

空中での手書き文字入力システム

- ウェアラブルコンピュータ環境での入力インターフェース -

園田 智也 村岡 洋一

早稲田大学 理工学部

〒 169-8555 東京都新宿区大久保 3-4-1

TEL:03-3209-5198

E-mail:{sonoda, muraoka}@muraoka.info.waseda.ac.jp

あらまし 本稿では、装着型ビデオカメラを伴ったウェアラブルコンピュータ環境での入力手法として、利用者が空中で文字を書く動作を行うと、その文字を計算機に入力できるシステムを提案する。本手法では、装着しているビデオカメラによって、利用者の手の動作を撮影し、計算機上で画像解析を行うことで文字入力を実現する。空中で文字を書く際には、その書いた軌跡が残らないため、利用者の文字入力の開始・終了や、文字を構成する線分の始点・終点を特定することが難しい。また、ウェアラブルコンピュータの環境においては、様々な照明条件・背景において、安定して動作検出する手法が必要である。そこで、本研究では (1) 様々な照明条件のもとで、画像フレーム中の利用者の手の領域を抽出する手法、(2) 画像フレーム間の輝度差分の重心位置を手がかりとして、空中での利用者の手の動きを追跡する手法、(3) 連続 DP マッチングにより文字を特定する手法の 3 つを提案する。また、空中での文字認識が根本的に抱える問題を解決するために、英数字の表記法を提案する。本システムをビデオカメラと PC により実装し、空中での手書き文字認識の実験を行ったところ、英数字・記号の 36 文字の認識率は、約 88.3 % であった。

キーワード 手書き入力, ウェアラブルコンピュータ, タッチパネル, DP マッチング, 利用者インターフェース

A Letter Input System of Handwriting Gesture

- A User Interface for Wearable Computers -

Tomonari Sonoda Yoichi Muraoka

School of Science and Engineering, Muraoka Lab. Waseda University

3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, JAPAN

TEL:03-3209-5198. Facsimile:03-3209-5198

E-mail:{sonoda, muraoka}@muraoka.info.waseda.ac.jp

Abstract This paper describes an input system which takes a user's handwriting gesture as a user interface for a wearable computer with a video camera and recognizes its character by using image processing. In tracking a pattern of handwriting gesture, it is quite difficult to determine a users' written letter since the letter is not visible. It is also difficult to recognize a user's hand gestures on the assumption that conditions of lighting often change in a wearable computer environment. We therefore propose three effective methods for recognizing the users' gestures; a method which approximately determines a user's hand region in an image frame under various conditions of lighting, a method which uses the center of gravity of the difference between adjacent image frames for tracking the users hand and a method which uses DP matching for determining the input letter in real time. Moreover, we propose a pattern of handwriting for alphabets and numbers. Testing of our methods against 36 letters of alphabets and numbers showed that its input recognition rate was 88.3%.

Keywords handwriting input, wearable computer, touch panel, DP matching, user interface

1 はじめに

ウェアラブルコンピュータの環境では、装着型ビデオカメラを用いることで、利用者の目の前の光景を撮影し、注釈などの付加情報を重ねた上で、装着しているディスプレイに表示することが可能であり [1, 2]、その付加情報に対する入力インターフェースの研究がなされている。このような環境では、場面に応じた様々な情報を入力できることが望ましい。このため、装着しているビデオカメラによって撮影した画像を解析することで、画像中の人物の顔を認識する手法や、手の動きを用いたマウスの構築手法が提案されている [3, 4]。これらの手法では、目の前の光景に結び付いた入力となってきたが、画像解析によって、計算機の制御や文字入力ができるれば、メールや ATM の暗証番号の入力などが実現し、望ましい。一方、ウェアラブルコンピュータ環境での文字入力を可能にする手法として、ペンの形状をした小型入力装置を用いる手法が提案されているが [5]、利用者が特定の機器を手を持たずに入力できるとより便利である。

そこで、本研究では、利用者が空中で任意の文字を書く動作を行うと、その動作を撮影した画像を解析することで、その計算機に入力できるシステムを提案する。手書き文字認識技術は、計算機上での文字列の検索手法や携帯用情報端末機器などのペン入力手法として普及しているが、空中で描かれる文字は、入力の開始・終了や文字を構成する線分の起点・終点が明確なため、認識することが難しい。また、ウェアラブルコンピュータ環境では、画像を撮影する背景や照明条件が、利用する場所によって大きく変化するため、ジェスチャ認識は容易ではない。特に画像処理において、以下のことが問題となる。

1. 様々な照明・背景の制約を考慮した上で、文字を書く手の領域を検出することが難しい。
2. 利用者の指は、文字入力の最中に様々な方向に傾き、指先の描く軌跡を追跡することが難しい。
3. 実際には見えない文字を、手の動きから特定することは難しい。

