

## 空中に指先で描かれた 続け書き数字の認識アルゴリズム

大倉 充<sup>†1</sup> 真鍋 亮介<sup>†2</sup>  
島田 英之<sup>†1</sup> 島田 恭宏<sup>†1</sup>

近年、ウェアラブルコンピュータや携帯電話などへの新たな入力方式の研究がさかんになりつつある。それらの機器では入力面の大きさに制約が生じることから、空中で描いた文字をカメラで撮影し認識することが考えられている。筆記具として指を考えた場合、手領域の抽出と指先の検出が大きな問題となるが、本論文では、指先が特定されたと仮定した場合の文字認識処理について検討している。空中に文字を書いた場合の問題点として、続け書きとなることで1文字の始点・終点が不明になることがあげられる。本論文では認識対象を数字に限定して、この問題に対応可能なアルゴリズムを考案し、認識実験よりその有効性を確認した。

### A Recognition Algorithm of Numerals Written in the Air by a Finger-tip

MITSURU OHKURA,<sup>†1</sup> RYOHSUKE MANABE,<sup>†2</sup>  
HIDEYUKI SHIMADA<sup>†1</sup> and YASUHIRO SHIMADA<sup>†1</sup>

In late years a study of a new input method to wearable computer or a cellular phone has become active. A constraint occurs in dimension of input aspects of those machineries, so the recognition of characters which are written in the air by a finger-tip is examined. One of problems of this plan is that the characters are written continuously. We limited a recognition object to a number in this report and devised the algorithm that could cope with this problem and confirmed the effectiveness than a recognition experiment.

### 1. はじめに

コンピュータの小型化、軽量化にともない、コンピュータを携帯するウェアラブルコンピュータ環境が実現化されている。ウェアラブルコンピュータの小型・軽量という性質により、一般的な入力機器であるキーボード、マウスなどは装置上にその領域を占めることでウェアラブルコンピュータの利点を損なうことになる。このことから従来の入力インタフェースとは異なる、新しい入力インタフェースがウェアラブルコンピュータには求められることになる。

近年、上記のことからコンピュータへの入力手段として、人間の動作を用いることが考えられており、たとえば指先の描く軌跡（文字、図形の描画）が注目されている。すなわち、空中に指先で描かれた文字、図形をコンピュータへの入力指示として使用するということである。

利用者が指先で空中に描いた文字を入力することのできるシステムを想定した場合、必要となる処理を大別すると、以下の3つの処理となる。

- (1) 環境変動に耐えうる手領域の抽出処理
- (2) 指先検出のための手領域の形状認識処理
- (3) 指先の軌跡で表される続け書きされた文字の認識処理

本論文では、処理(1)、(2)が構築されたと仮定した場合の処理(3)について検討している。

3次元空間内に描かれた文字パターンは、基本的に非目視文字、完全自由手書き文字および続け字となる。文字認識研究の観点からこれらの特徴を眺めると、まず筆記者の意図しない文字パターンの生成される可能性が高く、通常の紙面に書かれた文字パターンよりも手書きひずみの多様性が増加する。また単なる続け字ではなく、1つの文字が一筆書きされることにより、異なる文字が同形状のパターンとなる場合があり、さらに1文字からのストロークの分離の問題が生じる。加えて、前文字と後続文字の一部、場合によってはパターン全体が重なる可能性も否定できず、文字列からの1文字ごとの切り出しもかなり困難な問題となる。

<sup>†1</sup> 岡山理科大学  
Okayama University of Science

<sup>†2</sup> 四国システム開発株式会社  
Shikoku Systems Development Corporation

空中に描かれた文字の認識の研究例として、たとえば文献 1)–3) があるが、それらの研究では続け書きされた複数の文字の認識は検討しておらず、個別文字の認識にとどまっている。しかし入力インタフェースへの応用を考えた場合、ユーザが 1 文字の始点・終点をシステムに伝える動作が必要といった条件は好ましくない。

本論文では、新たな入力インタフェースの構築を目的として、空中に指先で続け書きされた文字の認識アルゴリズムの検討・作成を行った。なお、認識対象字種の検討は重要であるが、本論文では数字を認識対象として認識アルゴリズムを考案した<sup>4)</sup>。

