

# 空中に描かれた数式の計算システム

大倉 充・森田展行\*・濱本高志・島田英之

島田恭宏・塩野 充

岡山理科大学工学部情報工学科

\*カノーブス株式会社

(2001年11月1日 受理)

## 1. まえがき

近年、3次元空間内に描かれた軌跡（文字パターン）に着目した研究が幾つか報告されている。表1にそれらの研究内容を示す。研究の目的は、多人数間における情報の共有とウェアラブル・コンピュータや携帯電話への新しい入力方式の確立に2分されている。特に新たな入力方式としての側面を考えた場合、場所および記述面の制約を受けないという利点が多い。また入力機器の操作法を覚える必要がなく、マンマシン・インターフェースとしても優れていると考えられる。入力デバイスに関しては様々であるが、処理対象とするデータは指先等の空間座標とスクリーン座標に大別される。どちらの場合においても、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、まず非目視文字であることより、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歪みの多様性が増加すると思われる。また単なる続け字ではなく、一つの文字が一筆書きされることによりストロークの分離の問題が生じ、さらに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、一文字ごとの切り出しもかなり困難な問題となる。前者の問題に関して文献[2]では、手書きひらがなを対象として、一つの文字を構成するストロークの分離に主成分分析を適用して良好な結果が得られたと報告している。また後者の問題に関して文献[4]では、一文字の描画終了時に一定時間の静止状態を設ける方法と一文字の描画開始時に発光素子をオンとし終了時にオフとする方法の2通りの方法を検討しており、オンオフの方法により良好な切り出し結果を得たと報告している。

本研究では、新たなマンマシン・インターフェースの提供を目的とし、空中に描かれた字種（カテゴリー）の数を限定した数字・演算子の認識について検討した。更に一つの応用として、空中に描かれた数式を認識し計算するシステムの試作を行った。ただし実時間処理を実現するために、指先の軌跡ではなく手に持った豆電球の発光点の軌跡を認識するものとした。また上述した一文字ごとの切り出しの問題に対しては、文献[4]と同様に、一文字の描画開始時に豆電球をオンとし終了時にオフとする方法を採用した。

表1 最近の研究例

	研究目的	入力デバイス	入力情報	認識対象
文献[1]	多人数間の情報の共有	空気ペン	ユーザの立ち位置や ペンの空間座標等	—
文献[2]	ウェアラブル・コンピュータや携帯電話への新しい 入力方法の確立	磁気センサ	指先の3次元座標	—
文献[3]	同上	3次元ポインティング デバイス	指先の3次元座標	手書き数字
文献[4]	同上	先端に発光素子を 設けたペン	発光点の2次元座標	一筆書きの空中署名

## 2. システムの概要

屋内にて、被験者をデジタルビデオカメラにより撮影する。被験者は一筆書きの要領で、各文字の描画開始時に手に持った豆電球を点灯させ終了時に消灯させる。ただし各文字には後述する記述順序（記述ルール）が定められており、被験者はそのルールに従った描画を行わなくてはならない。また本システムでは発光点の検出処理を行うために、カメラの撮影範囲には他の発光物体を配置しないようにする必要がある。本来は撮影環境にロバストなシステムが望ましいが、本研究ではこの点に関しては検討しておらず、撮影時に発光点を誤りなく検出するための色範囲は手動で設定することとした。

