人工知能と対話技法 23－3
（1981. 11. 26）

2次元データ入力のためのオンライン手書き文字認識

佐伯 元司　　松下 武史
（東京工業大学 工学部 情報工学科）

1、はじめに

図面や数式といった2次元表現においては、文字や記号だけでなく、それらが書かれていた位置やその大きさも重要な情報である。さらに2次元表現では各種の文字が用いられる。キーボード装置は1次元の文字列しか扱えず、また文字セットが限られている。このため2次元表現の入力にはキーボードは適当でなく、オンライン手書き文字認識を用いて手で書いたものを直接入力することが望ましい。本報告では2次元表現の入力を目的としたオンライン文字認識について述べる。

過去にも手書き文字認識の研究は種々行われてきたが、1）2次元表現に使用される種々の文字・記号を統一的に扱えない。 2）入力ストローク列から文字を切り出す、いわゆる文字の自動分離ができない。このため文字が任意の位置に任意の大きさで書けない。 3）大型計算機、もしくはミニコン上でしか実用にならず、データ入力装置として使用するには実用的でない。などの問題があった。本研究では、

1）多種の文字を認識対象とし、字形をあらかじめ制限しないために、できるだけ人が文字を区別する際に意識して用いる特徴を使う。

2）文字が任意の場所に任意の大きさで自由に書けるために、入力ストローク列のどこからどこまでが1つの文字を構成しているかを自動的に判定する機構を組み込む。文字の分離はストローク型の組み合わせを使って自動的に行い、ユーザーは明示的に文字の区切りを知らせる必要はない。

3）入力装置としての実用性を上げるため、種々のシステムに組み込めるようにマイクロコンピュータを使用し、汎用性の高いコンパクトなシステムとした。このため、処理速度・メモリ容量の面からも十分実用になる簡単なアルゴリズムを開発した。

4）ユーザーが認識システムに慣れるための練習を援助する機能を設けた。誰が書いた文字でも90%位までの認識率をあげるシステムの実現は比較的容易であるが、それを越えると難しくなる。実験の結果、これは書き順・字体が異なったり、筆記具に不慣れなことが主な原因であるとわかった。タブレットのペンは使いにくく、慣れないと正しくデータが入力できないことがある。また文字のなかには、大文字・小文字の'O'などのように文脈を使わない限り判断できないものもある。これらは、0，oのようにある程度字体を制限せざるを得ない。このような理由により、最初ユーザーに若干の練習をしてもらい、システムに慣れてもらうのが現実的である。

以上の項目に基づいて、実用的なシステムを作成したので報告する。ほとんどのユーザーは1時間程度の練習で98%の認識率を上げることができた。

2、文字の特徴

2、1 特徴抽出の方針

人が一般の2次元の形状を識別するプロセスは明白ではないが、文字ではそれ

それの形状の特徴を記述し、どのようにしてそれらを互いに区別するかを述べることができる。例えば、'a'の場合では輪を描き、その後上から下へ進み、左回りで右上へはねる。'g'では前半までは同じであるが、最後は右に回って右上へはねる。ストロークの方向・曲がり方・位置関係などのどの特徴が重要であるかは個々の文字により異なる。多種の文字を扱うために、このような人が意識して区別する特徴をできるだけ用いて分類・識別することを基本の方針とする。

2次元の特徴抽出を人と同じように行うことは現実的には難しく、計算機で処理しやすいやり方に置き換えなければならないことがある。このため、変形した文字では人には読めても計算機では認識できないことが起こる。これを最小限にするため、実験により、どのような変形が多いかを調べ、その処理を特徴抽出に組み込む。さらに文字の区別に必要な特徴だけでなく、追加のチェックを行い、予想外の変形をした文字はできるだけリジェクトする。オンラインではリジェクトはユーザーに知らせることで直ちに入力し直すことができるからである。

