

指で文字を書く動作の 3 次元解析

島田 英之・古川 稔晃*・島田 恭宏
大倉 充・塩野 充・宮垣 嘉也

岡山理科大学工学部情報工学科

* (株) 総合システムサービス

(2001年11月1日 受理)

1 まえがき

文字認識の研究の歴史は長く、近年では OCR (Optical Character Reader) による紙上の文字の認識はごく普通に行われている。手書き文字に関してもその発展は著しく、ペン入力を用いたオンライン手書き文字入力による携帯端末も広く普及するに至った。

現在の文字入力インターフェイスの主流は、携帯電話に採用されている方法、すなわち、各キーに複数の文字を割り当てておき、キーを連続して押すことにより目的の文字を選択して入力する方法である。また、前述のオンライン手書き文字認識を応用し、タッチパネルに専用のペンを用いて文字を入力する方法も多く用いられている。前者は、場所を選ばず、入力装置をコンパクトにできる利点はあるものの、操作に慣れが必要であり、高速な入力は望めない。また、後者は、文字を書くという動作はごく自然であるが、筆圧を感知できるタッチパネルなどが必要となる点が利用の制約を生む。

最近では、以上のような背景や、3次元位置入力デバイスの普及によって、仮想空間内に文字や図形を描く空気ペンの試み [1]、発光するペンによる空中署名をビデオカメラで撮影して個人認証を行う試み [2]、3次元位置入力デバイスによる指書き数字認識の試み [3] などが報告されている。

本稿では、小型軽量な3次元位置センサを人差指に装着し、机の上、車の窓などの任意の平面に文字を書き、曲率などの柔軟な特徴量を用いることによって文字を認識する手法について提案し、実験を行なったことについて報告する [4]。なお、今回の実験での認識対象は、「ゐ」、「ゑ」、濁点、半濁点を除く平仮名 46 文字に限定した。

2 概要

2.1 3次元位置センサについて

本研究では、3次元位置センサ（以下、単に位置センサと呼ぶ）を人差指の先端に装着して文字の入力を行う。本研究で用いた位置センサ 3 SPACE ISOTRAK II (POLHEMUS 社製) は、トランスミッタが生成する磁場をレシーバが感知し、トランスミッタとレシーバの相対位置座標 (x, y, z) 、レシーバの方位、仰角、横転の6種の情報を20分の1秒ごとに得ることができる。なお、本研究では、このうち (x, y, z) の情報のみを用いた。

位置センサを装着して文字を入力している様子を、図1に示す。

2.2 空間内における文字ストローク

人差指に装着した位置センサにより得られる情報は、3次元空間内にて、速度が変化しながら軌跡を描く (x, y, z) の座標列である。ゆえに、まず最初に、実際の文字が存在する平面（以後、文字平面と呼ぶ）を特定し、3次元座標列を文字平面に投影して、通常の2次元の文字データに変換すれば、一般的な文字認識手法が適用可能となる。

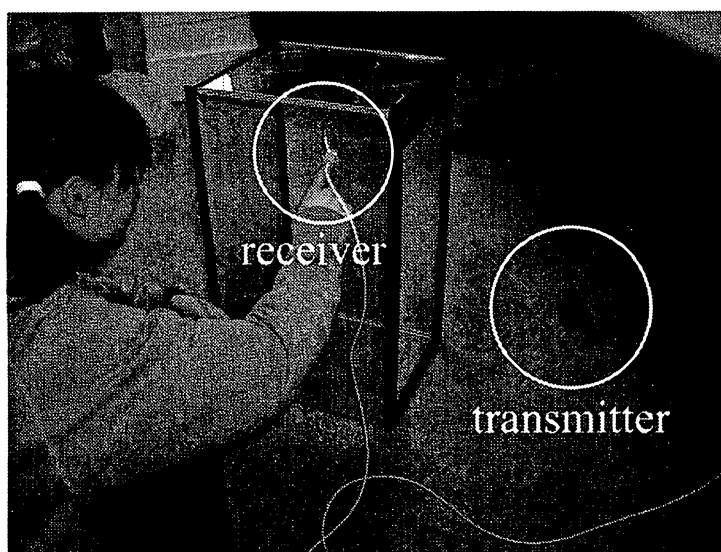


図 1: 入力の様子

図 2 に、文字「あ」の 3 次元座標列を平面に投影した例を示す。同図 (a) は任意に設定した平面に投影した結果、同図 (b) は適切な文字平面に投影した結果である。図から明らかなように、適切な文字平面に対して座標列が投影された場合にのみ、文字の形状を正しく認知することができる。

