# オンライン手書き文字認識システム JOLIS－1の定量的評価

中川正樹、池田裕治、相澤　正、滋田千冬、高橋延匡　（東京農工大学工学部）

## １．はじめに

今日の日本語ワードプロセッサやマイクロコンピュータ用のいわゆるワープロ・ソフトの流通を見ると、日本語情報処理は完全に開花した感がある。現実に我々が日常手にする文書の大半はワードプロセッサで作成されたものである。しかしながら、市販のワードプロセッサを実際に使用してみると、入力の問題、図表の処理、ファイル管理の問題など、情報科学、人工知能などからのアプローチを要する課題が山積している。日本語文書がメモリの中に蓄積できて、編集・再利用が可能であるというのは確かに大きなステップであったに違いないが、日本語情報処理を誰でもが自然に享受できるようにするには、今までよりもさらに本質的な問題を解決しなければならないように思われる。

我々は、我々自身の文書作成を支援する目的で、オンライン手書き入力(JOLIS)[4,5,7,8]、パタン認識および日本語情報処理の基礎となるオペレーティングシステム(OS/omicron)[10]、そして日本語文書の清書出力(JOSHO)[9]の研究開発を進めてきた。これは、研究室内で作成される各種ソフトウェアのドキュメンテーションにも役立て、毎年研究室のメンバー(修士学生)の半数が入れ替わる環境での、ソフトウェアの生産性、信頼性、保守性に対する日本語情報処理の効果を見るためでもある。我々はこれまでの経験から、ソフトウェアの設計仕様書の類は設計者自ら入力しなければ、誤りの混入は避けられないと痛感している。

本稿は、これら一連の研究のうち、オンライン手書き入力に関するものであるが、オンライン手書き入力の特徴、研究動向、我々の研究経過については、本研究会における前回の発表[4]に詳しいので、ここでは最小限の記述にとどめる。ただ、我々がなぜオンライン手書き入力方式を研究するのかを繰り返せば、それが日本人には最も自然な入力方式であり、筆記者の思考を妨げないということである。

よって、以下の節ではまず、JOLIS－1に至る過程で得られた知見を要約し、その認識に基づいて作成されたJOLIS－1の概略を提示する。そして、JOLISの認識モデルを構築し、前回の発表後に作成したオンライン手書き文字標本データベースを用いてJOLIS－1およびそのモデルの正当性を評価する。そして最後にJOLIS－1以降、どの方向を指向すべきかを検討する。

## ２．JOLIS－1に至る過程で得られた知見

本節では、JOLIS－1の以前に作成された予備実験システムおよびJOLIS－0で得られた知見を要約する。

(1) 直線的なストローク(画)を８方向に分類したが、非常に多くのストロークが、２つの隣り合う方向にまたがって分布しうる。

(2) 筆順はあまり安定した情報ではない。誤った筆順の文字でも認識されないと困る。

(3) ストロークの角の検出に筆速の低下を利用しようとしたが、筆速情報は全くと言っていいほど信頼できない。

(4) タブレットに対するペンの筆圧が弱いと筆点座標が取り込まれない。これはストロークの始めと終りに多く見られ、結果として最初か最後のセグメントを欠いたストロークとして識別されてしまう。

(5) ストロークの修飾(おさえ、はね)は、正しいストローク識別を極めて困難にする。

(6) 短いストロークの書かれる方向は不安定である。

(7) ゆるい角や小さな角は、前処理(平滑化処理、間引き処理)、セグメント化処理で欠落してしまう場合がある。これは特にストロークの終りに顕著で、これまで用いてきた平滑化処理がストロークの最後へ行く程、効果が大きくなるためである。

(8) JOLIS-0においては、たった１つのストローク誤識別のために、入力文字の認識に失敗する例が多かった。漢字のようなパタンには十分な冗長性があり、それを利用する必要性が痛感された。

(9) 標準的な標本文字データベースを用いた認識実験を行なわなければ、定量的な議論はできない。

## 3．JOLIS－1の概要

### 3.1　処理の流れ

オンライン手書き文字認識システムJOLIS－1は構造解析的手法を用いて、平仮名、片仮名、教育漢字、および幾つかの記号を認識する。JOLIS－1は図1に示すようにまず、手ぶれや量子化雑音をできるだけ除去するために入力ストロークの筆点位置座標系列に前処理を施す。そして、その前処理されたストロークを８方向の直線セグメントの系列に変換する。次に、そのセグメント化されたストロークを、方向変化(転回)とそれを挟むセグメントの列に情報圧縮し、ストローク辞書とのマッチングをとることによって、29種類の基本ストロークのいずれかに識別する。JOLIS－1は、このようにして得られた入力文字のストローク列を、標準字体のストローク列辞書(文字辞書)と相違度マッチングをとることによって入力文字を決定する。ただし、ストローク列情報だけでは識別できない文字の組については、ストロークの位置関係(付加条件)を調査して文字を決定している。