2.2 抽出する特徴

実現したシステムでは2次元表現の入力用ということで、漢字、ひらがな、片仮名などの文字は除外し、英数字(小文字も含む)といくつかの記号を対象とした。ここで抽出する特徴は以下の通りである。 1)ストローク各部の曲がり方：直線，左回転，右回転，どれくらい曲がっているか、輪になっているか。 2)ストローク各部の方向：直線ならその方向、曲線ならどちらを向いているか。 3)尖点の有無 4)特徴点の位置関係：ストロークの始点、終点，尖点，縦横方向の極大・極小点などの位置関係 5)ストローク各部の大きさ さらに複数ストロークで構成される文字においては 6)文字を構成するストロークの型 7)ストローク間の位置関係：上下左右の方向，距離 8)ストロークの大きさの比

1)のストロークの曲がり方については接線方向の変化を調べ、それによって判定を行う。曲線の向きは縦横方向の増減状態を調べることによって判定する。交点や接近箇所は人にとっては容易に抽出でき、わかりやすい特徴であるが、計算機では時間のかかる処理である。現在はこれらを直接抽出することは行わず、必要ならば始点，終点などの位置関係を調べることにより交点の有無を判定している。

2.3 文字分離の方針

ストローク間の距離は文字の区切りを判定する重要な基準であるが、人は単純に一定の距離以上離れているかどうかで判定しているのではない。ストロークが多少重なっていても2つの文字に読んだり、離れていても1つの文字として読む場合がある。人は複数のストロークの形と位置関係を見て、それらが有意な文字を構成し得るかどうかで文字の区切りを判断している。本システムではこの方式を効率的に使って、文字の分離を行っている。

2.4 識別の手順

1)入力ストロークに沿って長さと接線方向を求める。

2)短区間で方向が急変している箇所を見つけ、尖点とする。ストロークを尖点で分割する。これはストローク型の判定を行いやすくするためである。分割した

ものをセグメントと呼ぶことにする。
3）各セグメントごとに特徴を抽出し、分類する。
4）大きな特徴を持ったセグメント型を使って、ストローク型を分類する。
5）ストローク型の出現列、及びその位置関係などの特徴から文字の区切りを判定し、識別する。

3、文字の変形

システムは入力された文字を正しく認識できないことがある。それは
1）筆記具の扱いづらさから、データが欠けることがある。（図1）
2）形の同じ文字があり、単独では人も判断できないことがある。
3）前章で述べた特徴抽出は人のそれと全く同じでないため、変形に対して不変な特徴パラメータとなっていないことがある。

1），2）に関しては1章でも述べたように筆記者が練習を行うことによって改善させる。ここでは3）について述べる。どのような変形が生じ、それにより特徴抽出がどのように影響を受けるかを調べ、対策法を提案する。

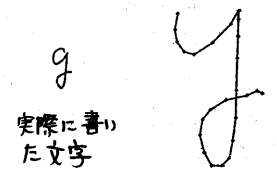


図1 データの欠け

3、1 局所的な変化

人は直線を書いたと思っても、実際に計算機に入力されるデータには、図2のように筆記者のブレやペンの押え、ハネ、さらにはノイズなどが重畳し、局所的に方向が変化している箇所がある。前章で述べたストロークに沿って方向変化を調べていく方法では、これらにより回転方向の判定を誤ったり、尖点を誤認したりする。比較的小さいものについては平滑化や標本化で除去できる。ペンの押えやハネといった大きい方向変化はセグメントの始点・終点付近によく出現するため、始点・終点の一定範囲内は回転方向の判定から除外する。また一定長にも満たない短いセグメントはノイズとみなして、無視する。

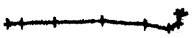


図2 局所的な方向変化

3、2 尖点

本システムではどんな文字でも尖点で分割しているが、実際は文字によって尖点の重要度は異なっている。尖点がどのように変形するかを調べると図3のようになる。同図(a)ではもともと方向変化の小さい尖点（弱尖点）がなまって消えた場合、(b)では逆に尖点の前後で逆方向に曲がり強調された場合である。方向変化の大きい尖点（強尖点）は(c)のようにループになってしまうことがある。尖点の検出はただ単に方向変化が一定値以上になるかどうかで行うの