## 2. 数字認識処理の概要

数字を続け書きした場合の問題点の 1 つは、文字列からの 1 文字ごとの切り出しの困難さ、つまり 1 文字の始点・終点が不明なことである。この問題に対し本論文では、入力フレームごとに得られる軌跡の点を終点と見なし、書き順とは逆向きにそれまでに得られている点列に対して毎回マッチングを行う手法を提案する。つまり数字が認識されて初めて始点が特定されることになる。1 文字が認識されると、それまでの点列は削除され再び次の文字の認識処理に入る。

### 2.1 数字の記述ルール

個人による数字の記述法の違いを調べるために 9 人の被験者に数字を書いてもらったが、個人によって数字の書き出し位置に違いが見られた。本論文では、各数字に対してテンプレートを 1 つだけ作成しているため、書き出し位置の違いは認識率の低下を引き起こす。そのため図 1 に示す表記法を定めることにより、使用者ごとに現れる個人性の影響を軽減することとした。図中、 $\bullet$  が記述開始点、 $\circ$  が記述終了点を表している。

### 2.2 特徴量

まず認識の対象となる軌跡(点列)をベジュー補間し、 $N$  個の点にリサンプリングする。次に書き順とは逆方向に 2 点間の角度を求め、獲得した  $N-1$  個の角度を特徴量とした。図 2 に示すように水平方向を 0 度、時計方向を正方向とし、得られた角度を  $1/10$  にして値域を 0 から 35 までの整数値としている。特徴点数は予備実験の結果より  $N=15$  とした。

### 2.3 テンプレートの作成

「0」~「9」の各数字に対して 1 つのテンプレートを作成した。例として数字「0」のテンプレートの作成法を述べる。まず被験者 9 人に「0」~「9」の数字を書いてもらう。任意の 1 人が書いた「0」と残り 8 人が書いた「0」~「9」の特徴要素を 1 つずつ比較し、各比較における一致率  $H = M/(N-1)$  を求める。ここで  $M$  は 2 つの特徴量を比較した際に、要

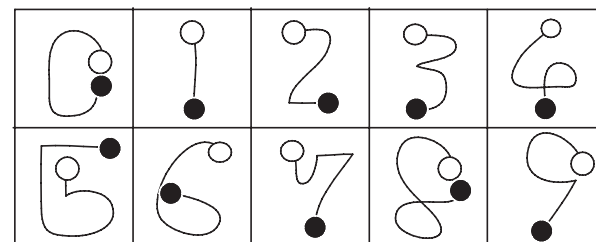


図 1 各数字の表記法  
Fig. 1 Notation of numerals.

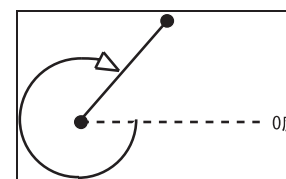


図 2 角度の定義  
Fig. 2 Definitions of an angle.

素の値が同じである要素数を表す。要素どうしの値が  $\pm 2$  の範囲内の違いであれば、要素の値は一致していると判定した。閾値  $T$  を設定し、 $T$  を 0.1 から 0.9 まで 0.1 刻みで増やし、 $H \geq T$  となる「0」の個数と「1」~「9」の総個数を各  $T$  に対して求める。そして「1」~「9」の総個数が 0 個となる最小の  $T$  を求め、その場合の「0」の個数を 9 で割った値を任意の 1 人が書いた数字「0」の認識率と見なす。被験者によって書かれた「0」の中で最も認識率の高い「0」の特徴量を、その際の閾値  $T$  とともにテンプレートとして採用する。「1」から「9」のテンプレートも同様の方法で作成した。またテンプレートを構成する各要素には重みを付けている。以下に、数字「0」を例として重み設定の考え方を述べる。なお、テンプレート作成時に 9 人に書いてもらった 90 サンプルを引き続き利用する。まず「0」のテンプレートと 8 人が書いた「0」の特徴量を比較し、要素ごとに 8 人の中で何人の書いたサンプルと一致しているかを調べる。また「0」と最も誤認しやすい数字を調べ(調査した結果は「6」であった)、同様に、要素ごとに一致している人数を調べる。テンプレート「0」の場合、要素番号 1, 2, 3, 4 の特徴量(数字の記述終了付近)において 8 人が書いた「0」との一致人数が多く、「6」との一致人数が少なかった。このことから、要素 1, 2, 3, 4 の特徴