本研究では、これらの問題を解決するために、以下の手法を問題提案する。

1. 利用者の手の領域の画素情報を、文字入力の直前に得ることで、手の領域を大まかに抽出する手法。
2. 隣接画像フレーム間の輝度値差分の重心位置によって、指の移動位置を解析する手法。
3. 連続 DP マッチングにより、入力文字を特定する手法。

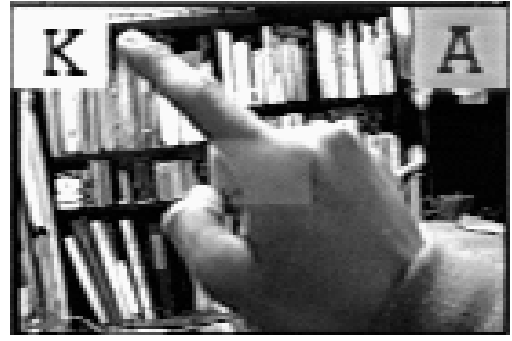


図 1: 画像フレーム内に文字を書く様子

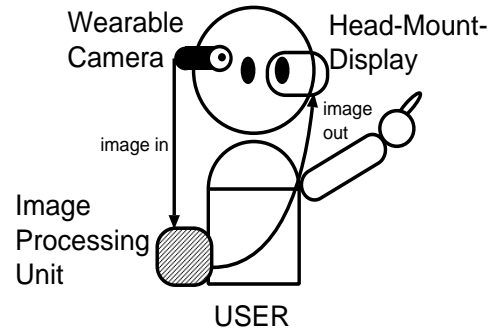


図 2: ウェアラブルコンピュータ環境

さらに、'K' と 'R' のように、空中で書いただけでは判別が難しい文字が存在するため、本研究では、空中で文字入力するための表記法についても提案する。

本研究で提案する手法を、PC とビデオカメラにより実装し、実験を行ったところ、英数字および単純な記号の 36 文字に対する認識率はブロック体の表記法で 61.7% であり、本研究で提案する表記法では、88.3% であった。

2 空中での手書き文字認識システム

以下では、本システムの概要、撮影環境、および空中での手書き文字認識で用いる画像処理の手順を述べる。また、処理上の問題点に対する本研究での解決手法を提案する。

2.1 本システムの仕様

図 1 は、本システムの利用者が空中で文字を書く様子を撮影したものである。本研究では、この利用者の指先の描く軌跡を、図 2 に示すようなウェアラブルコンピュータ環境において解析することで、文字入力を可能とする。本システムでは、利用者は、装着型のビデオカメラ、ディスプレイおよび計算機を身に付けている。ビデオカメラでは、利用者が実際に見ている光景が撮影され、計算機に映像が送られる。ディスプレイには、撮影された映像を重ねて、計算機上で生成された注釈情報等が表示される。

本システムでは、まず、利用者はビデオカメラの画像フレーム内に収まるように、手に何も付けずに人差し指を立て、空中で1つの文字を書く動作を行う。この際、文字を書き終った時点で手を画像フレームの外に出すものとする。次に、撮影された画像フレームは、継続的に計算機上で解析される。システムは、ある時点までの利用者の手が描くパターンが、計算機の記憶装置上にあらかじめ登録されている各文字の動作パターンの中の1つと一致したと判定した場合、その文字が入力されたと見なす。

ここで、利用者の文字の書き順などが、あらかじめ登録している文字のものとは異なる場合は、利用者は新しく自分の描くパターンを登録することができる。また、1つの文字に対して、複数のパターンを登録することが可能であり、その際には最も類似しているパターンが優先となる。

このシステムでは、さらに星型やハート型など、様々な形を描くパターンを記憶できるため、ウェアラブルコンピュータの制御などにも応用できる。

以下では、本システムの撮影条件と画像処理の手順、および処理上の問題点と、それに対する本研究での解決手法について述べる。

2.2 撮影条件

本システムでは、文字入力の際に、背景には移動物体が写っていないことを前提とし、カメラもほとんど揺らさないこととする。この制約は、利用者にとっては、地面を向いて利用する等の動作で容易に実現できるため、大きな負担にはならず、また、システムは精度の高い認識を行うことができる。さらに、背景と指先は肉眼でははっきりと区別できるほど、明るさが違うものとする。利用者の指先とカメラとの距離は約30cm~50cm程度とする。

照明は、特に制約を設けず、自然光、屋内外の灯等で撮影された画像内で、利用者の指の位置が判別できる程度の明るさとする。また、夜間でも赤外線ライトをカメラ側から照射することで、指先の動きを判別できる。