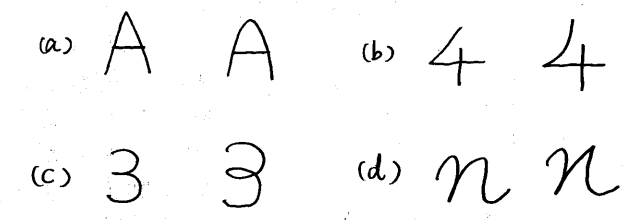


図3 尖点の変形

ではなく、(b)や(c)のような場合も考慮して行う。尖点が消失するという変形は(a)のように筆記者がなまらせて書いた場合だけでなく、4章で述べる平滑化や標本化といった前処理で失われる場合もある。(図4) これらの対策として1)セグメント型の識別ができなかったときは、尖点が検出できなかったとみて、セグメント中で最も方向変化の激しい箇所で分割し、再度調べる。2)短区間に変曲点が2つ出現したときは、その中点を尖点として分割する。この他に図3(a)のように消失の可能性のある弱尖点を持つ文字については両方のパターンを辞書に登録しておく。

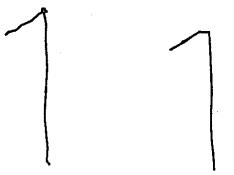


図4 尖点の消失

尖点が消失する現象とは全く逆に図3(d)のように曲線部に余計な尖点が出現することがある。出現する可能性のあるセグメントに対しては、以下の修正規則を使ってセグメントの合成を行う。例えば、変曲点を含むパターンにおいてはよく尖点が出現し、∩型のセグメントは↑と⊂の2つのセグメントに分割されることが多い。そこで↑と⊂というパターンが連続したならば、これを合成し、∩型に置き換える。修正規則は尖点の変形を扱うためだけでなく、字体のバリエーションを減らすものもある。図5に修正規則を示す。修正規則が適用できない変形については辞書を2通り用意する。この他に標本化を行う際にも方向変化の大きい箇所はその効果を残らし、なめらかな曲線を保存するようにしている。

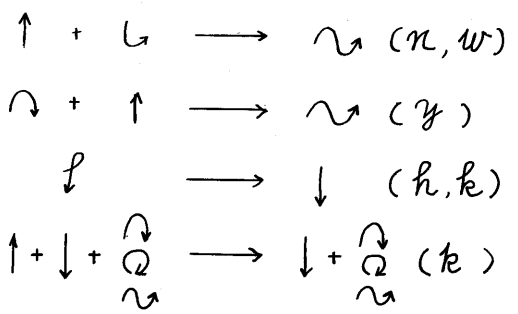


図5 修正規則

3、3 直線部の湾曲

図6のように標準パターンでは直線である箇所が変形して曲線に書かれることがある。同図のHの標準パターンでは1ストロークは|であるが、変形パターンでは)である。単独ではもちろんこれらのストロークは区別して扱わなければならないが、この後に2, 3ストロークが続くと同一視される。つまり)というストロークは状況により右括弧か直線かのいずれかに解釈される。この変形は以下のような方法で取り扱う。曲がり方の小さい曲線は曲線の性質を記録しておくが、いったん直線と判定し、ストローク型の判定, 文字の識別を行う。その結果、セグメントが直線か曲線かの判定が必要になったとき、記録しておいた特徴を使って判定する。