量が「0」の特徴をよく表し、逆に「6」の特徴を表さない特徴要素であると考えられる。そこで、これらの要素に他の要素より大きな重みを与えることで、テンプレートと「0」との一致率を高くし「6」との一致率を低くした。同様の考え方で「1」から「9」のテンプレートの各特徴要素に重み付けを行った。

#### 2.4 マッチング処理

本論文で提案する手法では、各フレームにおいて複数回のマッチング処理を行う。これは1文字の始点と終点が不明で、前文字と後続文字の間に認識の障害となる軌跡（運び）が入っているからである。本論文では、数字が画面内にある程度の大きさで描かれることを想定しており、多くの場合、数字を構成する点列が20点以上であることを確かめている。そこで、最も新しい取得点を終点と見なし、これまでに得られた点列中の1点（終点から見て20点前の点）を始点と仮定して最初のマッチング処理を行い、1点ずつ逆方向に始点をずらして全点とマッチング処理を行う間に、入力された数字を認識する手法を考案した（ベジュー補間は始点の変更されるたびに行い、特徴点数は  $N = 15$  に固定している）。現時点では全点とのマッチング処理を行うため、逆方向へのマッチング処理に特に意味があるわけではない。しかし、判定処理の精度が高まれば、全点とのマッチング処理を行う必要がなくなる。その場合には、「運び」を構成する可能性のある点列（始点候補）からのマッチング処理ではなく、終点から始点候補に向かってのマッチング処理を行うべきであると考え、本論文では逆方向へのマッチング処理を行うことにした。数字が認識された場合には、これまでに得られた点列をすべて削除し次の数字の認識処理に入り、認識不能の場合には、次のフレームで得られた点を終点として同様の手順でマッチング処理を行う。マッチング処理に用いた一致率  $R$  を式 (1) に示す（この式は実験を繰り返すことで導いた）。

$$R = \frac{\sum_{i=1}^{N-1} w_i \times s \times m}{\sum_{i=1}^{N-1} w_i \times (s \times m + t \times n)} \quad (1)$$

$$t = |ki - li| \quad (|ki - li| \leq 18) \quad (2)$$

$$t = 36 - |ki - li| \quad (|ki - li| > 18) \quad (2)$$

$$s = angle + 1 - t \quad (3)$$

式中、 $w_i$  は各テンプレートの要素ごとに与えた重みを、 $ki$  は入力パターンから抽出された特徴要素を、そして  $li$  は任意の数字のテンプレートの特徴要素を表している。 $m$  は特徴要素が一致していると判定された場合に値1となり、一致していないと判定された場合に値0

となる変数である。また、 $n$  は  $m$  とは逆の値となる変数である。 $t$  は要素値の差であり、値域は0から18（18以上の差が生じた際は式(2)の下の方の式を用いる）である。なお、後述する理由により特定の字種においては  $t = (36 - |ki - li|)/2$  を用いた場合もある。 $t$  も重みの役割を果たしており、 $t$  が大きければ（角度の差が大きい、つまり要素は一致していない）、 $R$  の値は小さくなる。逆に一致している場合は、要素の差が小さい（より一致している）ほど重みを高くするべきなので、式(3)によって重み  $s$  を設定する。式(3)中の  $angle$  は前述した要素どうしが一致していると見なす（角度）の範囲である（基本的には2である）。テンプレートの要素すべてと一致している場合に  $R$  は1となり、すべて一致していない場合に  $R$  は0となる。 $R$  の値が数字別に定められている閾値  $T$  を超えた場合に認識候補数字と見なされ、すべてのマッチング処理後、最大の  $R$  を示した数字と入力パターンは判定される。