以上の環境で撮影する画像に対して、以下に述べる処理を行うことで、指先の描く軌跡パターンを認識する。

2.3 画像処理上の問題と解決手法

本システムの計算機上での画像処理の手順と問題点、およびその解決手法を述べる。

利用者が空中で書いた文字を認識する上で考慮しなければならないことは、利用者が文字を書き始めた時点と書き終えた時点をかき出して検出し、その間の手の動きをかき出して読み取るかである。

ここで、本システムの仕様では、利用者は、文字を書き終えた時点で画像フレームの外へ手を出すので、その手の存在を検出すれば良いが、前提とする撮影条件では、背景

は完全には静止しておらず、撮影場所も毎回異なるため、様々な背景で安定して動作する設計が必要である。また、照明条件も一定でない。そこで、これらを考慮し、本システムの画像処理は、カメラからの画像フレーム入力があるたびに、以下の手順で行う。

1. 画像フレーム中の利用者の手の領域の検出。
2. 前回の画像フレームとの間での指先の移動量の検出。
3. 現在の時刻までに指先が描いた文字の特定。

ここで、次のことが問題となる。

1. 画像フレーム中の照明や背景の影響を受けずに、手の領域を検出することは難しい。
2. 利用者の指先の角度は、画像フレーム中の文字を書いている場所によって、大きく変化し、前回フレームでの指先の位置と、現在のフレームでの指先の位置の相対的な移動量を正確に求めることは難しい。
3. 書かれた文字は、実際には見えないため、文字を構成する各線分の始点・終点を詳細に求めることは難しい。

1. の問題は、利用者の手の存在が大まかに分かれば良いので、隣接画像フレーム間の差分を検出する手法が考えられるが、カメラが完全に固定されていないため、その誤差を完全に除くことは難しい。そこで、予め肌の色や明るさ(輝度)を登録して、その位置を特定する手法が考えられるが、使用する場所や照明条件ごとに、その値は異なるため、使用時にそれらの情報を特定できることが望ましい。

2. を解決する手法として、隣接画像フレーム間で光学的なフローを検出し、画像フレーム全体のフローと大きく異なる箇所を指先の動きと見なすことが考えられるが、背景が無地である場合、手の動きそのものがフロー検出に影響を与えるため、精度の高い結果が得ることが難しい。

3. の問題は、利用者の文字入力の開始と終了の間の軌跡を手がかりとし、最も類似するパターンの文字を決定すれば良い。しかしながら、'K'と'R'のように、空中で書いた軌跡だけでは、判別が困難な文字も存在する。

以上を考慮し、本研究では各々の問題に対する解決法として、以下の各手法を提案する。

1. 利用者の見ているディスプレイ上に、仮想のタッチパネルのボタンを提示し、利用者がそのボタンを押した際に、利用者の手の領域の情報(輝度値)を取得し抽出する手法。この手法により、取得した輝度値とは大きく異なる画素値を、処理の対象から削除することが可能となり、以下の処理において、精度の高い結果を得ることができる。

2. 隣接画像フレーム間で、同位置の輝度値差分を取り、その差分画像における、値の重心位置 $G(gx, gy)$ によって、利用者の指先の移動位置を決定する手法。この手法により、計算量の多い解析処理を用いずに利用者の手の軌跡を解析できる。
3. 連続 DP マッチングによる文字パターン特定法。本手法では、2. の処理で得た重心位置の 2 つの変量を用いて、高速な入力文字の認識処理を実現する。

以下、3 では、タッチパネル型インターフェースによる利用者の手の領域の抽出法、4 では、輝度差分画像の重心位置による指先の移動位置の決定法、さらに 5 では、連続 DP マッチングを利用した入力文字の特定法を述べ、空中で書いた軌跡だけでは特定できない文字のために、本システム独自の英数字の表記法を提案する。

3 タッチパネル型インターフェースによる利用者の手の領域の抽出法

以下では、手の領域を抽出する際の問題を述べ、提案するタッチパネルインターフェースによる解決法を述べる。

3.1 手の領域の抽出問題

ジェスチャ認識の処理において、画像フレーム内の動きのある領域を抽出することは、システム全体の性能に大きな影響を与えるため重要である。この際、固定されたカメラでは、隣接画像フレーム間の差分を取ることににより、その領域を抽出することが可能である [6]。

しかし、本システム環境では、ほとんど静止した状態で撮影していても、利用者のわずかな揺れにより、背景部分が動き、差分画像において、利用者の手の領域だけを優先的に抽出することは難しい。この際、光学的なフローを隣接画像フレーム間で求め、その影響を小さくすることは可能であるが、背景に特徴点が無い場合、手の動きそのものが光学的フローに大きな影響を及ぼすため容易ではない。