図6 直線部の湾曲

4、特徴抽出のアルゴリズム

本章ではタブレットから入力されたストロークより特徴を抽出し、型を判定するアルゴリズムについて述べる。

4、1　前処理

タブレットから入力された座標データに以下のような前処理を施す。

1）ストロークの始点の補正：ペンスイッチの動作の遅れにより生じるストロークの始点近傍のデータの欠けを補うため、ペンダウン直前の1点をデータに付け加える。

2）平滑化：筆記者のチブレやディジタル化の誤差を取り除くために隣り合う2点の移動平均をとる。

3）標本化：冗長な点を取り除くため、図7のようなウィンドウフィルタリングを施す。これは有効点P_1を中心として正方形(ウィンドウ)を定める。この正方形内にある点は冗長な点として捨て、外にある点、を次の有効点とする。曲がりの急激な所ではウィンドウを小さくし、逆に直線部では大きくする。ここではA(ストロークの大きさ ── 外接する方形の長辺 ── の1/16)，B(1/24)，C(1/32)の3段階のウィンドウを使い、隣点との方向変化が67.5°以下になるようにしている。

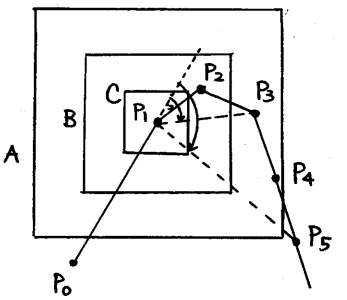


図7　ウィンドウフィルタリング

フィルタリングと同時に各点の接線方向と長さを求める。

4）終点付近のハネの除去：終点の近傍で方向変化の激しい箇所を探し、もしあれば、その箇所から終点までをハネとみなして除去する。

図8に前処理後データと接線方向の変化のグラフを示す。

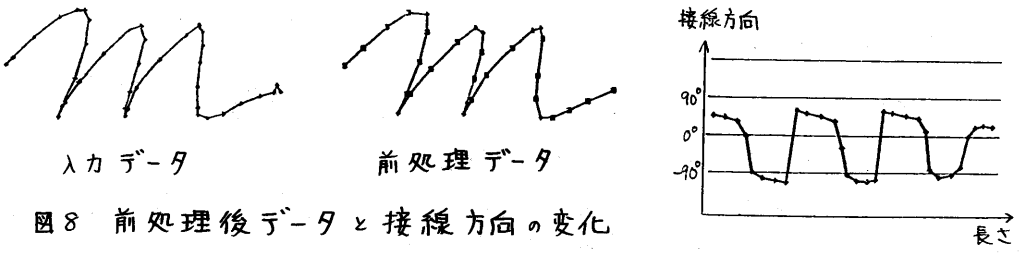


図8　前処理後データと接線方向の変化

4、2　尖点の検出

ストローク中の尖点を検出し、その箇所で分割する。これは尖点の前後では回転方向などの特徴が不安定であり、分割することによって安定した特徴を容易に抽出でき、型の判定が行いやすくなるためである。

尖点は以下の方法で検出する。筆跡に沿って、距離4ℓ(ストロークの大きさの1/16)の区間で方向変化が90°以上ある区間を見つける。この条件を満たす区間のうち、

1）4ℓ以内で方向の変化が十分大きくて、140°以上ある。

2）条件を満たす区間が連続し、全体として140°以上の方向変化がある。(ループの尖点：図3(c))

3）その直前あるいは直後の筆跡が逆方向に曲がり、方向変化が100°以上ある。(図3(b))

また回転方向を調べていったときに

4）ストローク長の1/10以内の区間に変曲点が連続する。

のうち、どれか1つを満たすものを尖点とする。

4、3　セグメント型の判定

検出した尖点でストロークをセグメントに分割する。各セグメントの回転方向、始点・終点の位置関係や方向、ストローク中での出現順を求める。さらに縦横方向の増減を調べることにより、セグメントの向きと極値点を求める。ただし、各セグメントの始点・終点付近のストローク長の1/15以内の区間での方向変化は無視し、1/20以下の長さのセグメントは無視する。直線か曲線かの判定は方向変化が45°以上あるかどうかで行う。

型を判定するための辞書は処理を高速に行うため、decision tree構造とし、まず回転方向、セグメントの向きといった特徴で大きく分類し、次第に各セグメント固有の特徴で細かく分類する。図9に辞書の一部を示す。