#### 2.5 追加処理

空中に数字を描画する場合、たとえば、「5」の書き始めの下へのストロークは「1」と同形状の軌跡となり、また「3」の描画では描画途中までは「2」の形状に近い軌跡となる。これらの空中文字描画特有の問題に対して以下の追加処理を実施した。

「1」と同形状の軌跡は描画途中に頻繁に現れる。そのため、画面半分以上の大きさの数字が書かれると認識処理を行うという条件を設定することで、この問題に対処した。また「5」だけでなく「4」、「7」、「9」の中にも「1」と同形状の軌跡が含まれており、「4」、「7」、「9」の認識途中で「1」が認識されてしまう場合もある。そこで他の数字が認識候補になっている場合は「1」を認識候補から除外するようにした。なお「1」に対しては、特徴量を比較する際に要素が一致していると判定する範囲を  $\pm 1$  と厳しく設定した。

上述した「3」描画中に現れる「2」と同形状の軌跡だけでなく、続け書きの場合、「4」を他の数字に続いて記述した場合、前に認識した数字の終点からの「運び」によって「2」と同形状の軌跡が現れる場合がある。本論文では続け書き数字を認識するために、各数字の特徴としてストローク中に存在する鋭角を補助的に利用した。たとえば、前文字からの「運び」と「4」の書き始めの斜め下方向へのストロークがなす角度は鋭角であるが、「2」の対応する位置は丸みを帯びており鋭角は存在しない。そこで続け書きで誤認識を引き起こす可能性がある「2」と「3」、「4」、「7」、互いに誤認識を起こす可能性がある「5」と「8」への対策として、鋭角の個数と大きさを使用した判定処理を行った。表1に本論文で設定した鋭角の条件を示す。これらの条件は予備実験を繰り返すことで定めた。2.4節で述べたマッチング処理の結果、これらの数字が認識候補となった場合には、鋭角の条件を満たすか否かの判定処理を行い、条件を満たした場合のみ認識候補とした。

表 1 鋭角の条件  
Table 1 Conditions of acute angles.

数字	ストローク中に存在する鋭角
2	正方向の鋭角は存在しない. 負方向の鋭角が 1 つ存在し, それは $-20^\circ$ 以上である.
3	水平軸を挟んで $-20^\circ$ から $20^\circ$ の範囲内の鋭角が 1 つ存在する.
4	鋭角が少なくとも 2 つ存在する.
5	正方向, 負方向の鋭角がともに 1 つずつ存在する.
7	正方向の鋭角が少なくとも 1 つ存在する.
8	鋭角は存在しない.

### 3. 認識実験

撮影場所の背景全体を青色にし, 被験者の指先に緑色のマーカを付けることで指先を正確に特定できるようにして, 考案した認識アルゴリズムの評価のために以下に示す 3 種類の認識実験を行った. 被験者の手の甲が撮影できるように Web カメラを被験者の肩付近に設置し, できるだけ指先を前後に移動させずに数字を描画することを実験における数字描画の条件とした.

#### 実験 (1): テンプレートの評価実験

被験者 10 人から 1 文字ごとの数字データを 160 個 (各数字  $\times$  16 個) 取得し, 1 文字ごとの認識実験によりテンプレートの有効性を調べた.

#### 実験 (2): 2 数を重ねて書いた場合の 2 桁の続け書き数字の認識実験

被験者 10 人から 2 桁の続け書きされた数字データを 500 個 (10 字  $\times$  10 字  $\times$  5 個) 取得し, 2 数を重ねて書いた場合の続け書き数字の認識実験を行った.