図1に本システムの概略図を示す。本システムでは、動画像の入力、発光点の検出処理および処理結果のTV画面出力に超高速動画像処理用ボード IMAP-VISION[5]（NEC インキュベーションセンター）を用いている。カメラからの画像は、コンピュータ（以下、PCと記す）に搭載された IMAP-VISION に、フレームサイズ縦240×横256画素、毎秒15フレームのカラー動画像として入力される。IMAP-VISIONでは入力フレーム中の発光点の検出処理を行い、PCに発光点の重心座標データを送る、この処理を、1文字の描画開始時から終了時まで繰り返して得られた軌跡画像の一例を図2に示す。この画像はカテゴリ『2』を示しており、カメラで撮影されたことより鏡像となっている。PCでは、1文字を構成する重心座標データが送られた時点で認識処理を行う。最後に『=』が認識された場合に数式の記述が終了したと判定し逆ポーランド記法による演算処理を行い、認識結果および演算結果を IMAP-VISION に文字列として送る。IMAP-VISION は送られた文字列をTV画面に表示する。なお本システムでは、四則演算のみが可能である。プログラムの開発には、PC側での処理にC言語、IMAP-VISION側では、ボード付属の1DC（One Dimensional C）と呼ばれるCライクな高級言語を用いた。

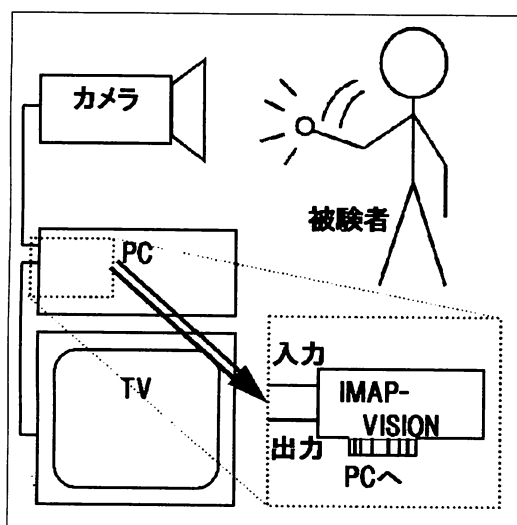


図1 システムの概略図

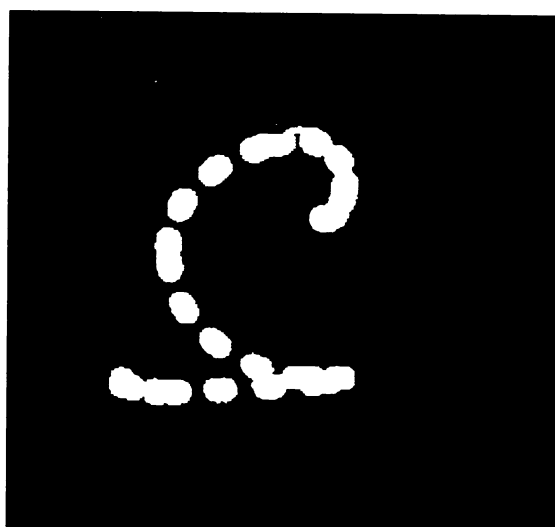


図2 軌跡画像の一例（カテゴリ『2』）

## 3. 認識処理の概要

### 3.1 認識対象カテゴリと記述ルール

本システムの認識対象カテゴリは、『0～9』の数字と『+ - × ÷ =』の演算子および『( ) .』の18カテゴリである。空中に描かれる文字は図3に示す記述ルールに従って一筆書きされる必要がある。図中、黒丸が描画開始点（以下、始点と記す）、白丸が描画終了点（以下、終点と記す）を表している。この記述ルールは認識性能を左右する重要なものであるが、本研究では実験を繰り返すことで定めた。

### 3.2 文字認識処理①

IMAP-VISIONより送られてくる重心座標データが途切れた場合に1文字の入力が終わったと判定し、入力された重心座標群を文字パターンとして認識処理を行う。まず重心座標のデータ数に着目し、10以下の場合には雑音として棄却する。次に重心座標データのx, y座標値の最小値を0とし、全てのデータの座標変換を行う。そして、以下の13種類の判定条件を用いてカテゴリごとに点数付けを行い、入力パターンのカテゴリを決定する。