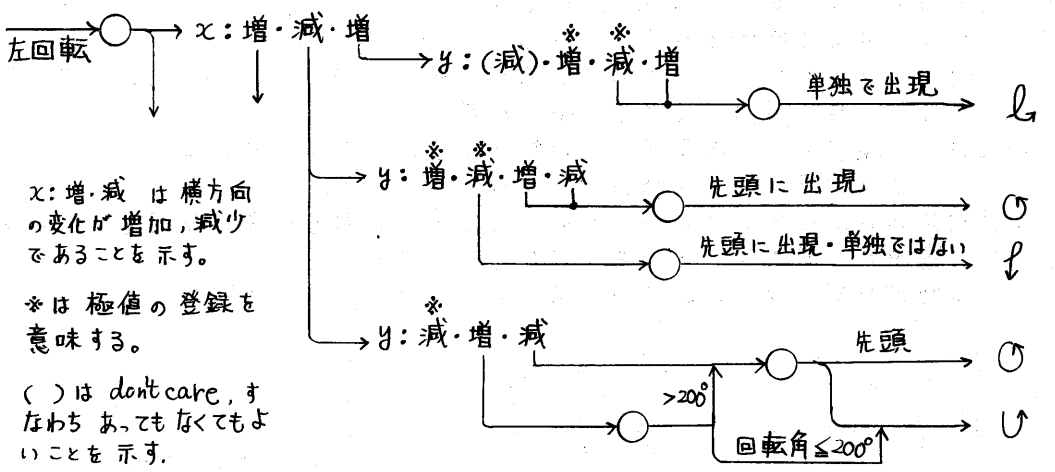


図9　セグメント型 判定 tree

セグメント型に分類できなかったときは、そのセグメント内で最も方向変化の大きい箇所で再分割し、もう一度判定をやり直す。

図10に認識対象文字を英数字にした場合のセグメント型を示す。例えばa，gの場合にはそれぞれ↺と↳，↺と↱という2つのセグメントから成っている。

4、4　ストローク型の判定

判定されたセグメント型を使ってストローク型を分類する。まず、セグメントの出現列に対し、図5の修正規則が適用可かどうか調べる。適用可能ならば、その規則に基づいてセグメント型を変換する。変換前のセグメント型も分類に使用する場合があるので、捨てないで残しておく。

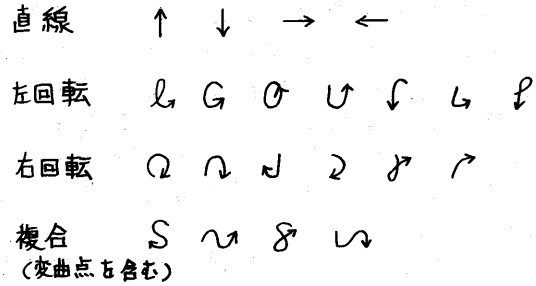
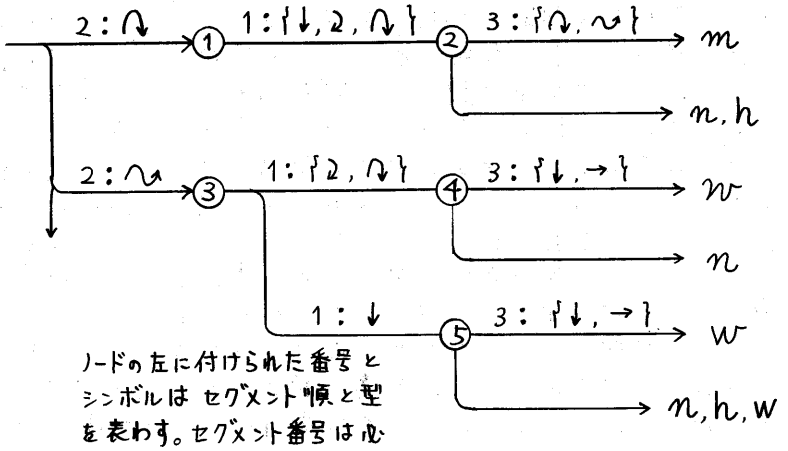