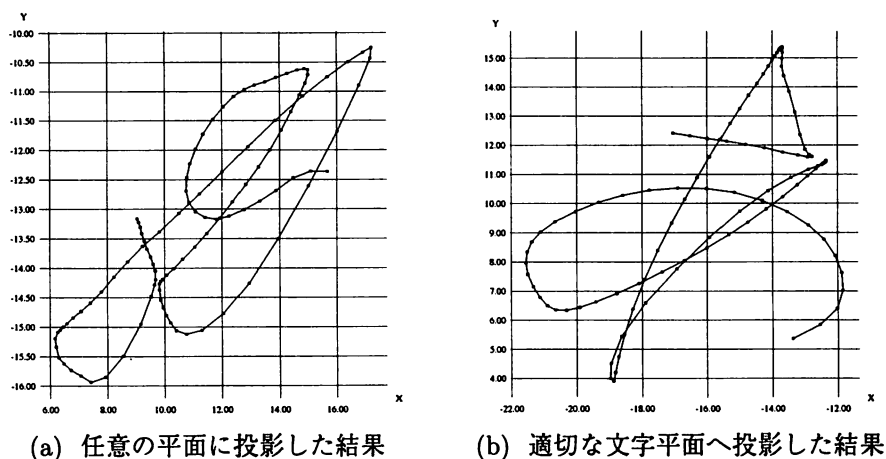


図 2: 文字平面への投影

単に 3 次元の軌跡を文字平面に投影するのみではひと筆書きの字となるので、これを文字の画（以後、ストロークと呼ぶ）ごとに分解できれば、画数などの情報も得られ、より精度の高い文字認識が期待できる。

入力された座標列のうち、文字平面にごく近い座標列が、実際に平面に書かれたストロークを構成し、それ以外の座標列は、ストロークから次のストロークに移動する間に指先を文字平面から離して移動させるための動作部分（以後、運びと呼ぶ）であると考えられる。ストロークと運びを分離するためには、まず最初に文字平面のパラメータが必要となる。しかし、本研究の場合、文字を書く場所は例えば机の上、車の窓などのような任意の平面を前提としているため、文字平面のパラメータは未知であるゆえに、このままではストロークのみの分離は難しい。

以上のような理由のため、正しい文字平面のパラメータを得るために、文字入力に先立って、ユーザに特定の入力操作を行わせることとした。これを予備入力と呼ぶことにし、以下に説明する。

2.3 予備入力による文字平面の特定

文字を入力する前段階として、まず最初に予備入力を行い、文字平面の情報を与えることにする。予備入力では、文字を書こうとする任意の平面から指を離さないように、時計の12時の方向から時計回りに一周する円を描くように指示を与える。そして、予備入力により描かれた円（以下、予備入力円と呼ぶ）の座標列について主成分分析を行うことにより、文字平面の特定を試みる。

予備入力円の座標列は、ほぼ文字平面上に分布している。ゆえに、文字平面は、予備入力円の座標列の重心を通り、予備入力円の座標列を主成分分析して得られる第1固有ベクトルと第2固有ベクトルで張られる平面、つまり第3固有ベクトルを法線に持つ平面にほぼ一致すると考えられる。

なお、予備入力円の座標列が文字平面上から大きく離れた座標列を含んでいると、以上の方法では文字平面が求められない。位置センサは、単に連続的に3次元情報を出力するのみなので、特別に、予備入力円の書き始めと書き終りは、現状ではユーザがキー操作で指示する。

予備入力により文字平面が特定されると、位置センサが遠方から文字平面に接した箇所が書き始めであり、文字平面から十分遠くに遠ざかった箇所が書き終りであると判断できるので、文字を書く軌跡の中から文字の書き始めと書き終りを検出できる。

2.4 文字平面の表裏の決定

第3固有ベクトルは、ストロークを構成する座標点列の分散が最大となる第1-第2固有ベクトル平面に垂直な長さ1のベクトルであるが、方向のみが重要であり、正負には意味がない。つまり、第3固有ベクトルは平面を規定するための法線を表してはいるものの、その符号が平面の正しい表裏を示しているわけではない。そこで、予備入力円が時計回りに描かれていることを利用することにより、表裏の判定を行う。