(1) 始点がパターンの左半分にあるか右半分にあるか

- (2) 始点がパターンの上半分にあるか下半分にあるか
- (3) 始点から見て後続のデータ点が左方向に存在するか右方向に存在するか
- (4) 始点から見て後続のデータ点が上方向に存在するか下方向に存在するか
- (5) 始点近傍に再びデータ点が現れるか否か (『0』『8』『9』)
- (6) 終点がパターンの左半分にあるか右半分にあるか
- (7) 終点がパターンの上半分にあるか下半分にあるか
- (8) 終点に向かうデータ点が左方向に存在するか右方向に存在するか
- (9) 終点に向かうデータ点が上方向に存在するか下方向に存在するか
- (10) 終点がすでに存在するデータ点近傍に位置するか否か (『0』『6』『8』)
- (11) 始点から見て9つの領域のどの場所に終点が位置するか
- (12) 文字を矩形で囲んだ場合に一定サイズよりも小さいか否か (『小数点』)
- (13) 文字を矩形で囲んだ場合に縦長になるか横長になるか

点数付けは、判定条件ごとに各カテゴリー別に定められた点数を基に行われ、判定条件が成立すると点数が加算されていく。表2にカテゴリー別の点数表を示す。ただし、この表での左右は描画時の左右とするため、入力パターンでは左右が反転している。点数の評価としては0点を「間違いなくそのカテゴリーではないと言えるもの」、9点を「間違いなくそのカテゴリーであると言えるもの」として、基本的には各判定条件におけるカテゴリーごとの合計値を最大の9点とした。これは特定のカテゴリーに得点が偏ることを防ぐためである。ただし、明らかにその条件を満たせば候補カテゴリーが絞り込まれる場合は最大27点とした(条件(5)、(10)、(13))。また判定条件(11)では、表2に示した値の1.5倍の点数が加算されることとした。この重み付けの根拠は、他の条件が2者選択であるのに対し、この条件は3者選択を行っているためである。ただしx、y両座標とも中央の場合は、『0』『8』に限定できるため「x座標が中央の点数」×「y座標が中央の点数」が加算されることとした。最後に判定条件(12)は、『小数点』のみを認識するために設定した。表3に記述ルールに従った理想的な文字パターンが入力された場合の獲得点数を示す。表中には獲得点数が2番目に大きいカテゴリーとその獲得点数も示してある。