#### 実験 (3): 2 数を横に並べて書いた場合の 2 桁の続け書き数字の認識実験

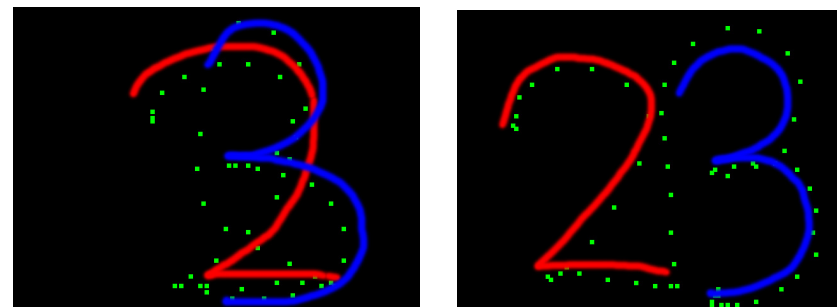
被験者 8 人から 2 桁の続け書きされた数字データを 400 個 (10 字  $\times$  10 字  $\times$  4 個) 取得し, 2 数を横に並べて書いた場合の続け書き数字の認識実験を行った.

実験 (2), (3) における数字を表す点列の例を図 3 に示す. 点列だけでは記述された数字が視認しにくいと考え, 図中には手書きの数字を書き入れている.

#### 3.1 実験結果

##### ・実験 (1)

実験結果を表 2 に示す. 認識失敗数は誤認識・未認識した回数である. 未認識とは, 数



(a) 重ね書きされた 2 数 (a) Two numerals overwritten.  
(b) 横書きされた 2 数 (b) Two numerals written horizontally.

図 3 記述された数字 (点列) の例  
Fig. 3 Examples of described numerals.

表 2 認識実験 (1) の結果  
Table 2 A result of recognition experiment (1).

認識対象数字	認識失敗数	認識率
0	0	1.000
1	①	0.938
2	0	1.000
3	1	0.938
4	③	0.813
5	1	0.938
6	2	0.875
7	③	0.813
8	①	0.938
9	①	0.938

字が記述されているにもかかわらず記述途中と判定されて認識結果が出力されなかったことを示す. 表中, 丸数字が未認識の回数を表している. 認識率は (認識に成功した個数)  $\div$  (各数字の取得個数 = 16) である. なお実験 (1) ではテンプレートの評価を目的としているため, 描画終了を使用者の方から通知して, 1 文字ごとの認識処理を行った.

本論文で作成したテンプレート「0」~「9」の平均認識率は 0.919 であり, ある程度の精

度で認識を行うことができているが、「4」、「7」の認識率が他の数字よりも低くなっている。1文字の個別認識精度は続け書き数字の認識精度に大きな影響を与えるため、より高い認識率が必要となる。

本実験では、「4」、「7」の認識失敗はすべて未認識であり、次のような原因が考えられる。たとえば数字「7」では、第2ストロークの途中で水平方向から左下方向に進行方向が大きく変化する。本論文ではDPマッチングの考えを取り入れていないため、急激にストロークの進行方向が変化すると、入力パターンの特徴要素とテンプレートの対応する特徴要素にズレを生じ、式(2)における $t$ の値が18を超えてしまう場合がある。この問題への対策として、たとえば記述ルールとして「7」を構成する各直線の比率をテンプレートから決めておけば、テンプレートに対応した場所に点がとられるようになり、認識率が高くなると予想できる。しかしそれは使用者の負担になる条件であり、使用者にとって使いやすいとはいえない。そこで「7」のテンプレートの特徴量で、隣り合う要素値の差が大きい要素番号では、比較する要素と一致していない場合に、その差を2分の1にして、ズレによる一致率の低下の影響を少なくした。なお「4」もストローク内に進行方向が大きく変化する箇所が3つあるため、要素のずれによる影響が大きい。そこで「7」と同様に要素が一致しない場合に、その差を2分の1にする要素番号を設定した。以下に示す実験(2)、(3)は、要素のずれを考慮して実験を行っている。

#### ・実験(2)

実験結果を表3に示す。左縦欄が1文字目、上横欄が2文字目を示している。表内の値は誤認識・未認識した回数であり、未認識の回数は丸数字で表している。

#### ・実験(3)

実験結果を表4に示す。表の形式は表3と同じである。

### 3.2 考察

実験(2)は重ねて書かれた2桁の続け書き数字の認識実験である。表3に示すように認識率0.988と高い認識率が得られた。しかし次の3つの状況において誤認識することが確認された。

- (1) 2.5節で述べたように続け書きを行う場合の「2」は多くの状況で誤認識される数字であり、その対策として認識するための条件を厳しくしている。そのため空中での数字描画にある程度慣れていないと認識されないことがある。
- (2) 2文字目の「5」が「3」と誤認識されてしまう場合があった。これはテンプレート、閾値の設定に課題があることを示す誤認識である。本論文では、9人の被験者が書いた

表3 認識実験(2)の結果

Table 3 A result of recognition experiment (2).

