

## 空中手書き文字入力システムの 構築に関する一考察

藤井祐介<sup>†</sup> 竹沢恵<sup>†</sup> 真田博文<sup>†</sup> 渡辺一央<sup>†</sup>

従来の手書き文字入力は、ペンやスタイラスを用いて、タッチパネルやタブレットなどの決まったエリアに筆跡を見ながら丁寧に書くことが前提であった。しかしこれでは、「いつでもどこでも」といったウェアラブル環境での文字入力に適しているとはいえない。そこで我々は、空中に書いた文字を認識する空中手書き文字入力方式に着目した。その中でもペン型の入力デバイスを用いて空中に書いた文字を認識する手法に注目し、そのシステムの試作を行った。本稿では濁点、半濁点を除いた全てのひらがなに対して本システムにより認識を行った結果を報告する。

### An Aerial Handwritten Character Input System

Yusuke Fujii<sup>†</sup> Megumi Takezawa<sup>†</sup>  
Hirofumi Sanada<sup>†</sup> Kazuhisa Watanabe<sup>†</sup>

Recent advances in computer technology have enabled the development of small mobile computers such as PDAs (Personal Digital Assistants) and palmtop computers. It is expected that various wearable computers will become common in the near future. Some character input systems for wearable computers have been proposed. However, most of them are not convenient because the user must carry several devices and careful writing is needed. Hence, a new character-input system that is more convenient is required. Therefore, in this paper, a new character input system designed for use in a wearable computing environment is proposed. The system recognizes characters handwritten in air. It requires only a device like a pen and users can write characters without looking at them carefully. This approach is suitable for a wearable computing environment. Results of simulations that verify the performance of the system are also presented in this paper.

### 1. はじめに

近年のコンピュータ技術の進化により、PDA (Personal Digital Assistant) やパームトップコンピュータといった小型で軽量のコンピュータが開発され、コンピュータを持ち運ぶことが可能となった。そして、これらのモバイルコンピュータの普及に伴い、メールや文書作成といったアプリケーションでの文字入力的手段として、従来のキーボードに換わり、音声入力[1]やタッチパネルでのペン入力などが用いられるようになった。しかし、これらの文字入力の方法は、近い将来広く普及するであろう次世代のウェアラブルコンピュータには必ずしも適しているとはいえない。例えば、音声入力では、周囲に人がいる状況で気軽に文字入力を行うことができない。また、従来のペン入力は、ペンやスタイラスなどのデバイスと、タッチパネルやタブレットなどの入力領域をもったインターフェースが必要となることから、携帯性に優れているとはいえない。さらに、「決まったエリアに」「筆跡を見ながら丁寧に」書く必要があり、いつでもどこでも気軽に文字入力をしたいウェアラブル環境に適しているとはいえない。

そこで我々は、ウェアラブル環境に適した文字入力の方法として、タッチパネルやタブレットにではなく、空中に書いた文字を認識する空中手書き文字入力方式[2][3][4]に着目をした。その中でも、ペン型のデバイスを用いて空中に書いた文字を認識する手法に注目し、この手法による文字認識システムを試作した。ペン型のデバイスを用いた空中手書き文字入力方式は、文字を書く場所を特定しないため、どのような場所でも文字を書くことができる。また、操作方法も容易であり、ペン型の入力デバイスのみで文字入力を行える。さらに、音声入力とは異なり、周囲に人がいる状況でも気軽に文字を書くことが可能であり、ウェアラブル環境に適しているといえる。

本稿では、試作した文字認識システムを用いて、濁点、半濁点を除いた全てのひらがなに対する認識実験を行った結果を報告する。また、認識精度向上のために辞書学習を取り入れ、その効果について考察を行った結果を報告する。

以下、2. で空中手書き文字入力システムの概要とその構成について述べ、3. で試作したシステムを用いて実験を行った結果を示す。そして、4. でまとめを述べる。

### 2. システムの概要

ペンやスタイラスを用いて、タッチパネルやタブレットなどに文字を書く従来のオンライン手書き文字入力方式では、筆記の位置情報、方向成分情報、長さなどを用いて文字を認識している。これに対し、空中に書いた文字を認識する空中手書き文字入力方式では、筆記の位置情報や長さを特定することができない。また、ユーザは筆跡を確認しながら書くことができないため、書いた軌跡が正確な文字の形になっている