点数付けが終了後、以下の2つの条件を満たした場合には入力パターンを最高点を獲得したカテゴリーと決定し、満たさなかった場合には棄却する。

- ・ 最高点が100点以上である。
- ・ 最高点と2番目の高得点との差が5点以上ある。

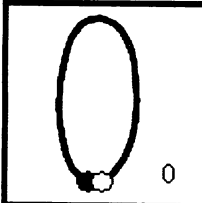
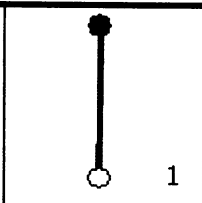
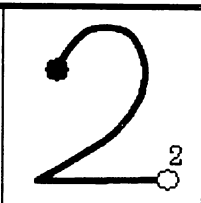
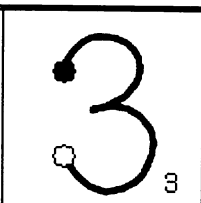
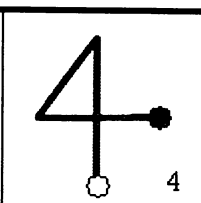
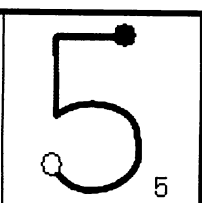
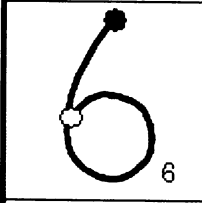
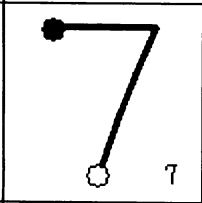
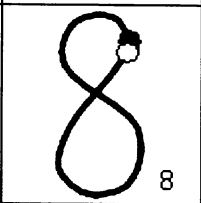
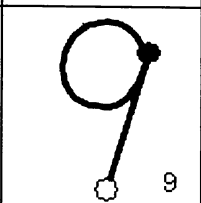
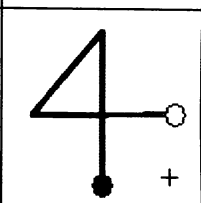
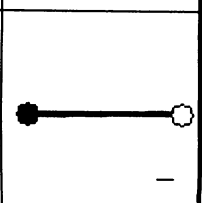
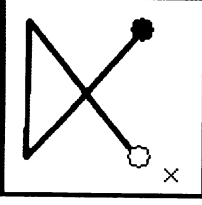
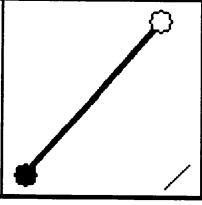
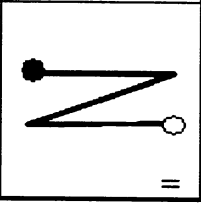
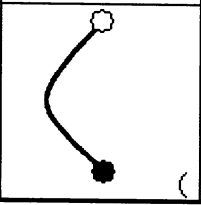
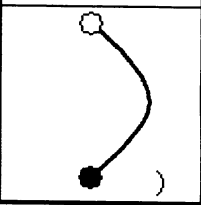
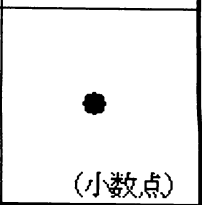
					
0	1	2	3	4	5
					
6	7	8	9	+	-
					
×	/	=	(	)	(小数点)

図3 各カテゴリーの記述ルール

表2 カテゴリー別の点数表

判定条件\カテゴリー		0	1	2	3	4	5	6	7	8	9	+	−	×	/	=	( )	.		
(1)	左半分	4	4	9	9	0	0	2	9	1	0	5	9	0	9	9	0	9	0	
	右半分	5	5	0	0	9	9	7	0	8	9	4	0	9	0	0	9	0	0	
(2)	上半分	0	9	9	9	4	9	9	9	9	9	0	4	9	0	9	0	0	0	
	下半分	9	0	0	0	5	0	0	0	0	0	9	5	0	9	0	9	9	0	
(3)	左方向	9	7	0	0	9	9	9	0	9	9	9	0	9	0	0	9	0	0	
	右方向	0	2	9	9	0	0	0	9	0	0	0	9	0	9	9	0	9	0	
(4)	上方向	9	0	9	6	9	0	0	0	9	8	0	5	0	9	0	9	9	0	
	下方向	0	9	0	3	0	9	9	9	0	1	9	4	9	0	9	0	0	0	
(5)	現れる	27	0	0	0	0	0	0	0	27	27	0	0	0	0	0	0	0	0	
	現れない	0	27	27	27	27	27	27	27	0	0	27	27	27	27	27	27	27	0	
(6)	左半分	5	5	0	9	5	9	9	6	1	6	0	0	0	0	0	0	9	0	
	右半分	4	4	9	0	4	0	0	3	8	3	9	9	9	9	9	9	0	0	
(7)	上半分	0	0	0	0	0	0	4	0	9	0	5	5	0	9	0	9	9	0	
	下半分	9	9	9	9	9	9	5	9	0	9	4	4	9	0	9	0	0	0	
(8)	左方向	2	3	9	0	7	0	0	0	9	0	9	9	9	9	9	9	0	0	
	右方向	7	6	0	9	2	9	9	9	0	9	0	0	0	0	0	0	9	0	
(9)	上方向	9	9	9	3	9	3	6	9	0	9	9	4	9	0	6	0	0	0	
	下方向	0	0	0	6	0	6	3	0	9	0	0	5	0	9	3	9	9	0	
(10)	位置する	27	0	0	0	0	0	27	0	27	0	0	0	0	0	0	0	0	0	
	位置しない	0	27	27	27	27	27	0	27	0	27	27	27	27	27	27	27	27	0	
(11)	x座標	左方向	0	1	0	3	9	9	9	0	0	6	0	0	1	0	0	1	1	0
		中央	9	8	0	6	0	0	0	2	9	3	0	0	7	0	0	7	7	0
		右方向	0	0	9	0	0	0	0	7	0	0	9	9	1	9	9	1	1	0
	y座標	上方向	0	0	0	0	0	0	0	0	0	9	1	0	9	0	9	9	0	
		中央	9	0	0	0	0	0	0	9	0	0	7	0	0	0	0	0	0	
		下方向	0	9	9	9	9	9	9	9	0	9	0	1	9	0	9	0	0	0
(12)	小さい	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	定	
	大きい	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
(13)	縦長	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	普通	27	0	27	27	27	27	27	27	27	27	27	0	27	27	9	27	27	0	
	横長	0	0	0	0	0	0	0	0	0	0	0	27	0	0	18	0	0	0	