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0	0
9	0	0	①	0	0	0	1	0	0	0

表4 認識実験(3)の結果

Table 4 A result of recognition experiment (3).

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	②	0	0	0	0	1	1	0
6	0	0	①	1	0	1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	1	0
9	0	0	0	0	0	1	0	0	0	0

た数字の中から最も認識率の高いものをテンプレートとして採用しているが、要素単位で値を設定すればマッチング処理の精度も高くなると思われ、この問題に対処できると考えている。

- (3) 2文字目の「6」が「0」と誤認識された。これは(2)の問題と同様に、1文字目を描画した後の「運び」によって引き起こされた問題である。この問題への対策としては、

たとえば「0」の表記法を変更して、始点は「0」を構成する点列において最小の垂直座標を有する、といった条件を与えることにより解決できると考えている。

実験(3)は並べて書かれた2桁の続け書き数字の認識実験である。表4に示すように認識率0.97と高い認識率が得られた。しかし次のような場合に誤認識が起こることが確認された。

- (1) 1文字目の終点が画面の下部にくる場合に2文字目の「1」が「7」と誤認識された。「7」の始点は画面上部にくるという条件を設定することで、この問題への対処が可能であると考えている。
- (2) 2文字目に書かれた「2」が未認識となる場合があり、次の2つの原因が考えられる。
  - 1文字目の終点から「2」の開始点までの「運び」が直線状となり、「2」の特徴を表す描画開始時の丸みが不十分で鋭角が出現した。
  - 1文字目を画面の中央に書いたために、2文字目の「2」を書くスペースが小さくなってしまったために、描画開始時の丸みが不十分で鋭角が出現した。
 これらの問題に対しては、現時点では、使用者の描画の慣れによって解決する以外に対策がない状況である。
- (3) 2数を重ねて書く場合と同じ理由で、2文字目の「5」が「3」と誤認識された。
- (4) 2文字目の「8」が「3」と認識された。これは(3)の問題と基本的には同じであるため、テンプレートの改良によって対処できると考えている。

最後に、実験(2)、(3)ともに現れた問題として、被験者が意図せずに描いた軌跡を数字として認識してしまったことがあげられる。この問題は空中における文字描画の本質的な問題であり、現時点では未検討となっている。

すべての処理をC言語で記述し、Pentium IV、3.0GHzのCPUを搭載したDOS/V機を用いて、Vine Linux2.6上で認識実験を行った結果、スループットは7.45フレーム/秒であった。

#### 4. おわりに

本論文では、空中に描かれた続け書き数字の認識アルゴリズムの提案を行った。続け書きの場合、始点、終点を特定することができない。そのため、毎フレーム取得する点を終点と見なし、また、これまでに取得した点を始点と仮定して、フレームごとに始点候補となる点の数だけマッチング処理を行う手法を考案した。本手法では、すべての始点候補点とマッチング処理を行い、その間に入力数字が認識されれば、これまでに得た点列を削除し、認識

されなかった場合は次のフレームで得る点を終点として再びマッチング処理を行う。さらに数字を構成しない点列である「運び」により、任意の数字を描画中に他の数字の軌跡を描いてしまうという続け書きにおける根本的な問題に対して、本手法では、各数字のストローク中に存在する鋭角の個数、また大きさを特徴量として用いることで対処した。

続け書きされた2桁の数字を認識する小規模な実験を行い、のべ被験者数18人から取得した900個のデータに対して認識率98%を得た。ただし本論文では、テンプレート作成に使用されたデータを記述した被験者と認識実験の被験者が同一である場合があるため、得られた認識率は(一種の)学習パターンに対する認識率と考えられる。加えて、被験者はある程度空中における数字描画に慣れた研究室内の者であった。そのため、初めて空中における数字描画を行う被験者による認識実験を行う必要がある。