そこで、本研究では、文字を入力する直前に、利用者の手の明るさ(輝度値)を取得し、それに近い値を持つ画素のみを処理の対象とすることで、手の領域を大まかに抽出する手法を提案する。この際、輝度値を確実に取得するために、ある瞬間に、画像フレーム内の絶対的な位置に手の領域が重なっていることが望ましい。そこで、本研究では、利用者が任意の瞬間に手の領域の輝度値を取得する手法として、利用者の装着しているディスプレイ上にタッチパネルを表示し、そのボタンを指先の動きによって押すことができるインターフェースを提案する。利用者は文字の入力の直前にタッチパネルのボタンを軽く触れるように手をかざすことで、容易に輝度値を取得できる。

なお、現行のシステムは、白黒階調画像を用いて処理しているため、本手法を輝度値の獲得に用いているが、予備実験から、肌の色情報(色相値)を用いることができる環境では、さらに精度の高い処理が行える。

以下では、本研究で提案するタッチパネル型のインターフェースについて述べる。

3.2 タッチパネル型インターフェース

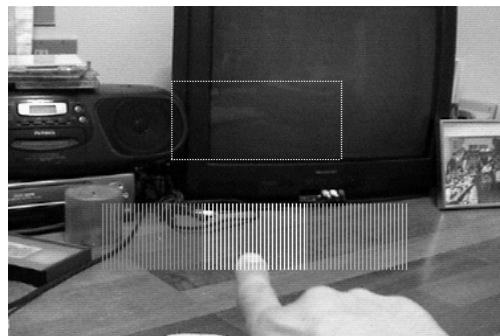


図 3: タッチパネル型インターフェースの例

図 3 に、本研究で提案するタッチパネル型インターフェースの例を示す。この手法では、利用者は、ディスプレイ上に示されたボタンを押すように、指先をボタンに重なるように移動させることで入力が行える。

本インターフェースの構築の際には、利用者がボタンを押す際に変化する画像中の輝度に着目する。具体的には、隣接フレーム間での輝度値の変化量の和をボタン領域ごとに求め、ある閾値以上の変化量を持つボタン領域には、利用者の手が重なっていると見なして処理を行う。すなわち、時刻 t における画像フレーム F_t 上の各画素の輝度値を $I(t, x, y)$ とし、タッチパネル上の N 個のボタン領域を B_n ($n = 0, 1, \dots, N-1$) としたときに、 B_n 上での輝度の変化量 S_n を次の式で求める。

$$S_n = \sum_x \sum_y |I(t, i, j) - I(t-1, i, j)|, \\ I(t, i, j), I(t-1, i, j) \in B_n$$

ここで、 $S_m = \max\{S_n | 0 \leq n < N\}$ が、

$$S_m > T * S_n \quad (T \text{ は定数. } 0 \leq n < N-1 \text{ かつ } m \neq n.)$$

を満たすとき、 B_n が押されたと判定する。

ここで、本研究の予備実験において、次のような誤入力が見られた。

- あるボタンを 1 回だけ押したい場合にも、連続して入力される現象。
- 背景画像の影響で、意図していないボタンが自動的に押されてしまう現象。

1. の現象は、入力操作の履歴を、時間軸によって管理することで、連続する入力を容易に制限できる。一方、2. の現象は、本システムの撮影条件のように、背景に移動物体が写っておらず、カメラ自身にも大きな揺れがない場合は、ほとんど発生しない。また、ボタンを複数組み合わせることにより、誤入力を低下させることが可能である。

本システムでは、利用者がボタンを一度押した後は、意図しないボタン入力が発生しないように、利用者の入力がない状態が一定時間続かない限り、再度ボタンを押すことができないようにしている。