予備入力円の点列を $\{P_1, P_2, \dots, P_N\}$ としたとき、

$$\begin{cases} \mathbf{a} = \mathbf{P}_{2N/3} - \mathbf{P}_1 \\ \mathbf{b} = \mathbf{P}_{N/3} - \mathbf{P}_1 \end{cases} \quad (1)$$

を求めておき、 $\mathbf{a} \times \mathbf{b}$ の z 成分の符号により平面の表裏を判定する（図3）。

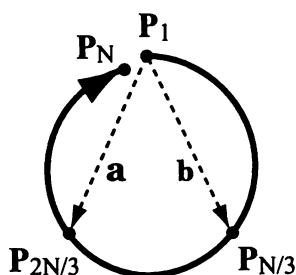


図3: 予備入力円

また、第1、第2固有ベクトルについてもベクトルの向きを補正し、第1固有ベクトルと第2固有ベクトルのそれぞれが文字平面の x 軸、 y 軸になるように、必要に応じて各ベクトルの符号を反転する。最終的に、第1、第2、第3固有ベクトルは、右手系の x, y, z 軸に対応し、このうちの xy 平面が文字平面に対応することになる。

また、予備入力円について、全座標点の重心から始点 P_1 へのベクトルは文字の大まかな向きを表しているため、このベクトルの角度に応じて、これ以降に入力されるパターンを回転変換し、 y 軸の正の方向が文字の上方向を指すようにする。後述するが、今回用いた文字認識手法は文字の回転に不変な特徴量を用いているため、必ずしも文字の回転変換の必要はない。この回転変換はあくまで、入力パターンに大まかな向きの情報を与え、画面表示その他の便宜を図るための処理である。

2.5 ストロークの分離

予備入力により得られた文字平面に投影した3次元座標列は、以前としてひと筆書きの状態である。そこで、運びの部分除去し、ストローク部分のみを残す目的で、ストロークの分離を行う。

文字のストローク部分の座標列は、文字平面にごく近い位置に存在すると予想できる。逆に、運びの部分は文字平面よりもやや離れた位置に分布していると考えられる。そこで、文字平面より上方向（表方向）に Δd だけ離れた、文字平面に平行な平面（以下、分離平面と呼ぶ）を考える。この分離平面よりも下に存在する座標列をストローク部分、上に存在する座標列を運びの部分とみなして、ストロークの分離を行う（図4）。

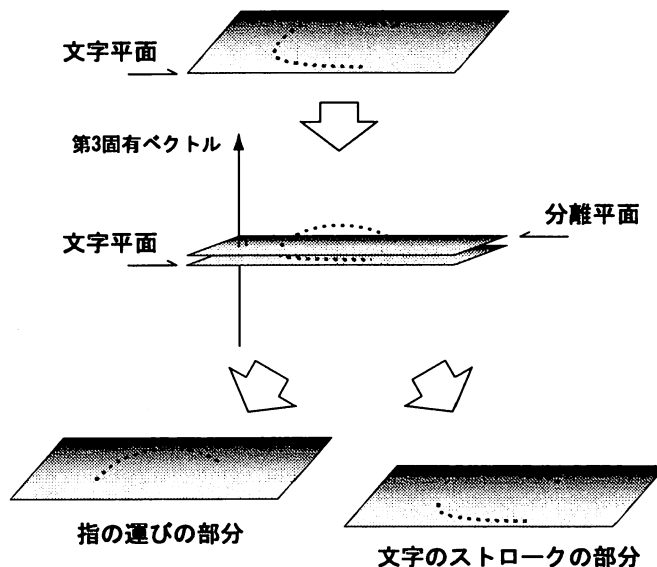


図4: ストロークの分離

2.6 重複点の削除

文字を書く動作において、特に、ストロークの書き始めや曲り角で手が停留した場合、その部分には多くの座標点が集まることになる。そこで、隣接する座標点間の距離を計算し、一定値を下回る距離の座標を重複点として削除する。図5に重複点の削除処理を施した結果例を示す。

2.7 座標列の補間

人間が文字を書く速度は一定ではないので、位置センサにより一定時間ごとにサンプリングされた座標列は当然ながら等間隔ではない。しかし、文字の大きさや書く速度によってストロークごとの座標点の数が異なると、特徴抽出の際に不都合が生じる。そこで、以下の手順によりストローク単位で座標点のリサンプリングを行い、ストロークの構成点数を統一した。