実用化に向けては、使用者の筆記速度への対応、記述ルールや描画位置条件の緩和、認識処理時間の短縮など、検討すべき事項が多い。

また、本論文における認識対象文字は数字だけに限定されているが、英字<sup>5)</sup>、記号なども認識対象とすることが可能であれば、より利便性の高いシステムの実現につながると考えている。今後は、環境変動に左右されない手領域抽出<sup>6)</sup>、指先検出のアルゴリズム開発を行い、本論文で述べた認識アルゴリズムの改良を行って、それらを統合したシステムの開発を目指したい。

謝辞 岡山理科大学工学部情報工学科南原英生教授と東恒人教授のご激励がなければ、本論文の完成はなかったと思っています。両先生に深謝します。

#### 参 考 文 献

- 1) 園田智也, 村岡洋一: 空中での手書き文字入力システム, 信学論 D-II, Vol.J86-D-II, No.7, pp.1015-1025 (2003).
- 2) 大倉 充, 森田展行, 濱本高志, 島田英之, 島田恭宏, 宮垣嘉也: 空中に描かれた数式を計算するシステムの試作, 映像情報メディア学会誌, Vol.57, No.10, pp.1367-1372 (2003).
- 3) 保呂 毅, 稲葉雅幸: 複数カメラを用いた手書き文字認識システム, 第14回インタラクティブシステムとソフトウェアに関するワークショップ(WISS2006), (デモ発表) (Dec. 2006).
- 4) 久茂田哲也, 真鍋亮介, 濱本高志, 大倉 充, 島田英之, 島田恭宏: 空中に指先で描かれた続け書き数字の認識実験, 平成17年度電気・情報関連学会中国支部第56回連合大会, 22-2, pp.353-354 (Oct. 2005).
- 5) 田畑友裕, 菅野貴文, 島田英之, 島田恭宏, 大倉 充: 空中に指先で描かれた続け書

きアルファベットの認識, 平成 19 年度電気・情報関連学会中国支部第 58 回連合大会, 22-15, p.221 (Oct. 2007).

- 6) 真鍋亮介, 濱本高志, 島田英之, 島田恭宏, 大倉 充, 宮垣嘉也: 動画像中の指先検出を目的とした動的肌色学習法, 平成 16 年度電気・情報関連学会中国支部第 55 回連合大会, 112215, pp.287-288 (Oct. 2004).

(平成 22 年 9 月 29 日受付)

(平成 22 年 11 月 5 日採録)



大倉 充 (正会員)

1983 年愛媛大学工学部機械工学科卒業, 1985 年同大学大学院修士課程修了. 1985~1986 年日立製作所勤務. 1990 年岡山理科大学大学院博士課程修了. 同年同大学助手. 1992 年同講師. 2004 年同助教授. 2007 年同准教授. 現在に至る. 電子情報通信学会, 映像情報メディア学会各会員. 画像処理, パターン認識, HCI, 拡張現実の研究に従事. 理学博士.



真鍋 亮介

2003 年岡山理科大学工学部情報工学科卒業. 2005 年同大学大学院修士課程修了. 同年四国システム開発(株)入社. 現在に至る. 在学中に HCI の研究に従事. 2004 年度電子情報通信学会中国支部奨励賞, および 2004 年度情報処理学会中国支部奨励賞受賞.



島田 英之 (正会員)

1989 年岡山理科大学理学部電子理学科卒業. 1994 年同大学大学院博士課程修了. 同年同助手. 2001 年同講師. 2007 年同准教授. 現在に至る. 電子情報通信学会, 映像情報メディア学会各会員. 画像処理, CG, HCI の研究に従事. 博士(工学).



島田 恭宏 (正会員)

1986 年岡山理科大学理学部電子理学科卒業. 1991 年同大学大学院博士課程修了. 同年同助手. 1997 年同講師. 2006 年同助教授. 2007 年同准教授. 現在に至る. 電子情報通信学会, 映像情報メディア学会各会員. 画像処理, パターン認識, HCI の研究に従事. 理学博士.