表3 理想的な入力パターンの獲得点数および2番目に大きな獲得点数

カテゴリー	0	1	2	3	4	5	6	7	8
得点	224	165.5	180	169.5	170	177	171	174	232
第2カテゴリー	8	5	7	7	5	4	5	3	0
得点	189	133.5	153	148.5	147	146	147	154.5	191
カテゴリー	9	+	−	×	/	=	( )		
得点	171.5	172	161	177	180	168	177	177	
第2カテゴリー	5	/	/	4	( )	−	/	/	
得点	138	153	139.5	145.5	150	148	148.5	148.5	

### 3.3 文字認識処理②

後述する予備認識実験の結果、前述した認識処理だけでは誤認識および棄却を生じる場合の多いことが明らかとなった。そのためシステムに認識処理の追加を行った。以下、本節では、棄却および誤認識を生じる場合の多い2つのカテゴリーの組に対する個別のカテゴリー決定アルゴリズムについて説明する。ここで述べる処理は、前述した認識処理の結果、認識候補順位が1位および2位に2つのカテゴリーが現れた場合に限り行われる。

#### (a) 『2』と『=』の場合

入力パターンの外接矩形枠の縦横の長さの比を調べ、1:1.5以下の縦長であれば『2』、1:1.5より大きな横長であれば『=』と判定する。この追加処理は得点差が5点未満であっても適用されるため、棄却が減ることになる。

#### (b) 『3』と『7』の場合

入力パターンを構成する重心座標データを入力順に調べ、データのx座標値が+方向から一方向に変化する変曲点があるか否かで判定する(図4参照)。調べる座標データは、全データから始点および終点近傍のデータを除いた60%のデータである。変化する個所があれば『3』、なければ『7』と判定する。この追加処理も得点差が5点未満であっても適用されるため、棄却が減ることになる。

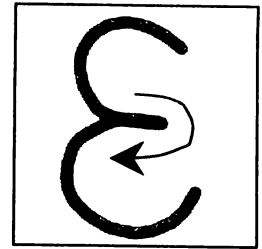


図4 変曲点

#### (c) 『3』と『9』の場合

入力パターンの始点の位置が左右どちらにあるかを調べ、左にあれば『3』、右にあれば『9』と判定する。この追加処理は、得点差が5点以上あった場合にのみ適用されるため、得点差が5点未満であれば棄却されることになる。

#### (d) 『=』と『-』の場合

上述の(b)と同様にデータのx座標値が+方向から一方向に変化する変曲点の有無を調べ、変曲点があれば『=』、なければ『-』と判定する。この追加処理は得点差が5点未満であっても適用されるため、棄却が減ることになる。