図10　セグメント型

英小文字には m, m, m などのように書き始めや書き終わりのセグメントに様々なバリエーションがあるため、不変のセグメントにまず注目して分類を行う

ていく。mの場合では第2セグメントの∩が不変である。このように大きな特徴を用いることにより、ストロークを確実に分類し、範囲を狭めることができ、処理の高速化に有効である。ストローク型分類用の辞書もtree構造になっている。図11にその一部を示す。



ノードの左に付けられた番号とシンボルはセグメント順と型を表わす。セグメント番号は必ずしも必要ではない。ノードの下から出ている枝は入力データが番号で指定されたセグメントを持っていない場合である。

図11 ストローク型判定tree

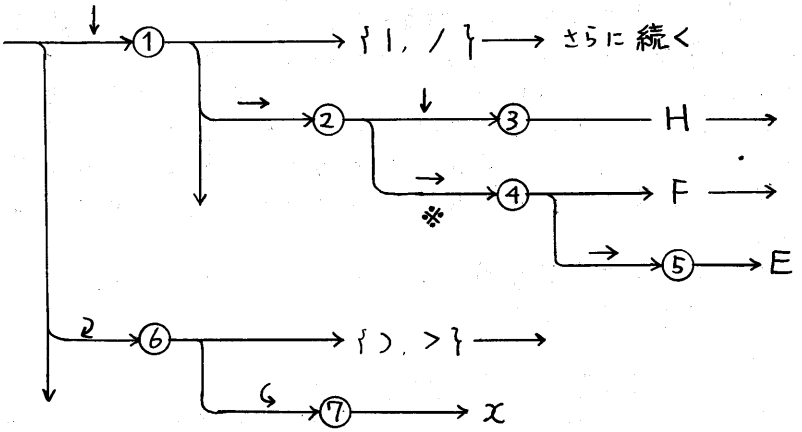
5、文字の分離と識別

文字の自動分離はストローク型の出現列から有意な文字を構成し得るかどうかを調べることによって行う。このために図12のような識別treeを用意した。このtreeの各枝にはストローク型と以下のような条件が付加されている。1)ストローク間の位置関係：ストローク中の始点・終点などの特徴点の位置で表わす。2)ストロークの回転角 3)始点から終点への方向，始点・終点付近の方向 4)ストロークの縦横比

次に入力されたストロークの型に合う下位への枝がなかったり、あってもその枝に付けられた条件を満足していなければ、そこを文字の区切りと判定し、分類を行う。

唯一の文字に分類できても、そこでチェックをやめず、文字として正しい形をしているかどうかの追加のチェックを行い、あいまいな文字をリジェクトする。

文字の分離に失



ノードの左上のシンボルはストローク型を表わす。各枝に付けられている結合条件は省略したが、例えば※には「第1，2ストロークを囲む方形領域内に第3ストロークの重心が含まれる」という条件が付加されている。

図12 文字の分離・識別 tree

敗したとき、つまり図12のノード②のような有意な文字が全く存在しない箇所が文字の区切りと判定されたときは、そこまで通ってきた経路中で有意な文字を含む最も近いノードで分離し直す。この場合、ノード②に最も近いノードは①であるから、ここで分離が起こり、識別結果は1か/になる。

この他に、一部のストローク型に2重の意味を持たせている。例えば、回転角が90°以下の2には本来の形状の意味以外に↓という直線ストロークの意味も与える。)というストロークが入力された場合、最初はこれを↓とみなして分類を行っていく。分類を行っていった結果、どの複数ストローク文字にもならなかったときには、本来の形状に戻して再び識別を行っていく。この例では、最初ノード①に達するが、識別に失敗すると、treeの根に戻り、ノード⑥に達する。

文字の分離を行う手段としてはこの他にストローク間の入力待ちの時間を使っている。つまりストロークを書き終えてから0.8秒以上経っても次のストロークが入力されないときは、そこで無条件に文字の分離を行う。