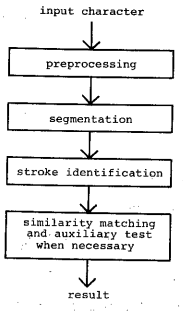



図1 処理の流れ

### 3.2　前処理

セグメント化処理において、入力ストロークの形状を安定に抽出できるように、まず次の順で前処理を行う。

(1) 平滑化処理

$$X_i = \frac{3X_{i-1} + x_i}{4} \qquad Y_i = \frac{3Y_{i-1} + y_i}{4}$$

$(x_i, y_i)$：i番目の入力筆点の原座標，$(X_i, Y_i)$：平滑化後の座標

(2) 間引き処理

$$(X_j, Y_j) = (X_i, Y_i) \quad \text{if} \quad |X_i - X_{j-1}| \geqq \Delta \quad \text{or} \quad |Y_i - Y_{j-1}| \geqq \Delta \qquad \Delta = 0.4\text{mm}$$

$(X_i, Y_i)$：平滑化後の座標，$(X_j, Y_j)$：間引き処理後の座標

さらに、ストロークの始めと終りに顕著な、不安定なペンの動きやストローク修飾を除去するために、ストロークの始終点に正方形のマスク（間引き処理における$\Delta$=0.8mm）をかけて、ストロークの前後を一様に無視している。

### 3.3　セグメント化処理

前処理された座標系列を８方向の直線セグメントの系列で近似する。JOLISでは次のような方法を考察した。まず座標系列の進行方向が図2のPaで示される８方向の境界を横切るとき、そこでストロークを分割する(セグメント化処理フェイズ1)。そして分割点を結ぶベクトル(セグメント)は図2の８方向境界Pbで分類する。ここで、もし連続した２つのセグメントが同一方向になるならば１つにまとめる（セグメント化処理フェイズ２)。

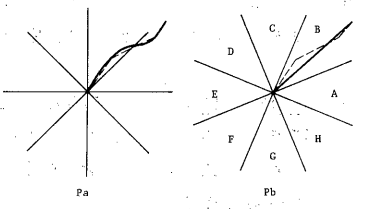



図２　セグメント化処理のための８方向分割境界PaとPb

この方法によると、PaやPbの境界付近をゆらぐストロークでも、不必要にセグメントに分割されることがない。

以上までの処理の例を図３に示す。

入力パタン　　前処理の効果　　セグメント化処理フェイズ１　　セグメント化処理フェイズ２

図３　前処理、セグメント化処理の効果

### 3.4 ストローク識別

セグメント化処理によって得られた８方向セグメントの系列は図４に示すように、転回とそれを挟むセグメントの系列に情報圧縮される。そして転回を含まないものは、その方向によって表１に示す８種類の直線ストロークの何れかに分類される。また、もし転回を含んでいれば、表２に示すストローク辞書で同じ転回の個数をもつ項目とマッチングをとることによってストロークを識別する。もし、どれとも判別できない場合には、本来のストロークの代りという意味でJOKER(JR)ストロークとされる。

表２のストローク表現において、２つの転回の間にくるセグメントの方向は規定されないことが多い。このことは、ストロークをセグメントの系列で表現するのではなく、その方向変化で表現していることを意味する。これは後者の方が前者よりも安定した情報だからである。同一のストロークでも、セグメント列の変化はかなりある。しかし、ストロークの転回系列はそう変りうるものではない。したがって、ストロークの転回情報に着目した本ストローク識別法は、セグメント系列のまま、その変化を許容してマッチングをとろうとするDP-matching法と好対称をなす。また、このストローク表現は別の利点も有している。それは、データ表現であるために追加・変更が容易なこと、および、ストローク識別に要する時間がほとんど無視できる程小さいことである。

JOLISの処理においては、ストロークの幾何学的変形に対処するのは以上の段階までであり、以後はストロークのシンボル列を扱うだけである。

### 3.5 文字認識

入力文字を構成するストローク列が決定されると、それは文字辞書内の同じストローク数を持つ標準字体のストローク列とマッチングされる。ここで、文字辞書内の標準字体の記述には表１の基本ストロークばかりでなく、図５に示すオーバーラッピング・ストロークが用いられている。

オーバーラッピング・ストローク Ｓ、Ｔ、Ｕ、…、Ｚはそれぞれ ＣとＤ、ＤとＥ、ＥとＦ、…、ＢとＣとマッチする。これらのストロークは文字辞書のサイズを小さくするために予備実験システムから導入されている。第２節で述べたように、多くの文字で８方向の分割境界の隣り合う２方向にまたがってストロークが分布しうる。たとえば、“木”の場合、‘↙’のストロークと‘↘’のストロークはそれぞれ、ＦまたはＧ、ＧまたはＨのいずれでもかまわない。したがって、もしオーバーラッピング・ストロークを用いなければ“木”のために４つの表現“木”(ＡＧＧＧ)、“木”(ＡＧＦＨ)、“木”(ＡＧＧＨ)、“木”(ＡＧＦＧ)を文字辞書に登録しなければならなくなる。オーバーラッピング・ストロークを用いれば、この場合、１つの表現ＡＧＶＷを登録すれば済み、文字辞書全体としては、その大きさを１/３０にすることができた[４]。

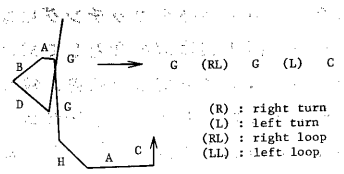



図4 “↓↻”の情報圧縮表現

表１ JOLIS-1の基本ストローク

| Straight Strokes | | Cursive Strokes | | | |
|---|---|---|---|---|---|
| A | → | I | | HD | |
| B | ↗ | J | | HE | |
| C | ↑ | K | | HF | |
| D | ↖ | L | | HG | |
| E | ← | M | | HH | |
| F | ↙ | O | | HI | |
| G | ↓ | P | | HJ | |
| H | ↘ | Q | | HK | |
| | | HA | | HL | |
| | | HB | | HM | |
| | | HC | | | |