## 4. 実験

### 4.1 実験概要

実験は屋内にて行い被験者数は10名である。まず各被験者に各文字パターンの記述ルールを伝え、あらかじめ定めた2種類の数式を空中に豆電球で3回描いてもらった。被験者は文字パターンの描画開始時に手に持った豆電球を点灯させ、終了時に消灯させるといった一筆書きの要領で記述を行うものとした。被験者はカメラに向いて描画し、自分の描いた軌跡および認識結果をディスプレイで確認することはできないことを実験の条件とした。各文字パターンを記述後、筆者らが認識結果を確認し、誤認識あるいは棄却の場合には、被験者にその文字が認識されるまで再入力してもらった。認識実験で記述してもらった2種類の数式を以下に示す。

①  $52 \times 3.14 =$

②  $60 + (8 - 9) \div 7 =$

実験の様子を図5に示す。

### 4.2 実験結果

まず入力された数式が正しく認識された場合には、システムは正しい表示結果および演算結果を出力することが確認された(図6参照)。予備認識実験結果を表4に示す。全ての入力パターンが1回の入力で認識された場合には、各カテゴリーの入力回数は30(『=』は60)となるが、誤認識および棄却が生じた場合には再入力する必要があるために入力回数が増える(100%の認識率を得た場合の総入力回数は570となる)。本研究における認識率Dは次式で定義した。

$$D = (A - B - C) / A \quad A: \text{入力回数} \quad B: \text{誤認回数} \quad C: \text{棄却回数} \quad (1)$$

また表中の認識精度Eとは、郵政省郵政研究所が実施した「文字認識技術コンテスト」[6]における審査項目の1つとして提案されたもので、本研究では次式で定義した。

$$E = (10B + C) / A \quad (2)$$

この認識精度Eの値域は0~10であり、0に近いほど精度が高く、全てのパターンが棄却された場合に1となる。予備実験より、認識率D=0.922と認識精度E=0.180の結果が得られた。誤認識および棄却の内容を調査し、対策として3.3節で述べた認識処理の追加を行い、予備実験と同内容の実験を行った結果を表5に示す。認識率D=0.976と認識精度E=0.163の結果が得られ、認識率、認識精度ともに向上した。