本システムでは、このインターフェースを用いることで、任意の瞬間に利用者の手の領域の輝度値を取得できる。

3.3 手の領域の輝度値の取得方法

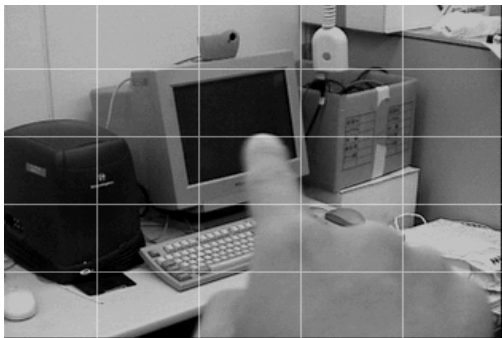


図 4: 輝度値の取得用ボタン

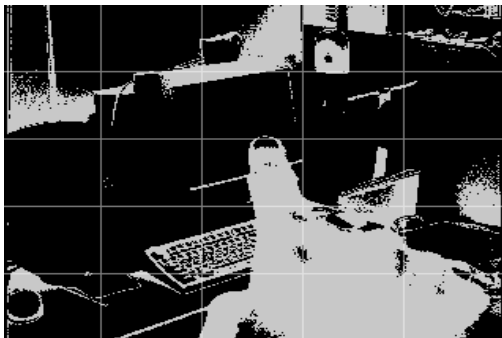


図 5: 手の領域の輝度値による 2 値化画像

図 4 は、画像フレームを縦横に 5*5 の領域に分割したものである。本システムでは、このうちの中央の領域を、手の領域の輝度値を取得するためのものとして利用しており、他の 24 個の分割領域をその他のボタンと見なして、利用者がボタンを押しているかどうかの判定を行う。

ここで、本システムでは、利用者が中央のボタンを押した瞬間に、画像フレーム中の特定の領域 R に手の領域が重なっていることを前提とする。本システムでは、 R の位置は利用者が右利きの場合、25 分割した一番右下の領域としており、左利きの場合一番左下としている。ボタンが押された際には、システムは R 上の画素値に関する

ヒストグラムを作成し、最も度数の大きな画素値 v を求める。その後、現在の画像フレームにおいて、 $v \pm \epsilon$ を満たす値を持つ画素のみを残し、残りの画素をすべて処理の対象外と見なす。図 5 は、図 4 のボタンを押した結果得られた v に対して、 $v \pm \epsilon$ ($\epsilon = 10$) を満たす画素とそうでない画素を 2 値化して表示したものである。 $v \pm \epsilon$ を満たすものを白く表示している。

ここで、本システムでは $v \pm \epsilon$ で残した画素について、隣接画像フレーム間で差分をとり、その総和がある閾値 $Hand$ を越える場合には、利用者の手が画像フレーム内にあり、文字を書いている最中だとみなし、そうでない場合には、文字入力を行っていないと見なす。

以下では、このようにして得られた画像を用いて、指の描く軌跡を特定する。

4 輝度差分画像の重心位置による指先の移動位置の決定法

本手法では、利用者の指の動きを追跡する手法として、隣接画像フレーム間の輝度値差分の絶対値を重みとした、画素値の重心位置を用いる手法を提案する。すなわち、重心位置の遷移パターンによって、入力された文字を特定する。

ここで、時刻 $t, t-1$ における画像フレーム F_t, F_{t-1} 間の各画素の輝度値の差分 $|I(t, x, y) - I(t-1, x, y)|$ を重みとしたとき、その重心の位置 $G(gx, gy)$ は、次の一連の式で求められる。

$$\begin{aligned}
 X &= \sum_i \sum_j i * |I(t, i, j) - I(t-1, i, j)| \\
 Y &= \sum_i \sum_j j * |I(t, i, j) - I(t-1, i, j)| \\
 W &= \sum_i \sum_j |I(t, i, j) - I(t-1, i, j)|
 \end{aligned}$$

これより、

$$\begin{aligned}
 gx &= X/W \\
 gy &= Y/W
 \end{aligned}$$

本システムでは $I(t, i, j)$ として、上述の処理で得られた、手の領域の輝度値に近いものは、その値を用いており、それ以外では値を 0 としている。

図 6 は、以上の式により求めた、アルファベットの 'B' を描いた際の重心位置の軌跡を示している。この図が示すように、重心位置 $G(gx, gy)$ の軌跡は、利用者の手の動きに伴っている。