表２ ストローク辞書

| Number of Turns | Representation of Stroke | Stroke Code |
|---|---|---|
| 1 | H-A (R) D-G | I |
| | A-B (R) H-A | J |
| | F-G (RL) F-G | HA |
| | H-A (RL) A | HB |
| | D-E (RL) F-G | HG |
| | A-C (LL) G-A | HG |
| | C (R) A | HH |
| | A (L) C | HI |
| | F-G (RL) H-A | HJ |
| | ? (L) ? | K |
| 2 | A (R) ? (L) A-C | L |
| | F-G (L) ? (R) D-G | O |
| | G (RL) ? (L) A-C | HD |
| | A (R) F (L) G | HK |
| | G-H (R) D-E (L) H-A | HL |
| | G (L) B (RL) F-G | HM |
| 3 | A (R) ? (L) ? (R) D-G | M |
| | F (L) ? (R) ? (L) H-A | Q |
| | A (R) ? (L) ? (RL) G-A | HC |
| | F-G (R) ? (L) ? (R) H-A | HE |
| | F-G (L) ? (R) ? (L) A-B | HF |
| 4 or more | A (R) ? (L) ? (R) ? (L) A * | P |

$s_1$-$s_2$: matches any segment from $s_1$ to $s_2$ counterclockwize.
? : matches any segment.
* : matches any sequence of segments and turns.

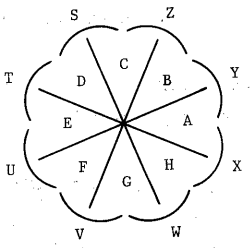



図５ オーバーラッピング・ストローク

入力文字と標準字体のマッチングには相違度を用いる。入力文字と標準字体との間で、その相違度を対応するストローク間の相違度の和として求め、その最小のものを選ぶことによって入力文字を決定している。これは、手書きによる変形および修飾、筆点座標のサンプリング・エラー、前処理ミス等に起因するストローク誤識別に対処するためであり、構造解析的手法においても、パタンの変動に対処するためには相違度の概念が必要であるという認識に基づいている。漢字のようなパタンには十分な冗長性があり、たとえ幾つかのストロークに誤識別が起きても、文字としての認識には差し支えない場合が多い。

入力文字と標準字体との相違度は対応するストローク間の相違度の和として定義した。そして、ストローク間の相違度は、本来ストロークｔであるべきものが、入力ストロークｓとして識別あるいは誤識別される頻度が大きい組について、小さな値をとるよう経験的直感によって設定した。池田らの方式においては、もっと緻密にストローク間距離が設定されているが、その設定方式においてJOLIS－1のそれと本質的に変わるところはない[２]。

ところが、この設定方式には効果とともに問題があることが明らかになったので、現在では次節に述べるように、より厳密な統計的理論に基づく相違度の設定を行っている。

以上のようにして、入力ストローク列に対して、それに最も近い標準ストローク列が見出される。ただし、ここで問題があるのは、ストローク列情報だけでは文字を一意に決定できない組が存在することである。JOLIS－1では、こうした文字の組に対して、付加条件を設定し、ストロークの始終点の位置関係等を用いて入力文字を決定している。たとえば、“土”と“土”と“工”を区別する付加条件は図６のようになる。入力文字は、まず関係部の各条件について調べられる。その結果と文字を決める付加条件がマッチすれば、その文字に決定する方式である。

| | 文　字 | | |
|---|---|---|---|
| 付加条件 | 土 | 土 | 工 |
| B(1)<yB(2) | Y | Y | N |
| B(1)<xB(3) | Y | N | N |

Ｙ：満足する，Ｎ：満足しない
Ｂ(ｎ)：第ｎストロークの始点
＜x,y：Ｘ(Ｙ)座標の大小判定

図６“土”、“土”、“工”を区別する付加条件

## ４．JOLIS－1の認識モデル

JOLISでは、まず文字の構成要素(pattern primitive)を識別し、その結合体(構造体)として文字を認識するという構造解析的な認識モデルが前提となっている。JOLIS－1の場合は、構成要素が29種類の基本ストロークであり、その結合関係は筆順である。構造解析的手法では、この構成要素を決めることがまず問題となり、判断を導くべき理論もない[1]。しかし、オンライン手書き文字認識の場合は、ペンのup/down情報によってストロークの分離･抽出が容易であり、また、文字を書くときの基本単位がストロークであるので、それを構成要素とした。我々がこのような認識モデルを採用したのは、漢字の構造を認識に反映し、略字やくせ字を統一的に処理したいと考えたためである[4,5]。このことは統計的手法には望めない。

しかし、構造解析的方式を、計算機言語処理に見られる構文解析やオートマトンを用いて実現することには重大な欠陥がある。それは、構文解析あるいはオートマトンにおける状態遷移を用いると、パタン中のささいなエラーによってパタン全体をリジェクトしてしまうからである。オートマトンを用いて実用化された数少ない例に郵便番号読取り機があるが、そこではエラーまで考慮した状態遷移を付加することによってこの問題に対処した。しかし、この対処法は10個の数字を識別することには使えても、数千の文字を分類することには極めて不向きである。