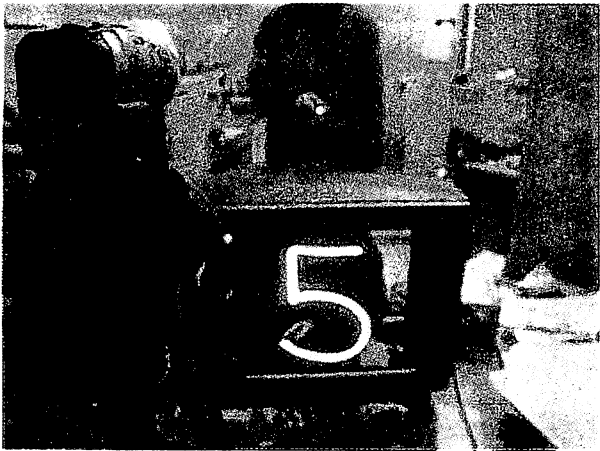


図5 実験の様子

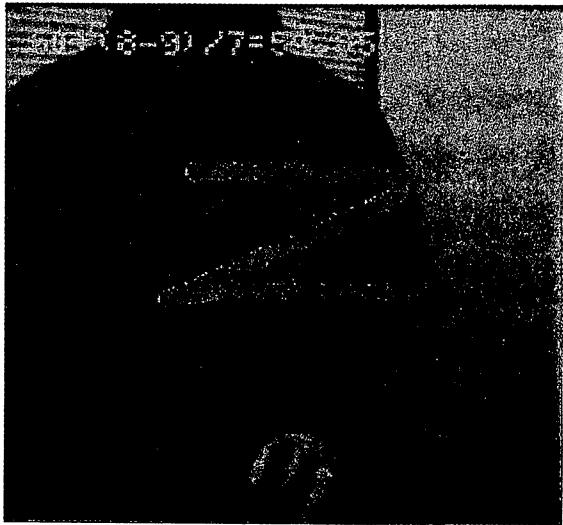


図6 数式演算結果画面

表4 予備認識実験結果

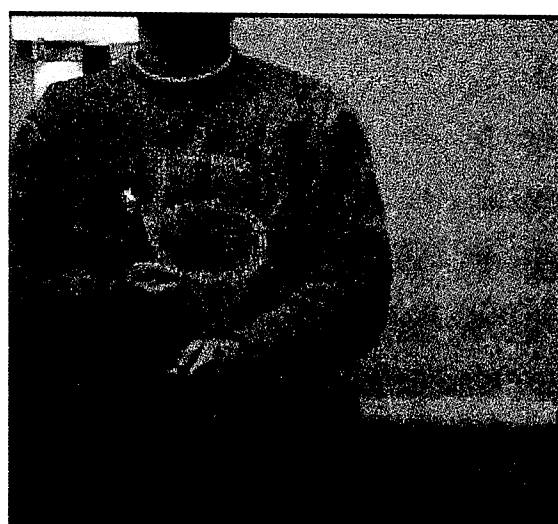
カテゴリー	0	1	2	3	4	5	6	7	8	
A 入力回数	30	30	35	43	30	30	30	30	34	
B 誤認回数	0	0	0	0	0	0	0	0	1	
C 棄却回数	0	0	5	13	0	0	0	0	3	
D 認識率	1.0	1.0	0.857	0.698	1.0	1.0	1.0	1.0	0.882	
E 認識精度	0.0	0.0	0.143	0.302	0.0	0.0	0.0	0.0	0.382	
カテゴリー	9	+	−	×	／	=	(	)	・	計
A 入力回数	34	31	34	30	30	72	30	33	32	618
B 誤認回数	3	0	0	0	0	2	0	1	0	7
C 棄却回数	1	1	4	0	0	10	0	2	2	41
D 認識率	0.882	0.968	0.882	1.0	1.0	0.833	1.0	0.909	0.938	0.922
E 認識精度	0.912	0.032	0.118	0.0	0.0	0.417	0.0	0.364	0.063	0.180

表5 最終的な認識実験結果

カテゴリー	0	1	2	3	4	5	6	7	8	
A 入力回数	30	30	30	38	30	33	31	30	30	
B 誤認回数	0	0	0	3	0	3	1	0	0	
C 棄却回数	0	0	0	5	0	0	0	0	0	
D 認識率	1.0	1.0	1.0	0.789	1.0	0.909	0.968	1.0	1.0	
E 認識精度	0.0	0.0	0.0	0.921	0.0	0.909	0.323	0.0	0.0	
カテゴリー	9	+	−	×	／	=	(	)	・	計
A 入力回数	30	30	30	30	30	61	30	30	31	584
B 誤認回数	0	0	0	0	0	1	0	0	1	9
C 棄却回数	0	0	0	0	0	0	0	0	0	5
D 認識率	1.0	1.0	1.0	1.0	1.0	0.984	1.0	1.0	0.968	0.976
E 認識精度	0.0	0.0	0.0	0.0	0.0	0.164	0.0	0.0	0.323	0.163



(a) カテゴリー『3』



(b) カテゴリー『5』

図7 誤認識となった入力パターン例

## 5. 考察

### 5.1 認識実験結果に対する考察

最終的に得られた認識率は97.6%であり、かなり良好な値が得られたと思われる。しかし予備実験結果に比べると、棄却回数は大幅に減少したものの誤認回数は増加しており、そのため認識精度は0.017しか向上していない。追加した認識処理は、前段階の認識処理で棄却となった場合にも適用されるものが多く、そのために前述した結果になったと考えられる。誤認回数の多かったカテゴリーとして『3』と『5』が挙げられる。これらのカテゴリーの典型的な誤認例を図7に示す。同図に示されるように始点もしくは終点の近傍に他の点列が存在した場合に、(a)の場合には『9』と(b)の場合には『6』と誤認識された。非目視文字であるために、被験者の意図したパターンと異なるパターンが生成された結果である。本システムにおける認識処理ではこの問題への対処が不十分であり、改善の必要がある。今後の課題の一つとしたい。