<sup>†</sup> 北海道工業大学  
Hokkaido Institute of Technology

とは限らない.そこで本システムでは,筆記の方向成分情報のみから文字を認識する.  
 今回我々は Web カメラをペンに取り付けてデバイスを試作し,この試作デバイスで空中に文字を書き,その動画像から方向成分情報を得て文字を読み取っている.筆記の方向成分情報を得る方法には加速度センサや角速度センサを用いる方法もあるが,扱い易さから Web カメラを用いている.なお,今回用いた Web カメラから得られる動画像は,320×240画素の24ビットカラー画像であり,フレームレートは15fpsである.

### 2.1 処理手順

図 1. にシステムの処理手順を示す. 主な処理は 3 つであり,動きベクトル検出,コード化,マッチングからなる. 各処理について以下に詳しく説明する.

#### 2.1.1 動きベクトル検出

本システムでは, Web カメラを用いて得られる動画像から局所相関演算[5]により動きベクトルを求める. 局所相関演算とは,二つの異なるフレームの局所画像間で下式の SAD 演算を行い,残差が最小となる位置から動きベクトルを得る手法である.

$$D^f(u, v) = \sum_{x, y \in R} |I^f(x, y) - I^{f+1}(x + u, y + v)| \quad (1)$$

ここで,式中の $I^f(x, y)$ は, $f$ フレーム目の画像中の座標 $(x, y)$ の輝度値, $R$ は $f$ フレーム目での参照領域, $D^f(u, v)$ は動きベクトル $(u, v)$ に対応する残差(相関値)を表している. 1画素以下の動きベクトル検出精度を必要とする場合,累積加算する画素 $(x, y)$ (以下,代表点と呼ぶ)の数を100点以上に設定する必要があるが,今回は8方向の判別を行うのみであるため,入力画像を25個のブロック(ブロックの大きさは $64 \times 48$ 画素)に分割し,各ブロックの中央画素を代表点とした. よって, $m$ 番目のブロックの代表点の座標を $(x_m, y_m)$ としたとき,動きベクトル $(u, v)$ に対応する相関値 $D^f(u, v)$ は次式により算出される.

$$D^f(u, v) = \sum_{m=1}^{25} |I^f(x_m, y_m) - I^{f+1}(x_m + u, y_m + v)| \quad (2)$$

#### 2.1.2 コード化

検出された動きベクトルから得られる動き情報を,上下左右とその間の斜めを含む8方向,そして動きが少ない不定の9通りに分類し,コード化する. 各方向に割り当てたコードを図 2. に示す. 動画像の各フレームから得られる動きベクトルに対してコード化を行い,それらのコードを連結したものを入力コード列と呼ぶこととする. なお,本システムでは,空中に文字を書く際の手のブレや,筆記速度の変化などの影響を軽減するため,以下の手順で入力コード列を編集している.

- (i). 前後に同じコードがある場合,そのコードを前後のコードで置き換える.

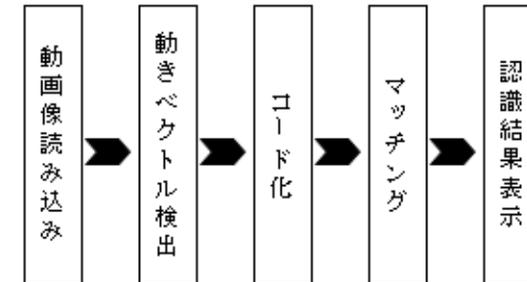


図 1. システムの処理手順

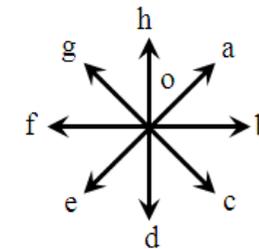


図 2. 9通りの方向情報

- (ii). 連続するコードを一つに纏める.