構文解析あるいはオートマトンにはまたもう1つの欠陥がある。というのは、この方法では認識結果として“accept”か“reject”しかあり得ず、どれくらい“acceptible”なのかという距離の概念を導入することが難しいからである。

以上の考察から、JOLIS－1では構文解析やオートマトンは用いずに、入力ストローク列を直接標準字体のストローク列とマッチングすることによって入力文字を決定する。これには、入力パタンが高々20個程度のストローク列であるために、その認識に構文解析やオートマトンを必要としないということも幸いしている。つまり、JOLIS－1では入力ストローク列と標準ストローク列について、対応するストローク単位に相違度を求め、その和の最も小さい標準字体が示す文字を認識結果とする。この方式の妥当性は以下で議論するが、こうすることによって構造解析的認識に統計的距離の概念を導入することができる。

入力文字と標準字体の相違度には、本来ならば構造の違い(たとえば、筆順違いやつづけ字)も考慮すべきであろう。そうすれば、筆順違いやつづけ字のパタンを標準字体に登録せずに済み、文字辞書の一層のコンパクト化が実現できる。しかし、JOLIS−1では、まず構成要素間の1対1の相違度のみ考え、他の構造的変形は辞書登録で対処することにした。構造的に変形したパタンに何らかのハンディを付ければ(たとえば、そのパタンの出現確率を小さく設定する)、正しい構造のパタンとの間に相違度を設定することになり、辞書サイズとしては不利でも、モデルとしては等価であるはずである。

このことからも、JOLIS−1では必然的に複数の字体に対してそれぞれを標準パタンとして登録することになる。パタン認識においては、1カテゴリあたり1標準パタンというのは確かに理想ではあるが、特徴空間において、すべてのカテゴリが標準パタンを中心に正規分布をするとは考え難く、むしろ、カテゴリの歪な分布を幾つかの正規分布の重ね合せで近似すると考えた方が現実的である。

以下の節ではこの考えに沿って、もう少し厳密にモデルの構築を行う。

## 4.1 ストローク間相違度の設定

相違度の導入により、文字辞書内の標準パタンを増やさずに、“それらしい”文字の認識が可能になった。しかし一方、その設定値の妥当性を再考させられる幾つかの例が認められた[6]。適当な相違度の設定のためには理論が必要である。

入力ストローク列 $S = s_1 s_2 \cdots s_n$ (JRを含む可能性あり) が文字辞書内の標準ストローク列 $T = t_1 t_2 \cdots t_n$のつもりで書かれる確率$P(T \mid S)$ を考えると、この値を最大にするTが表現する文字を認識結果とすべきであろう。

まず、ベイズの定理により、 $P(T \mid S) = P(S \mid T)\, P(T) \,/\, P(S)$

両辺の $-\log$ をとって、

$$-\log P(T \mid S) = -\log P(S \mid T) - \log P(T) + \log P(S)$$

確率$P(S \mid T)$ について、i番目のストローク$t_i$を書こうとしたとき、それが$s_i$となる確率は、$t_1, \ldots, t_{i-1}, s_1, \ldots, s_{i-1}$ に依存しないとすれば、

$$P(S \mid T) = \prod_{i=1}^{n} P(s_i \mid t_i)$$

したがって、

$$-\log P(T \mid S) = \sum_{i=1}^{n} -\log P(s_i \mid t_i) - \log P(T) + \log P(S)$$

ここで、$-\log P(T \mid S)$ を文字間相違度 $Dc(S, T)$, $-\log P(s_i \mid t_i)$をストローク間相違度 $Ds(s_i, t_i)$ とすると、それらは相違度に対する我々の要求を満たしている。つまり、

(1) 入力ストローク列 S との文字間相違度を最小にする T は、$P(T \mid S)$を最大にする。

(2) 文字間相違度 $Dc(S, T)$ の大小には、ストローク間相違度 $Ds(s_i, t_i)$ のiについての和が反映される。

(3) 本来、$t_i$であるべきストロークが$s_i$として誤認識される確率が大きい程、$Ds(s_i, t_i)$ は小さくなる。

(4) 入力文字に対する標準文字Tの文字間相違度は、Tの出現頻度が大きい程、つまり、$P(T)$ が大きい程、小さい値をとる。したがって、出現頻度の大きい文字を認識しようとする。確率$P(T)$を表現Tの出現頻度バイアスと呼ぶ。( ただし、今回の実験では、$P(T)$ は一定としている。)

(5) 確率 $P(s_i \mid t_i)$ はデータベースを基に決定できる。

## 4.2 確率$P(s \mid t)$ の算出におけるオーバーラッピング・ストロークの取扱い

確率$P(s \mid t)$ は、書こうとしたストロークtが、ストロークsに識別される確率であるが、書こうとしたストロークとは、標準字体におけるストロークということになろう。ここで問題になるのは、tがオーバーラッピング・ストロークの場合である。オーバーラッピング・ストロークは2つの隣接するストロークを1つにまとめたものと考えることもできるし、全く1つのストロークで、その近隣のストロークと小さい相違度を持つストロークと考えることもできる。前者の考え方によれば、tがsに識別された事象は、tがカバーするストロークt', t''のいずれかがsに識別されたと計数すべきであるし、後者だと、tを1つの独立したストロークであるとし、tがsに識別された事象として計数すべきである。