以上で得られる G の遷移パターンは、次に述べる連続 DP マッチングにより、DB のパターンと照合する。

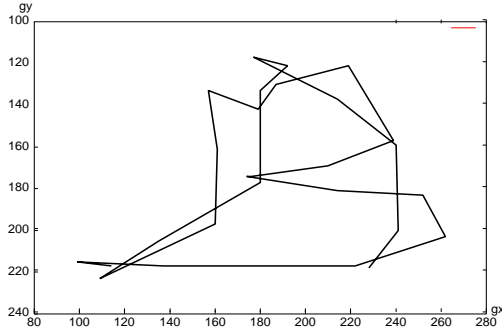


図 6: 'B' を書いた際の重心 $G(gx, gy)$ の軌跡

5 連続 DP マッチングを利用した入力文字の特定法

以下では、利用者に描かれる文字を、連続 DP マッチングにより認識する手法を述べる。

5.1 連続 DP マッチングによる距離の測定

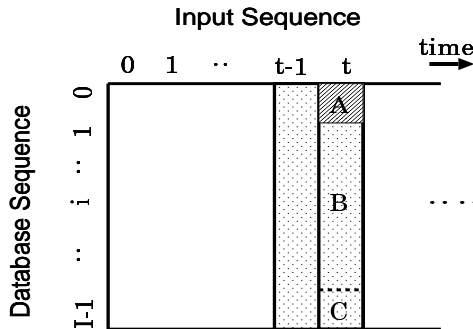


図 7: 連続 DP マッチング

本手法では、空中で描かれた文字を認識する際、画像フレーム間の輝度差分の重心位置の遷移パターンを、入力とデータベース (DB) との間で照合する。この照合には、ジェスチャ認識等でよく用いられる連続 DP マッチングを用いる [6, 7]。同様の手法として、HMM(hidden Markov model) を用いることも考えられるが、本システムでは学習させるデータが音声認識処理などと比較して少ないため、処理がより単純な DP マッチングを用いている。

ここで、利用者の入力における重心位置の系列を x, y 方向でそれぞれ Q_x, Q_y とし、同様に、長さ I の DB 系列における重心位置の系列を S_x, S_y とする。それぞれの系列の要素が取り得る整数値の集合を C_x, C_y としたとき、入力と DB との要素間の距離 d_x, d_y を次のように定義する。

$$\begin{aligned} Q_n &= q_0, q_1, q_2, \dots, q_t \in C_n \\ S_n &= s_0, s_1, s_2, \dots, s_{I-1}, s_i \in C_n \\ d_n(q_t, s_i) &= |q_t - s_i| \\ &\text{(ただし } n = x, y) \end{aligned}$$

連続 DP マッチングの計算は、図 7 のように 2 次元のマッチング行列 $m(i, t)$ を用いて、時刻 t までの入力の重心位置の系列と DB 系列との間の距離 D を求める。はじめに $t = 0$ における距離の計算を行い、次に時刻 t における入力パターンと DB の先頭パターンの距離の計算を行う (Step A)。続いて、時刻 $t-1$ の結果を利用して、その時刻における入力パターンと DB 中のすべての位置のパターンとの距離を計算し (Step B)、最後の行で得られた結果を、時刻 t における入力系列 Q と DB 系列 S との間の距離 D とする (Step C)。この処理を各 DB 中の系列ごとに行い、時刻 t までの入力系列とすべての DB 系列間の距離を求める。

以下に、各 Step A ~ C における要素の計算を示す。なお、 $t = 0$ における $m(i, 0)$ の値は、すべて $m(i, 0) = \infty$ ($0 \leq i < I$) とする。

- Step A の計算

$i = 0, t > 0$ において、

$$m(0, t) = d_x(s_0, q_t) + d_y(s_0, q_t)$$

- Step B の計算

$0 < i < I, t > 0$ において、

$$\begin{aligned} m(i, t) &= d_x(s_i, q_t) + d_y(s_i, q_t) + \\ &\min(m(i-1, t), \\ &\quad m(i-1, t-1), \\ &\quad m(i, t-1)) \end{aligned}$$

- Step C の計算

$$D(Q, S) = m(I-1, t)$$

以上で定義した連続 DP マッチングの距離 D は、時刻 $t-i \sim t$ の間の利用者の入力系列 Q が S の系列に類似するほど、小さな値を取り、一致する場合、0 となる。

5.2 入力文字の決定

DB 中の Num 個の登録データに対する、時刻 t での連続 DP マッチングの距離 $D(Q_t, S_k)$ ($k = 0, 1, \dots, Num-1$) の中から、実際に入力された文字を決定する手順を述べる。

図 8 は、本研究で利用した DB 中に登録している英字 'A' ~ 'C' に対する、時刻 t (フレーム) での D の距離の遷

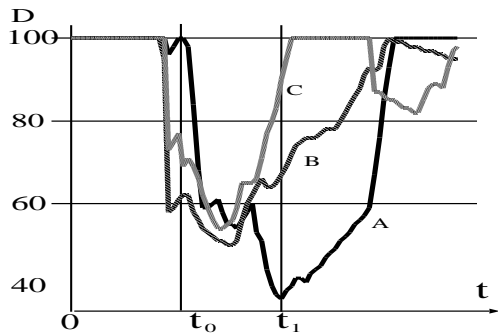


図 8: 'A'~'C' に対し、'A' を入力した際の距離 D

移を示しており、時刻 $t_0 \sim t_1$ の間で、'A' の文字を入力している。この図に観察できるように、 $t_0 \sim t_1$ では 'A' に対する距離 D が他の文字の距離よりも小さくなっている。特に入力動作の終る t_1 の直前の時点では急激に小さくなり、しばらく最小値を維持しながら、再び増加していく。

本システムでは、この性質を利用し、利用者の入力終了している (画像フレーム外にある) という前提で、以下の条件をすべて満たす $D(Q_t, S_i)$ ($0 \leq i < Num$) が観測された時刻 t で、 S_i の系列が入力されたと判定する。