6、練習機能と認識実験

6、1 練習機能

練習の目的は

1)書き方が標準と異なったり、書き方に着しくくせのある文字が誰でも普通いくつかある。このようなくせを知り、標準の書き方を覚える。

2)タブレットペンに慣れる。

ことである。このための練習を援助することを目的として、以下のような機能を組み込んだ。書いた文字が正しく認識されなかったとき、計算機が読み込んだデータと抽出した特徴をわかりやすく図形で表示して、筆記者に書き方の悪い所が直ちにわかるようにした。図13にその表示例を示す。これは抽出した特徴のうち、ストローク数とセグメント型を使って合成したもので、必要なら、他の特徴も表示することもできる。また表示だけからはわからない時のために標準字体を書いた表も用意している。

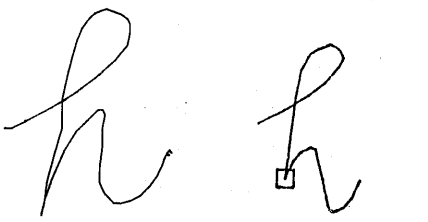


図13 練習援助機能

実際の練習では計算機が書く文字を指示し、筆記者の書いた文字が正しいかどうか自動的に判定する。

6、2 認識実験と結果の検討

実験には、分解能0.1mm, サンプル速度64点/秒の磁歪式タブレットと汎用の8080マイクロコンピュータを使用した。使用メモリ量はプログラム(作業領域も含む)が24kB, 辞書が6kB, 処理時間は文字によっても違うが、0.5～1.5秒であった。

練習の効果がどれくらいあるかを調べるために、全く予備知識の与えられていない複数の被験者に本システムの練習機能を使ってもらい、練習過程での認識率を測定した。実験対象文字は英数字62文字＋特殊記号16文字とした。また数日後、同じ被験者に対し練習効果がどれくらい残っているかを調べるために同じ実験を

行った。

8人の被験者に対し、実験を行い、認識率を測定した結果を図14に示す。1回の練習では全対象文字を1回ずつ書いた。表1に第1回目及び第4回目の主な誤り原因を示す。

第1回目の認識率は80～97%と巾があった。英字や特殊記号は字体や書き方が統一されていないため、実験中にも様々な書き方が見られた。文字の形がもともと似ており、本システムでは字体を変えて区別している文字の誤り（例えば、小文字・大文字のO）やタブレットに慣れていないことからくる誤りも多い。これらを合わせると誤り原因の80%以上になり、筆記練習が必要なことがわかる。

第4回目では、これらが改善され、認識率も97～100%となった。ここまでの練習時間は30分～1時間、文字数にして200～300文字程度であった。逆に4回目を過ぎると認識率が若干落ちてくる。これは筆記者の手が疲れてしまったためと考えられる。

練習の効果は数日後でも十分残っており、標準字体を忘れているという現象は2、3見られただけであった。

リジェクトと誤認識は第4回目ではほぼ同数となってしまい、リジェクトを多くするという方針をあまり満足していない。これは誤り原因の大部分が文字の形があいまいであり、似た文字同志の区別がうまくできなかったという理由による。似た文字の区別をはっきりと行うには、1)まぎらわしい文字の字体を変