JOLIS－1は現在、Ds(s,t)としてmin{Ds(s,t′), Ds(s,t′′)}を用いているが、これは前者の解釈に合致しているので、P(s,t)の計算にも前者の考え方をとるものとする。したがって、たとえば図7のようにWストロークがAストロークと識別された場合、GストロークかHストロークがAストロークに識別されたとする。そして、この場合、より可能性の高い、HストロークがAストロークに識別された事象とする。前者の考えに従うとしても、WがAになる事象を、P(A|H)とP(A|G)の比率で分配することも考えられるが、そうするのであれば、Ds(s,t)を以下のようにすべきであろう。

tはt′かt′′のいずれかを意図したものだとすれば、

$$P(t'|t)+P(t''|t)=1$$

$$P(s|t)=P(s|t')\,P(t'|t)+P(s|t'')\,P(t''|t)$$

もし $P(t'|t)=P(t''|t)$ ならば

$$P(s|t)=\{P(s|t')+P(s|t'')\}/2$$

$$\therefore\quad D_s(s,t)=-\log(P(s|t')+P(s|t''))+\log 2$$

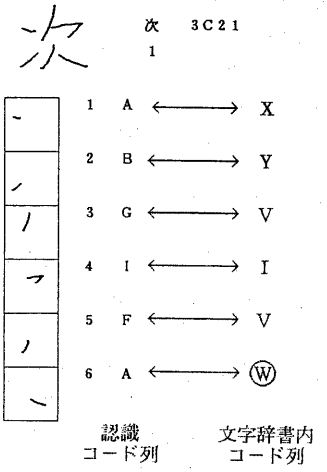
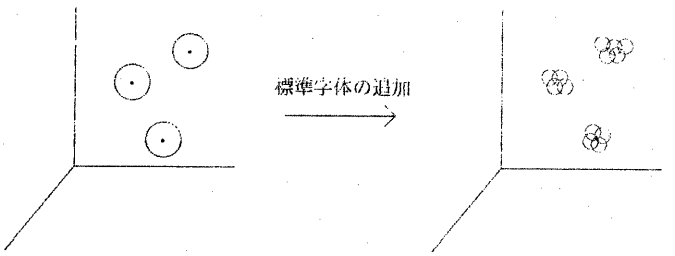



図7　オーバーラッピング・ストロークとの対応

## 4.3　文字表現追加の理論

ストローク間相違度は、書こうとしたストロークが別のストロークに認識される確率で決るのであるが、書こうとしたストロークとは、標準字体におけるストロークということになろう。したがって、いま2種類の文字辞書 $Dict_1$ と $Dict_2$ があり、$Dict_1$ はある文字の標準字体 $T = t_1,t_2,\cdots,t_n$, $T' = t_1',t_2',\cdots,t_n'$, の両方を、$Dict_2$ はTのみ含んでいるものとする。そして、その文字の入力ストローク列 $S = s_1,s_2,\cdots,s_n$, をストローク間相違度の設定に使うものとする。いま、$s_1 = t_1'$ かつ $s_1 \neq t_1$ と仮定する。辞書 $Dict_1$ では、$t_1'$ がそのまま $s_1$ ($s_1 = t_1'$) に認識されたものとなり、辞書 $Dict_2$ では、$t_1$ が誤って $s_1$ に認識されたものとなる。したがって、$Dict_2$ では $t_1$ と $s_1$ のストローク間相違度は、$Dict_1$ における値より小さくなり、入力文字に対して一定相違度以内の文字を認識する方式では、$Dict_2$ における標準文字の方が、より広い変動を認識する結果となる。

要約すると、標準字体数の異なる2つの文字辞書があるとき、標準字体数の小さい辞書において、各標準字体が相違度一定以内でカバーする領域の方が、標準字体数の大きい辞書で、各標準字体がカバーする領域よりも大きい。つまり、上記の方法でストローク間相違度を設定すれば、標準字体数が少ないときには、相違度で文字変動をカバーする比重が大きく、標準字体の追加を通じて、その数が大きくなってくれば、相違度の比重は減って、一文字に複数の標準字体でカバーする割合いが大きくなることを意味している。これは、標準字体を必要に応じて登録していく方式では、重要な性質である。この概略を図8に示す。




一文字あたりの標準字体が少ないうちは、相違度で字体の変動をカバーする。特徴空間におけるパタンの占める領域はおおまかに規定される。

一文字あたりの標準字体数が増してくると、相違度の比重は減って、どれかの標準字体で字体変動をカバーするようになる。そして、この方が特徴空間におけるパタンの占める領域の形をより正確に記述する。

図8　ストローク列特徴による特徴空間における標準字体追加の影響（概念図）

### 4.4 学習の可能性

以上の理論で個人差を反映するものは以下の２つである。

(1) 文字表現Ｔの出現頻度バイアスＰ(Ｔ) （表現Ｔを文字辞書に登録することは、Ｐ(Ｔ)を０からある値に修正することと考えられるので、これもこの範疇に入る。）

(2) ストロークｔがストロークｓと識別される確率Ｐ(ｓ｜ｔ)