● 条件 1

$0 \leq t_x < Time$ において、 S_i が

$$D(Q_{t-t_x}, S_i) = \min\{D(Q_{t-t_x}, S_k) \mid 0 \leq k < Num\}$$

かつ

$$D(Q_{t-t_x}, S_i) \geq D(Q_{t-t_x-1}, S_i) \text{ をすべて満たす。}$$

● 条件 2

$$D(Q_t, S_i) < Distance \quad (Distance \text{ は定数})$$

本研究では各条件で用いる $Time, Distance$ を予備実験に基づいて、経験的に決定している。条件 1 は、時刻 t において、最小の D 値を持つパターンを有力候補と見なししており、利用者の文字入力が終わった際に、該当した文字パターンの D がある一定の時間 ($Time$)、最小値を保ちながら増加する傾向にあることを意味している。また、条件 2 では、最小値がある一定の閾値 ($Distance$) より小さい値を取ることを意味している。

5.3 空中での手書き文字表記法

上述した手法により、ブロック体の英数字を入力する予備実験を行ったところ、(B, R, K, P), (C, L), (4, 9) などの文字間で誤認識が頻繁に発生した。この中には 'R' と 'K' や '4' と '9' など実際に人間が判別が困難なものもある。

この問題は、通常書かれる文字が、紙の上などで軌跡を残すことが前提となっており、画像だけでは、その軌跡を

得ることができないことが原因である。これは、学習アルゴリズムなどで、利用者の癖をより精密に判定させるだけでは、解決することが難しい。

そこで、本研究では、空中での手書き文字認識が根本的に抱えるこの問題を解決するために、空中で手書き文字入力をするための英数字の表記法を提案する。ここで、文字の設計に際し、予備実験から以下の指針を得た。

- 画像フレーム内の手の動きは、必ず一筆書きになるため、一筆書きの文字を設計した方が利用者の誤入力を抑えることができ、望ましい。
- 描かれる文字の軌跡は、輝度差分の重心によって求められるため、手を画像フレームの下方向から出す場合は、縦方向よりも横方向の方の変化量が多い。このため、文字は横方向に多くの変化量を持つように設計した方が認識率が高い。
- 利用者が文字を書き終えた時点をシステムに認識させ、また、'P' と 'R' などの文字間の誤認識を抑えるために、文字の最後を画像フレームの外に出るように設計した方が良い。

以上の指針に基づいた上で、本研究では、多くの利用者に普及している Palm 社の Graffiti を参考にし、空中で手書きするための文字を設計した (図 9)。この際、できるだけ Graffiti に従い、'U' と 'V', 'P' と 'R' などの誤認識が起きやすい文字に関しては、最後にフレーム外に手を出す方向を定めることで、認識率を高める工夫をした。

利用者は、中央のタッチパネルボタンを押した後、各文字の始点まで指先を運び、文字を書いた後、画像フレーム外に手を出すことを想定している。なお、図 9 の各文字の周りの枠は、画像フレームを表している。

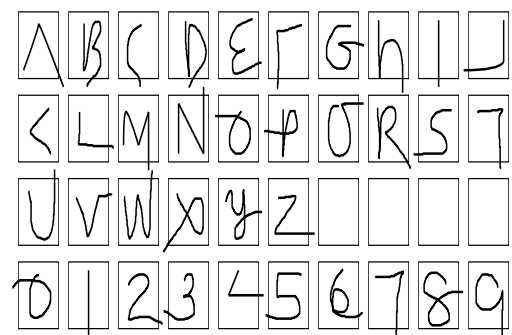


図 9: 空中での手書き文字のための英数字の表記法

6 実験

本稿で提案したシステムを実装し、一連の評価実験を行った。以下では、実験環境と実験内容を述べる。

6.1 実験環境

本実験では、2.2 で述べた撮影条件 (室内、蛍光灯の照明) のもとで、デジタルビデオカメラ (SONY:PC-100) を利用者の顔周辺に持って撮影した。映像は、320*240 のフレームサイズで白黒 256 階調画像で記録し、デスクトップ型 PC (CPU AMD: Athlon 500MHz, RAM:128M, OS: LINUX, Graphic Board: BT878) 上で処理した。実験中の画像の背景は、図 1 のように室内の本棚で、撮影中のぶれは、最大で上下に 12 ピクセル、左右に 9 ピクセルであった。