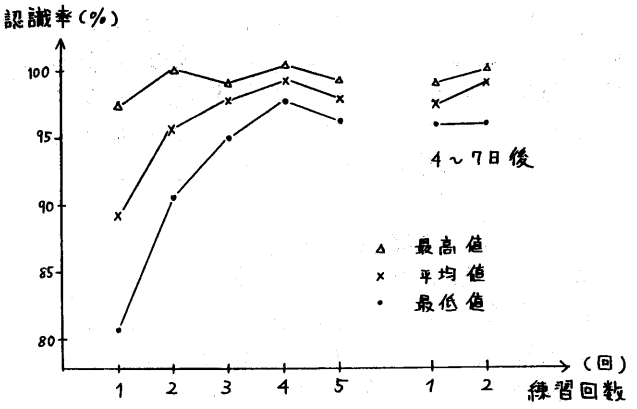


図14 練習過程における認識率の変化

表1 主な誤り原因

	第1回目		第4回目	
	誤認	リジェクト	誤認	リジェクト
1 書き方があいまい	3.9	0	0.47	0
2、書き順，字体が違う	2.4	1.9	0.21	0.1
3、ペンの不慣れによるデータ欠損	0.9	0.9	0.12	0.35
4、尖点が検出できなかった	0.7	0.3	0	0.12
5、手ブレによる尖点出現	0	0.8	0	0.12
合計（誤った割合）	7.9	3.9	0.8	0.7

表中の数値は全入力データ数に対する割合(%)

（誤りの例）

1. O（小文字）を 大文字と誤認
   r（r）を Vと誤認
2. Gを ①→②↓① と書いた（aと一）
   Eを ①→②→③ と書いた（cと一）
3. 図1 参照。
4. hの尖点が前処理で失われた。hに誤認


前処理後データ

え、区別がはっきりとつくようにする。2)適切な特徴を選択する。ことが考えられる。1)については大文字・小文字のOの区別などに使用しているが、ある程度筆記練習が必要になる。また筆記者があまり違和感を感じるような字体に変えるのは好ましくない。2)については、表1のrとVの区別などのように適切な特徴の抽出が計算機では難しいものもある。

6、3　文字分離機能の評価

文字の分離は良い結果が得られ、2次元表現の入力に十分使用できることが確認できた。図15に手書きで数式を入力した例を示す。数式中の文字は完全に分離でき、正しい認識ができた。

$$\frac{\frac{c}{2}}{a^2+b^2}+\frac{d}{3}$$

((C)/(2))/(a**(2)+b**(2))+(d)/(3)

上が入力データ，下が計算機の解釈

図15　数式入力例

いったん文字の分離を誤ると、その後の文字の分離にまで影響が及ぶ可能性があるが、実験では1文字後までしか影響が現われなかった。これは分離を誤るとほとんどリジェクトになり、そのため筆記者が気付いて運筆を中断するためである。文字分離を誤り、誤認識をひき起こしたのは、人が見てもまぎらわしい場合だけであった。

7、おわりに

本研究ではマイクロコンピュータ上で実現可能なオンライン文字認識の一手法を示した。識別は人が文字を区別するときに注目していると考えられる簡単な幾何学的特徴を抽出して行った。この手法は英数字・特殊記号だけでなく、ループの検出や交点の検出機能を付け加えれば、ひらがなやカタカナの認識にも拡張可能である。また文字分離のアルゴリズムは漢字のへんやツクリを分離、識別にそのまま応用することができ、漢字の認識にも有効である。

また、認識実験の結果や数式入力例より本システムが2次元表現の入力に有効であることが確認できた。認識率は1時間程度の練習でほとんどの人が98%以上となった。入力装置として使用したタブレットペンは非常に使いづらいため、一度に大量の文字を入力するのは筆記者に対してかなりの負担になるであろう。使いやすい筆記具の開発が今後の課題である。

謝辞：本研究に対し、日頃、御指導頂いた東京電機大学の徳坂衛教授、本学の榎本肇教授，ならびに熱心に御討論頂きました研究室の皆様に深く感謝いたします。

参考文献

1、M. Hosaka , F. Kimura : An Interactive Geometrical Design System with Handwriting Input , Proc. IFIP'77, vol 7, PP167～172 (1977)
2、G. M. Miller : On-line Recognition of Hand-generated Symbols , Proc. AFIPS, vol 35, PP 399～412 (FJCC 1969)
3、藤原他：接線ベクトル列を用いたオンライン手書き文字の認識，情報処理，vol 17, No 3, PP191～199 (1976)
4、松下他：手書き文字・シンボルのオンライン入力，第21回情処学会予稿集