項目(1)について、文字ｋを表すいくつかの表現を$T_1$, $T_2$, …,$T_l$とすると

$$P(T_i)=P(k)P(T_i \mid k)$$

Ｐ(ｋ)は文字ｋの出現確率であるから、入力(認識)対象の文章で決まる値である。したがって、Ｐ($T_i$)の個人差はＰ($T_i$｜ｋ)の形(つまり文字ｋを書く際に$T_i$の表現で書く確率)で表現される。

項目(2)についてはこの形で個人ごとに設定することができる。

両者とも入力者の傾向を学習していき、これらの値に反映していくことが可能である。また、個人辞書にすべきか、全体で1つの文字辞書で済むかは、Ｐ(Ｔ) (あるいはＰ(Ｔ｜ｋ))とＰ(ｓ｜ｔ)が各人でどれくらい異なるかに依存する。

### 4.5 ストローク生起の独立性仮定について

本来あるべきストローク列Ｔが、実際のストローク列Ｓになる確率Ｐ(Ｓ｜Ｔ)について、ストロークは独立に生起すると仮定して、

$$P(S \mid T)=\prod_{i=1}^{n} P(s_i \mid t_i)$$

としたが、これは構造解析的手法においてパタン・プリミティブの独立性を仮定していることになる。統計的認識においては、現実的レベルでの直交化をした後、それらの独立性を仮定することは妥当であろうが、構造解析的手法におけるパタン・プリミティブは、そもそもある構造(文脈)の中で相関をもって生起するものであるから、この仮定には危険が伴っている。しかし、この仮定によって、構造解析的手法においても、人間の経験的距離の概念が導入できることになる。したがって、この仮定に基づいて設定した相違度によって文字認識率の向上が見られるならば、これはＰ(Ｓ｜Ｔ)の一次近似として許容されるのではないかと考える。

## ５．認識実験

JOLIS-1の認識例を図9に示す。認識時間は0.7秒／字で、手書きの速度に十分追従できる。現時点における文字辞書の大きさは約30ｋBytesである。

JOLIS－1のインプリメンテーションを終えた際、作成者３名で予備的認識実験を行った結果、作成者以外の標本文字をできる限り大量に収集して、以下のことを行う必要性を認識した。

（1）標準字体の表現を追加してゆき、その認識率との関係を追跡する。

（2）標準字体の表現数の増加に伴い、追加すべき付加条件の推移を見る。

（3）認識モデルで述べたようにストローク間相違度を統計的に設定し、その効果を見る。

以上のことが、JOLIS－1における文字表現方式の妥当性の評価につながる。

さらに、より基本的なことであるが、JOLIS－1の各処理について標本文字データベースを基にした検討を加えなければならない。

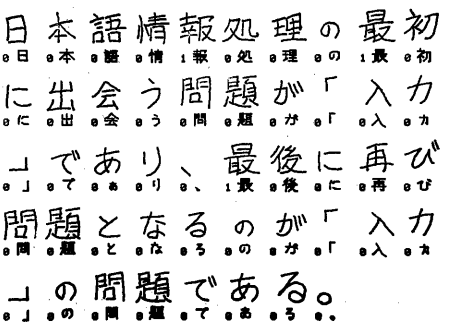



図９　JOLIS-1の認識例
認識結果と認識に要した相違度を
各入力文字の下に示す。

しかし、現時点ではデータベースの登録文字がディスク容量の不足から4名分しかなく、また集計を終えたのはそのうち1名分なので、本節ではその被験者の文字について統計的相違度の設定と文字辞書拡張の効果を報告する。

## 5.1 標本文字データベースを用いた統計的相違度設定の効果

被験者1名分の全認識対象文字の学習パタンと評価パタンについて、以下の評価実験を行った。

(0)経験的直感に基づいて設定した相違度(表3)と初期文字辞書$Dict_1$による認識実験

ここで、文字辞書$Dict_1$と学習パタンに対する上記実験を解析して$P(s \mid t)$を求め、$-\log_{10}$をとって表4に示す統計的相違度を得た。そして以下の実験を行った。

(1)統計的に設定した相違度と文字辞書$Dict_1$による認識実験

統計的相違度の設定によって、この被験者の文字パタンにおいて、表5に示す認識率の向上をみた。他の被験者のデータも集計して、数値の精密化を図る一方、相違度値の有効桁数の検討等を行う必要はあるが、表3の値を第一次近似と見ることはできよう。

表3 経験的に設定した相違度

| | | | 文字辞書内の標準ストローク | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 直線ストローク | | | | | | | | 複合ストローク | | | | | | | |
| | | | A | B | C | D | E | F | G | H | I | J | K | L | M | O | P | Q |
| 認識したストローク | 直線ストローク | A | 0 | 1 | 2 | 3 | 4 | 3 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| | | B | 1 | 0 | 1 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | C | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | D | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | E | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | F | 3 | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | G | 2 | 3 | 4 | 3 | 2 | 1 | 0 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| | | H | 1 | 2 | 3 | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| | 複合ストローク | I | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 1 | 2 | 2 | 1 | 2 |
| | | J | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | K | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 2 | 2 | 2 |
| | | L | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 |
| | | M | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 |
| | | O | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 |
| | | P | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 |
| | | Q | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |

表4 統計的に設定した相違度

| | | | 文字辞書内の標準ストローク | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 直線ストローク | | | | | | | | 複合ストローク | | | | | | | |
| | | | A | B | C | D | E | F | G | H | I | J | K | L | M | O | P | Q |
| 認識したストローク | 直線ストローク | A | 0 | 2 | 4 | 4 | 4 | 3 | 2 | 1 | 2 | 4 | 2 | 4 | 2 | 4 | 4 | 4 |
| | | B | 4 | 0 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | | C | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | | D | 4 | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | | E | 4 | 4 | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | | F | 3 | 4 | 4 | 4 | 4 | 0 | 2 | 2 | 2 | 4 | 2 | 4 | 4 | 4 | 4 | 4 |
| | | G | 2 | 4 | 4 | 4 | 4 | 2 | 0 | 2 | 2 | 4 | 2 | 4 | 2 | 4 | 4 | 4 |
| | | H | 3 | 4 | 4 | 4 | 4 | 3 | 2 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | 複合ストローク | I | 4 | 2 | 4 | 4 | 4 | 4 | 2 | 3 | 0 | 4 | 4 | 1 | 4 | 4 | 4 | 4 |
| | | J | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 4 |
| | | K | 4 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 0 | 4 | 4 | 4 | 4 | 4 |
| | | L | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 4 | 4 | 3 | 0 | 4 | 4 | 4 | 4 |
| | | M | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 3 | 4 | 4 | 4 | 0 | 4 | 4 | 4 |
| | | O | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 1 | 0 | 4 |
| | | P | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 4 |
| | | Q | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 0 |

表5 認識率の変化

| | 学習パタン | | 評価パタン | |
|---|---|---|---|---|
| | 直感的相違度 | 統計的相違度 | 直感的相違度 | 統計的相違度 |
| 正認識 | 86.2% (1002) | 89.0% (1034) | 87.5% (1017) | 89.8% (1043) |
| 誤認識 | 6.6% (77) | 6.8% (79) | 5.7% (66) | 5.9% (69) |
| 認識不能 | 7.1% (83) | 4.2% (49) | 5.7% (79) | 4.3% (50) |

(注) カッコ内はサンプル数

## 5.2 文字表現の追加と相違度再設定の効果

上記認識実験の結果を解析して、筆順間違い等のための文字表現を追加した文字辞書$Dict_2$を作成した。そしてさらに以下の実験を行った。

(2) 辞書拡張だけの効果をみるため、認識実験(0)での相違度と文字辞書$Dict_2$による認識実験

(3) 文字辞書$Dict_2$と、学習パタンに対する認識実験(2)を解析して設定し直した相違度による認識実験

実験結果を表6に、先の認識実験の結果も併せて示す。また、$Dict_1$と$Dict_2$の画数別による文字表現数および付加条件の必要なものの数を表7に示す。

表６．認識結果

| | 認識実験(0) $Dict_1$ | | 認識実験(1) $Dict_1$ | | 認識実験(2) $Dict_2$ | | 認識実験(3) $Dict_2$ | |
|---|---|---|---|---|---|---|---|---|
| | 学習パタン | 評価パタン | 学習パタン | 評価パタン | 学習パタン | 評価パタン | 学習パタン | 評価パタン |
| 正認識 | 87.2% | 88.4% | 90.0% | 90.6% | 94.7% | 93.5% | 95.4% | 93.5% |
| 誤認識 | 5.8% | 4.9% | 5.9% | 5.2% | 2.6% | 3.8% | 2.8% | 3.9% |
| 認識不能 | 7.0% | 6.7% | 4.1% | 4.2% | 2.7% | 2.7% | 1.8% | 2.6% |

表７．文字辞書 $Dict_1$ と$Dict_2$ における全表現数と付加条件の必要な表現数の割合

| 画数 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12~20 | 合計 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Dict_1$ | 全文字表現数 | 22 | 57 | 101 | 109 | 162 | 184 | 250 | 264 | 262 | 252 | 244 | 797 | 2704 |
| | 付加条件つき | 5 | 39 | 45 | 35 | 35 | 22 | 12 | 10 | 12 | 3 | 1 | 0 | 219 |
| $Dict_2$ | 全文字表現数 | 22 | 57 | 101 | 110 | 163 | 189 | 252 | 275 | 269 | 266 | 252 | 815 | 2771 |
| | 付加条件つき | 5 | 40 | 45 | 36 | 36 | 25 | 13 | 10 | 13 | 3 | 1 | 0 | 227 |

$Dict_1$から$Dict_2$への拡張に伴ってストローク間相違度を補正したが、変更があったのは識別結果と標準ストロークの組で、(H，F)の3から4、(G，A)の2から3、(M，I)の3から4、の3組だけであった。これは、この拡張が時間的制限から極めて小規模であったことと、相違度値の有効桁数が大まか過ぎたことに要因があろう。結果として、文字認識精度への寄与は学習パタンには見られるものの、評価パタンでは確認されなかった。

## 5.3 JOLIS－1の問題点

以下、認識実験を通じ明らかになったJOLIS－1の問題点を列挙する。

(1)ストローク列による文字表現は、10画以上の文字については一意に文字を決定するが、低画数の文字では付加条件に頼らざるを得ない。また、低画数の文字ではストロークの誤識別の影響が大きい。

(2)ストロークの８方向セグメントによる近似は、漢字と片仮名には十分でも、平仮名、記号まで含めると、その分解能の低さから、幾つかの文字、たとえば、“つ”と“フ”、“し”と“レ”等、が同一のセグメント列で記述されてしまう。低画数の文字では、ストロークの形状抽出をもう少しきめ細かく行う必要がある。