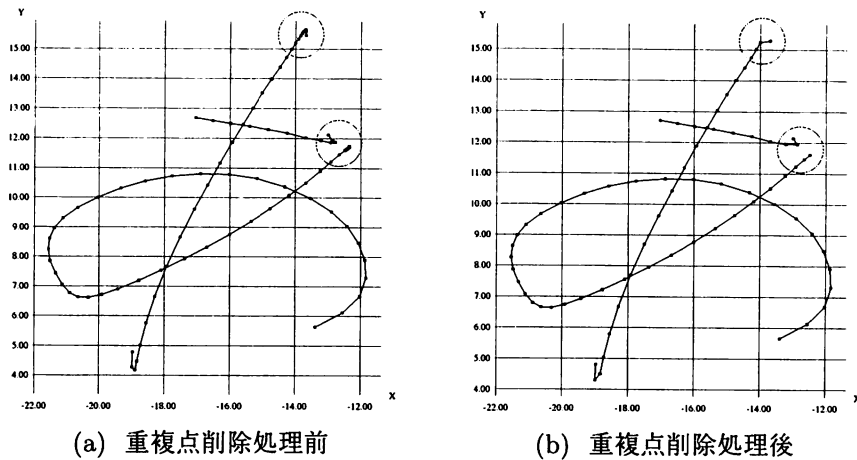


図 5: 重複点の削除例

- (1) ストロークを十分な細かさでスプライン補間し、座標点を追加する。
- (2) 当初より増加した座標点列から、新たにストロークを構成したい点数分、ほぼ等間隔ごとに座標点をサンプリングする。

図 6 に補間処理前の座標列と補間処理後の座標列の例を示す。この例では、各ストロークは 50 点となるように補間されている。

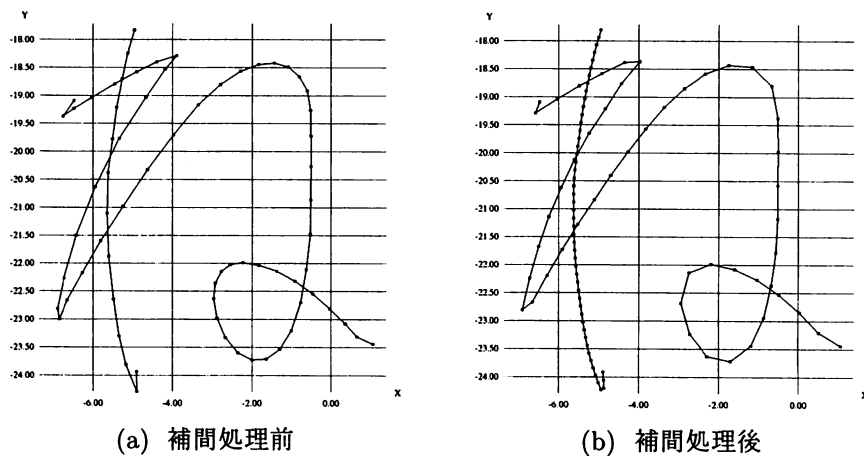


図 6: 補間処理例

2.8 文字種の特定

以上の前処理段階を経た座標列について、文字種を特定する。

2.8.1 画数による文字種の絞り込み

この段階では既に、ストロークの分離処理によって、文字の画数が明らかになっている。そこで、パターンマッチングに先立ち、画数を用いて文字種をある程度絞り込む。

表 1 に、画数による平仮名の分類表を示す。なお、「そ」「ふ」「や」「ゆ」「さ」「き」など、個人の癖によって画数が異なる文字があるが、この表に合致するように書き方を統一してもらうことにする。

表 1: 画数による平仮名の分類

1 画	くしそつてのひへるろん
2 画	いうえこすちとぬねみめゆよらりれわ
3 画	あおかけさせにはまむもやを
4 画	きたなふほ

2.8.2 特徴抽出

位置センサにより入力された文字は、前処理によって座標点数については正規化されるが、傾きや大きさについては正規化がなされていない。そこで、回転とスケールに対して不変な特徴量として、曲率と、文字内のストロークの比率を用いることにする。

