-10 程度の小型計算機でも約 20 端末の同時処理が原理的に可能であることも確認された.

最後に日ごろご指導いただく神原所長, 芳根, 沼倉両部長に謝意を表する.

参考文献

1) 萩原他: 手書き過程による文字の識別について, 信学会オートマトン・インホメーション理論研究会 (1968. 9)
2) 花木他: 遠隔端末からの実時間文字認識, 情報処理 (1970. 10)
3) 寺井他: ON-LINE 手書き文字認識, 信学会オートマトン研究会 (1971. 7)
4) M. I. Bernstein: Computer Recognition of On-Line Hand-written Characters: *RAND* (Oct. 1964).
5) G. F. Groner: Real-time Recognition of Hand printed Text.; *FJCC* (1964).
6) W. R. Nugent: The On-line Recognition of Cursine Writing Using Geometric-Topalogical Invariants of Stroke Succession; 1st Annual IEEE Computer Conference '67.
7) Davis et al.: The RAND Tablet A Man-Machine Graphical Communication Device; *FJCC* (1964).
8) 藤崎他: 手書き数字のオンライン認識の一方法, 昭和 46 年電子通信学会全国大会

(昭和46年9月7日受付, 11月26日再受付)