- (iii). 不定コード(o)を削除する.

#### 2.1.3 マッチング

入力された動画像から得られる入力コード列と各文字の辞書コード列を比較してコード間の距離を算出し,入力した文字の判定を行う. なお,本システムでは,距離の算出に DP マッチングを用いる.

### 2.2 辞書

辞書には,正解となる辞書コード列を一文字につき一つ登録する. 表 1. にあらかじめ登録した辞書コード列の一部を示す. しかし,ユーザによっては文字に癖のある書き方をすることがあり,この場合は入力コード列が辞書コード列と類似せず,文字が正しく認識されない可能性がある. そこで我々は,ユーザの癖を辞書に反映させるため,書いた文字が正しく認識されなかった場合は,その入力コード列を辞書に追加する逐次学習法[6]を取り入れ,認識精度の向上を図った.

表 1. 辞書コード列の例

文字	辞書コード	文字	辞書コード
あ	bdcdefghabcde	か	bcdegdec
い	dacd	き	bfdc
う	cdcbe	く	ec
え	cbeacb	け	dbde
お	bdefghabcdec	こ	bcedcb

表 2. 学習を行わない場合の認識率

	一画	二画	三画	四画	平均
筆記者 A	85.0%	80.9%	77.3%	87.0%	82.6%
筆記者 B	92.3%	75.6%	69.6%	69.0%	76.6%
筆記者 C	90.5%	67.6%	71.2%	63.0%	73.1%
筆記者 D	79.5%	56.8%	64.6%	73.0%	68.4%
筆記者 E	81.8%	69.5%	59.6%	75.0%	70.8%
平均	85.8%	69.5%	68.5%	73.4%	74.3%

### 3. 実験

本システムの性能を評価するため、5人の筆記者にひらがな46文字を20回ずつ書いてもらい、認識実験を行った。

なお、試作したシステムでは、筆を降ろして筆記中であるか、それとも単に筆を上げて移動中であるかを判断できないため、文字を書く際一筆ごとに動画像を分けて保存している。

#### 3.1 文字認識率

表 2. および表 3. に文字認識率を算出した結果を示す。ここで、表 2. は辞書学習がない場合の結果であり、表 3. は逐次学習法による辞書学習を取り入れた場合の結果である。表 2. から、辞書学習がない場合、筆記者によって認識率に差があることが分かる。これは、あらかじめ登録した辞書コード列が個人の特徴と必ずしも合っていないことが原因であると考えられる。

一方、表 3. から、辞書学習を取り入れた場合、筆記者5人の認識率の平均は87.2%であり、学習を行わない場合に比べて約13%認識率が上がっていることがわかる。この結果より、辞書学習を行うことで個人の書き方の癖を反映した辞書を生成し、誤認識を防ぐことができたと考えられる。

#### 3.2 逐次学習法の問題点

実験により、逐次学習法を取り入れることで、文字認識率の向上を確認することができた。しかしながら、逐次学習法には解決すべき問題が存在する。一つは、逐次学習法により辞書に追加登録された文字Aの入力コード列が、別の文字Bの辞書コード列と類似していた場合、文字Bの入力を文字Aと誤認識してしまう可能性があることである。もう一つは、辞書容量の増加により、マッチングを行う回数が増えるため、学習を行うごとに処理時間が増加してしまうことである。そこで本稿では、後者の問題に対し、辞書に追加可能な入力コード列の数を最適な値に設定することで解決を試みる。

辞書に追加可能な数Lを0~6、および∞と変化させ、文字認識率と認識に要する処

表 3. 学習を行った場合の認識率

	一画	二画	三画	四画	平均
筆記者 A	91.8%	92.1%	89.6%	96.0%	92.4%
筆記者 B	96.4%	87.6%	89.6%	90.0%	90.9%
筆記者 C	91.4%	87.6%	88.1%	90.0%	89.3%
筆記者 D	86.8%	80.0%	79.6%	73.0%	79.9%
筆記者 E	85.9%	83.8%	80.4%	84.0%	83.5%
平均	90.5%	86.2%	85.5%	86.6%	87.2%