曲率は、以下のように曲率を計算した。曲線上の 1 点を P_i ，両側に隣接する点をそれぞれ P_{i-1} , P_{i+1} とすると、これら 3 点より以下の 2 個の単位ベクトルが求まる。

$$\begin{cases} \mathbf{n}_1 = \frac{\mathbf{P}_i - \mathbf{P}_{i-1}}{|\mathbf{P}_i - \mathbf{P}_{i-1}|} \\ \mathbf{n}_2 = \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{|\mathbf{P}_{i+1} - \mathbf{P}_i|} \end{cases} \quad (2)$$

従って、曲率 R は以下のように求められる。式中の sign は計算結果の符号のみを表す。

$$\mathbf{R} = \mathbf{n}_2 - \mathbf{n}_1 \quad (3)$$

$$R = \text{sign}(|\mathbf{n}_2 \times \mathbf{n}_1|) |\mathbf{R}| \quad (4)$$

R は、曲線が進行方向に向かって時計回りに曲がる時には負の値を、反時計回りに曲がる時には正の値を取る。

また、各ストロークは、座標列の補間により、同一の構成点数に正規化されている。つまり、ストロークの長さの情報が失われているため、例えば「い」と「り」、「な」と「ほ」など、文字種によっては曲率の情報のみでは区別が困難となる。そこで、1 文字内に含まれる各ストロークの長さの比率（以下、ストローク比と呼ぶ）を第 2 の特徴量として用いる。

2.8.3 標準パターンの作成

まず、平仮名 46 文字の各ストロークについて、1 ストロークについて求めた曲率のベクトル（曲率ベクトルと呼ぶ）の標準パターンを作成する。本研究は、文字単位ではなくストローク単位のマッチングを行うため、例えば 3 画で構成される平仮名の「あ」には、3 個の標準パターンが必要となる。

平仮名 46 文字を被験者に書いてもらい、ストロークの分離処理により分解した文字ストロークごとの 3 次元座標列を 2 次元座標列に変換したものを、学習サンプルとして保存する。46 文字分の全文字ストローク 104 個の座標列を 1 サンプルとする。各々の文字ストロークの座標列について、前述した各種前処理を施した後に曲率を算出し、学習サンプル数で平均したものを、1 ストローク分の標準パターンとする。

次に、曲率の標準パターンと同様に、学習サンプルよりストローク比を求め、平均することにより、ストローク比の標準パターンを作成する。これについては、文字種と同数の 46 個の標準パターンが得られる。

2.8.4 パターンマッチング

マッチング処理では、あらかじめ作成しておいた曲率の標準パターンと入力文字の曲率ベクトルのユークリッド距離（以下、曲率ユークリッド距離と呼ぶ）を計算し、これを用いて文字認識を行う。

また、前述のように、曲率のマッチングのみでは判別が困難な文字種もあるため、ストローク比の標準パターンと入力文字のストローク比のユークリッド距離（以下、比率ユークリッド距離）も同時に計算して用いる。

ストローク比の特徴は補助的に用いることと、比率ユークリッド距離が1以下の値を示すことから、曲率ユークリッド距離を d_R 、比率ユークリッド距離を d_r （ただし、入力文字と画数が一致している文字に属するストロークのみ）として、以下の式により2種類のユークリッド距離を合成する。そして、最小の d が示す文字を、認識対象文字が属する文字種であると決定する。

$$d = d_R(1 + d_r) \quad (5)$$

3 実験

3.1 実験の概要

実験のために収集したサンプル数は、被験者20人分の20サンプルである。被験者1人につき1サンプル（46文字）を入力してもらう。全サンプルのうち、前半の10サンプルを学習サンプルとして用い、曲率の標準パターン及びストローク比の標準パターンの作成に使用した。後半の10サンプルを未知サンプルとし、認識実験を行なった。

入力に先立って、1度だけ前述の予備入力動作を行わせたのち、46文字の入力を順番に行うように指示する。また、画数と書き順は、予め定められたものに従って入力してもらう。

3.2 実験結果

認識結果を表2に示す。比較のために、パターンマッチングの際に曲率にストローク比を併用した場合としなかった場合についての結果を併記した。なお、正解文字数は、46文字 × 10サンプルの計460文字中の正解数である。

表 2: 認識結果

データの種類	学習		未知	
ストローク比の使用	なし	あり	なし	あり
正解文字数	431	436	388	398
認識率 (%)	93.7	94.8	84.3	86.5