文字入力時、被験者 (1 人: 右利き) は、カメラの右下側から手を伸ばし、タッチパネルボタン (押下判定の閾値 $T = 5$) を押して入力を開始した。また、入力文字は、英数字のうちのいずれかを入力することを前提とし、予め被験者自身の手書き文字で、各文字について、ブロック体の文字 3 個、および 5.3 で提案した表記法の文字 3 個ずつを別々のデータベースに登録した。さらに、'I' と 'l' や 'O(オー)' と '0(ゼロ)' などの類似した表記による誤りを防ぐため、入力の際には英字と数字は切替えて行った。

手の領域を抽出する際の画素の値の範囲 ϵ は 5 とし、手が画像フレーム内にあるかどうかの閾値 $Hand$ を経験的に 1000 とした。また、文字判定の際の連続 DP マッチングでの距離の最大値は、100 となるように設定し、文字の決定処理の際には $Distance = 50, Time = 5$ とした。

6.2 スループットと遅延

本システムの実時間性を検証するために、画像処理のスループットと、実際の画像入力から表示までの遅延時間を 10 分間計測した。その結果、画像表示のスループットは、平均 9.9 fps であり、処理の遅延時間は、平均 1.2 フレーム (約 40 msec) であった。出力される画像フレームを肉眼で見ても、利用者の入力した文字が判別でき、この性能は文字入力を行う上で特に問題とならないことが確認できた。

6.3 文字認識率

本システムの性能を評価するために、ブロック体記法と 5.3 で提案した表記法の英数字 36 文字の各々に対して、10 回ずつ入力を行い、意図する文字が正しく入力された回数によって、認識率を計測した。表 1 に結果を示す。

この表に示されるように、ブロック体は総合で、61.6 % であるのに対し、本稿での表記法では 88.3 % と高い認識が得られた。実験中、ブロック体表記の数字では、'0' と '6' や '7' と 'l' などに、誤認識が多く見られた。また、本実験で文字を登録した人物以外の被験者に同様の実験を行ったところ、提案する文字表記法において、数字だけの認識率は、100 回中 79 回 (79.0 %) の結果より、本システムは、各個人が文字を登録するか、不特定利用者に適用するために学習アルゴリズムが必要となると考察できる。

表 1: 空中での手書き文字認識率

	成功回数/入力回数	認識率
英字 (ブロック体表記)	168/260	64.6 %
数字 (ブロック体表記)	54/100	54.0 %
英数字総合 (ブロック体表記)	222/360	61.6 %
英字 (提案した表記)	229/260	88.1 %
数字 (提案した表記)	89/100	89.0 %
英数字総合 (提案した表記)	318/360	88.3 %

7 まとめと今後の課題

本稿では、空中での手書き文字入力システムを提案した。本研究では、タッチパネル型のインターフェースにより、利用者の手の領域を特定し、輝度差分の重心位置で指先の軌跡を追跡し、連続 DP マッチングによる文字列を特定するという一連の手法を提案した。また、独自の英数字の表記法により、通常の表記よりも認識精度の高い入力インターフェースを構築できた。

今後は、英数字以外の日本語や様々な記号や不特定利用者認識についても研究を行いたい。また、実際に装着可能なコンピュータ環境を構築し、実験を進める予定である。

参考文献

- [1] Tony Jebara, Cyrus Eyster, Josh Weaver, Thad Starner and Alex Pentland. "Stochastics: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers", Proceedings of the International Symposium on Wearable Computers, October 1997.
- [2] 興梠 正克, 蔵田 武志, 坂上 勝彦, 村岡 洋一: パノラマ画像群を位置合わせに用いた注釈つき映像の実時間提示システム, 信学技報, PRMU-99-64, pp.1-8, September 1999.
- [3] 蔵田 武志, 興梠 正克, 栗田 多喜夫, 村岡 洋一, 坂上 勝彦: 気の利いたウェアラブルビジョンシステムの実現に向けて, 信学技報, PRMU-99-270, pp.65-71, March 2000.
- [4] 蔵田 武志, 坂上 勝彦, 興梠 正克, 村岡 洋一, : 気が利いたウェアラブルビジョンシステムの実現に向けて - 状況把握に有効な要素機能の実時間実装, 画像ラボ, 2000.6.
- [5] 山本 吉伸, 椎尾 一郎: 空気ペン - 空間への描画による情報共有 -, 情報処理学会第 59 回全国大会 3ZA-3, 1999.
- [6] 向井 理朗, 高橋 裕信, 遠藤 隆, 中沢 正幸, 松村 博, 岡 隆一: マルチモーダルヒューマンインタフェースのノートパソコンへの実装, 信学技報, PRMU-98-70, pp.69-75, May 1998.
- [7] 岡 隆一, 西村 裕一, 向井 理朗: しぐさで伝える, 電子情報通信学会誌, Vol.82, No.4, pp.332-339, 1999.
- [8] Palm Inc., <http://www.palm.com/>.