理時間を調べた結果を表 4. に示す。なお、L=0 では学習を行わず、L=∞では無制限に入力コード列を辞書に追加することを意味している。また、各筆記者の平均認識率は一画から四画までの全ての文字に対する認識率の平均を表しており、平均処理時間は1文字の認識に要する時間の平均である。表から、Lの値が増えるにつれて平均認識率が上がっているが、L=4を超えた辺りから認識率の上昇は頭打ちとなっていることがわかる。また、処理時間については、Lの値を大きくしてもわずかな増加で抑えられているが、長文を書くことを考えた場合、処理時間を出来るだけ抑えられるようLの値を小さい値にすることが望ましい。以上の結果から、辞書に新たに追加可能な数は4が適しているといえる。

### 4. おわりに

本稿では、空中に書いた文字を認識するシステムの試作を行い、そのシステムにより認識実験を行った結果を報告した。認識実験の結果、本システムにより空中に書いた文字を74.3%認識することができた。さらに、誤認識した文字に対して学習を行うことで、空中に書いた文字を87.2%認識することができ、認識率を約13%上げることができた。

今後は、辞書にコードを追加する方法についてさらなる検討を行い、文字認識率の

表 4. 辞書追加数を変化させたときの平均認識率

	L=0	L=1	L=2	L=3
筆記者 A	82.6%	89.6%	91.2%	91.4%
筆記者 B	76.6%	85.3%	89.6%	90.1%
筆記者 C	73.1%	82.4%	87.1%	88.3%
筆記者 D	68.5%	75.7%	77.6%	78.7%
筆記者 E	70.8%	77.9%	81.1%	83.5%
平均認識率	74.3%	82.2%	85.3%	86.4%
平均処理時間	497msec	546msec	548msec	560msec
	L=4	L=5	L=6	L=∞
筆記者 A	92.4%	92.4%	92.4%	92.4%
筆記者 B	90.8%	90.9%	90.9%	90.9%
筆記者 C	89.1%	89.1%	89.3%	89.3%
筆記者 D	80.2%	80.3%	80.0%	79.9%
筆記者 E	83.3%	83.6%	83.5%	83.5%
平均認識率	87.2%	87.3%	87.2%	87.2%
平均処理時間	571msec	577msec	584msec	584msec

向上を目指す予定である。また、現在はデバイスの設計上、筆を降ろして筆記中であるか、それとも単に筆を上げて移動中であるかを判断できず、一画ごとに動画像を生成しているが、今後は筆記中であるかを判断可能なデバイスを作成し、ユーザが扱い易いシステムにしたいと考えている。

### 参考文献

- [1] 尾本幸弘, 後藤真孝, 伊藤克亘, 小林哲則: “音高の意図的な変化を利用した音声入力インタフェース” 電子情報通信学会論文誌 D-II, Vol.J88-D-II, No.3, pp.469-479(2005年3月)
- [2] 片桐雅二, 杉村利明: “ビデオカメラを用いた空中署名による個人認証の試み”, 電子情報通信学会技術研究報告, Vol.101, No.125, pp.9-16(2001年6月)
- [3] 園田智也, 松岡洋一: “空中での手書き文字入力システム”, 電子情報通信学会論文誌 D-II, Vol.J86-D-II, No.7, pp.1015-1025(2003年7月)
- [4] 西田好宏, 苗村昌秀: “ビデオカメラを利用した空中非目視手書き文字入力方式”, 電子情報通信学会技術研究報告, Vol.105, No.608, pp.119-124(2006年2月)

- [5] 森田俊彦: “局所相関演算による動きの検知と追跡”, 電子情報通信学会論文誌 D-II, Vol.J84-D-II, No.2, pp.299-309(2001年2月)
- [6] 木村義政, 小高和己, 鈴木章, 佐野睦夫: “携帯型ペン入力インタフェース用個人辞書の学習” 電子情報通信学会論文誌 D-II, Vol.J84-D-II, No.3, pp.509-518(2005年3月)