4 考察

実験結果より、未知サンプルについて最高で86.5%の認識率を得た。この認識率が実用においてどの程度の性能であるかは使用目的によって判断が分かれるが、非目視文字入力（文字を書いた筆跡が残らない文字入力）は入力パターンの字形が崩れやすいことや、文字を書く際の場所、傾き、大きさの制約を設けていないことを考慮すれば、本手法の正当性を裏付ける実験結果が得られたものと考えられる。

また、入力パターンの変動に強い特徴量として曲率を用いたが、平仮名は、その多くが曲線で構成されていることから、特徴量として曲率を用いたことは、結果として有用であったといえる。逆に、文字種が平仮名ではない場合、特徴量そのものを見直す必要が生じることも考えられる。

今回の実験では、特徴量として曲率のみを用いる場合に比べて、ストローク比を併用した場合には学習サンプル未知サンプル双方について認識率の向上が見られた。この結果について詳細に調査すると、「い」と「り」の混同や、「た」と「き」の混同が大幅に改善されていることが判明した（表 3）。しかし、当然のことながら、「ん」や「て」などの 1 画の文字は、ストローク比を定義できないために改善が見込めない。曲率とストローク比にいかなる荷重を置いて最終的な特徴量とするかについて検証する必要がある。

表 3: ストローク比による誤認識の改善

誤認識の多い文字種	誤認識された数		改善された数
	比なし	比あり	
い	11	→ 2	9
ん	9	→ 9	0
て	8	→ 8	0
ね	8	→ 8	0
き	6	→ 3	3
ら	6	→ 6	0
わ	6	→ 5	1

5 むすび

本稿では、任意の場所で、適当な平面に指で文字を書く動作を行うことによってコンピュータに文字が入力できるユーザインターフェイスを提案し、位置センサを用いた指で文字を書く動作の解析と、その結果より取得した文字情報による文字認識実験について述べた。実験の結果、学習サンプルで 94.8%、未知サンプルで 86.5% という認識率が得られた。

まず、文字入力に先行して予備入力を行う方法を提案した。これにより、文字平面のパラメータが取得でき、表裏の判定、ストロークの分離、文字の書き始めと書き終りの判定などが可能となり、3次元空間内での文字認識の問題を通常の文字認識の問題に帰着させることができた。

また、入力パターンは、文字が崩れやすい非目視手書き文字であることが前提なので、重複点の削除、ストローク単位のリサンプリングなどの有効な前処理を施したのち、特徴量として曲率を使用することにより、変動に強い認識系を構築した。また、曲率では判定できない文字種については、ストローク比を併用することにより、その誤認識の一部を克服することができた。

現状では、予備入力の方法の改善、あるいは予備入力そのものの省略、特徴量の改良、平仮名以外の文字種に対する有効性の検証などの課題は多いが、柔軟な文字入力インターフェイスのためのアプローチとして今後の発展が期待できるため、さらに詳細な研究を継続する予定である。

参考文献

- [1] 山本吉伸, 梶尾一郎: “空気ペン – 空間への描画による情報共有 –”, 第 59 回情処全大論文集 (4), pp.39–40, 1999.
- [2] 片桐雅二, 杉村利明: “ビデオカメラを用いた個人認証の試み”, 信学技報, PRMU2001-34, 2001.
- [3] 竹下鉄夫, 兼子陽一, 江崎信行: “3次元空間における指書き数字認識”, 信学技報, PRMU2001-76, 2001.
- [4] 古川稔晃, 島田英之, 塩野 充, 宮垣嘉也: “指で文字を書く動作の 3次元解析”, 平成 12 年度電気・情報関連学会中国支部第 51 回連合大会講演論文集, 132505, p.338, 2000.

Three Dimensional Analysis of Writing by Finger Movement

Hideyuki SHIMADA, Toshiaki FURUKAWA*, Yasuhiro SHIMADA,
Mitsuru OHKURA, Mitsuru SHIONO and Yoshiya MIYAGAKI

Department of Information & Computer Engineering,

Faculty of Engineering,

Okayama University of Science,

Ridai-cho 1-1, Okayama 700-0005, Japan

**SGS Co., Ltd.*

(Received November 1, 2001)

Since pen devices have become popular on mobile terminals, handwritten character recognition has been actively researched. Because the user of mobile terminals has to input data in various situations, a more free and easy input method is desired. In this paper, we first perform a three-dimensional analysis of the action of writing a character by finger movement. This is a very natural motion of the hand. Then, we conduct a recognition experiment of handwritten characters on any plane in a three-dimensional space. Using features such as curvature, good recognition rates can be obtained.