(3)ストロークのはね、おさえ等の修飾を除去するために、ストロークの始めと終りの部分を一様に除去しているが、この方法だと、修飾を取りきれない場合や、本来あるべきセグメントを除去してしまう場合が起こる。

(4)ある部分パタン(偏、つくり等)を誤った筆順や、略して書く習慣を持っていると、その部分パタンを含むすべての文字で認識してもらえない結果となる。ある被験者の場合、この理由による誤認識・リジェクトが1162文字中40文字近くもあった。JOLIS－1でこれらの文字を登録するためには、それらの文字１つ１つについて、自分の筆順、あるいは、略字体を辞書に加えなければならない。

## ６．おわりに

JOLIS方式は前節に述べたようにまだ幾つかの本質的な問題を抱えている。しかし、JOLIS-1はそのような問題点を析出させるというだけでなく、同時に一定の成果を示したと考えている。それらは以下のとおりである。

(1)平仮名、片仮名、教育漢字および幾つかの記号の混在した対象文字に対して一応の認識率を示したこと
(2)独自のセグメント化処理を考案し、その有効性を示したこと
(3)ストローク表現におもに方向変化を用い、その安定性と、識別の高速性を示したこと
(4)オーバーラッピング・ストロークによる文字辞書の縮小化を実証したこと
(5)相違度(とくに統計的相違度)の採用による認識率の向上を示したこと
(6)標準字体、標準ストロークをすべてデータで表現し、その追加と変更の容易さを示したこと
(7)構造解析的手法に統計的相違度の概念を組み入れることに成功したこと

現状では、データベースの規模が不十分で、JOLIS-1方式の限界を見極めることには至っていない。したが

って今後、データベースを拡充して、評価の精密化を図る必要がある。また、文字辞書の拡張とそれに伴う相違度の調整を進めていくことにより認識率をどこまで向上させられるのか、興味深い問題である。その際、認識率の向上を阻むのは、文字辞書の拡張に伴って追加しなければならない付加条件の増大であろう。またJOLIS-1は文字認識システムとしてのみならず、オペレーティング・システムＯＳ／ｏ下で、清書システムJOSHOと結合して、日本語文書作成システムとして実用に供し、その人間工学的評価を行いたいと思う。

日本語文字認識の難しさは、その字種の多さばかりではなく、低画数の仮名から高画数の漢字へと及ぶ、その幅広い字体のスペクトラムにある。我々はこの興味深い対象に対して、構造解析的手法を基本とし、統計的相違度の概念を取り入れて、両者の長所を生かした認識方式を確立したいと考えている。

謝辞

本研究において御助言、御支援を賜っている土井康弘明星大学教授、本学科の阿刀田央一助教授、金子俊一助手に深謝する。またJOLISプロジェクトに参加した、五十嵐道弘（現・日本電気）、木村慎一（現・キヤノン）、青木克郎(現・日本電気)、真鍋俊彦(現・東芝)、伊藤幸雄（現・日立メディコ）の各位に謝意を示す。

参考文献

[１] Fu, K.S, :"Syntactic Methods in Pattern Recognition", Academic Press, New York (1974)

[２] Ikeda, K. et al:"On-line Recognition of Hand-Written Characters Utilizing Positional and Stroke Vector Sequence", Pattern Recognition vol.13, no.3 pp.191-206 (1981)

[３] 池田裕治 他:"オンライン手書き文字データベースの試作と、それを用いた評価の方針", 本学会第26回全国大会予稿集 (1983)

[４] 中川正樹 他:"オンライン手書き文字認識システムJOLIS-1の設計と試作", 本学会日本文入力方式研究会資料3-1 (1981)

[５] Nakagawa, M. et al:"On-Line Recognition of Handwritten Japanese Characters in JOLIS-1" ,Proc. of the 6th ICPR, Munich, pp.776-779 (1982)

[６] 中川正樹, 阿刀田央一, 青木克郎, 池田裕治, 高橋延匡:"オンライン手書き文字認識システムJOLIS-1における統計的辞書マッチングとデータベースの活用", 本学会第26回全国大会 (1983)

[７] Nakagawa, M. et al:"On-line Handwritten Charactar Recognition as a Japanese Input Method", Proc. of Intl. Conf. on Text Processing with a Large Character Set, Tokyo (1983)

[８] 高橋延匡:"日本文入力の現状と展望", 情報処理 vol.23 no.6 (1982)

[９] Takahashi, N. et al:"浄書:Japanese Output Server with HOspitality", Proc. of Intl. Conf. on Text Processing with a Large Character Set, Tokyo (1983)

[10] 高橋延匡　他:"MC68000用小型OS:OS/ｏの開発", 本学会計算機システムの制御と評価研究会資料21-6 (1983)

付録　ハードウェア環境

JOLIS－１は主記憶64kbytes、10Mbytesのディスクを有するミニコンピュータHITAC-10Ⅱ/A上に実現されている。また入力装置として、分解能0.1㎜、サンプリングレイト100点/秒のタブレットを用いている。入力ペンには、タブレットとの圧着情報を出力するために、マイクロスイッチが組み込まれている。しかし、その遊びが大きく、普通の大きさの文字が書きにくいため、文字記入枠は2㎝×2㎝としている。