### 5.2 システムに対する考察

まず本システムへの入力形態について述べる。豆電球を使用している点は大きな問題と考えているが、入力動作として空中に軌跡を描くこと自体は、特にキーボードやタッチペンを使用する場合と比較すると、ユーザに対して非常に負担の少ない操作と思われる。このために情報入力機器に慣れていないユーザ、あるいは片手しか動かすことのできない身体が不自由なユーザにもほとんど予備知識なく使用可能という利点がある。将来的には、豆電球ではなく指の軌跡を入力パターンとすることでユーザへの負担を更に軽減することが望ましいと考えている。ここで述べたシステムへの入力形態の変化が実現するならば、システムの実用性が増し様々な応用システムの開発が可能となる。

入力部では、発光点を色情報を用いて抽出しているが、これは周囲の環境の変化などで閾値が変化することが考えられる。現在、この閾値は手動で与えているが、汎用性を考えると自動で検出させることが望ましい。今後は、数式の入力処理に移る前にシステムになんらかの動きを検出させ、その動いている物体の色を追跡させるなどして対応すべきであると考えている。

次に本システムのパターン認識能力について述べる。被験者数10名という小規模な実験しか行っていないが、得られた結果はかなり良好なものであった。これは、実験を繰り返すことで誤認識を生じやすいカテゴリーの組み合わせを調査し、認識処理を強化したためである。したがって、本システムで認識対象としたカテゴリー群に対する認識能力はかなり高いと考えている。ただし5.1節で述べた課題があり、またカテゴリー数が少ない点と通常の記述順序から逸脱した記述ルールが存在する点を考慮すると、認識処理には改善の必要があることも明らかである。特に、各カテゴリーの記述ルールが認識処理に直接反映される点は改める必要があると考えている。理由として、まず、通常の記述順序に従ったパターン記述がユーザにとって負担が少ないこと、更に、新たなカテゴリーの追加が認識処理に大きな影響を与え、場合によっては認識率の著しい低下をもたらす可能性のあることが挙げられる。

演算部では逆ポーランド記法を採用しているために、仮に新たな演算子の追加があった場合にも演算部の大幅な変更は

必要ないと考えている。なお現時点では四則演算しかできないので演算機能の拡張は必要であり、例えば、整数除算、剰余計算、階乗計算などの追加が考えられる。

次に現時点でシステムに持たせている機能について述べる。まず誤認識された文字の訂正機能がある。この機能は、ディスプレイを見ながらユーザが入力する場合を想定して設定されており、ユーザがパターンを入力後そのパターンが誤認識された場合に、ある一定時間ユーザが入力を停止すると1つ前の表示文字が消去されるというものである。ユーザが入力停止時間を長くすると表示文字が入力の逆順に消去されていき、この消去されるまでの時間は加速度的に早くなる。このため消去開始までの時間設定が問題であり、ユーザが入力に手間取ると正しい表示文字が消去される結果となる。この機能はユーザが認識結果（表示文字）の正誤を判定しているために、認識システムが持つべき本質的な機能とは言えないが、ある種の訂正機能は必要と考え設定した。次に入力パターンが棄却された場合、ユーザにピープ音で棄却されたことを通知する機能を設定した。例えば、ユーザが講演等を行う際にシステムを使用し、その表示画面を聴衆に向けユーザが見ることのできない状況下で、この機能は必要不可欠となる。また現時点では、入力された数式のチェック機能を持たせているが、今後は一文字が認識された時点で、以下に示す不正入力の制限機能を持たせるべきであると考えている。

- ・1つの数字で小数点が連続して複数個入力できない
- ・括弧のバランスチェックを行い、『(』と対応の取れない『)]』は入力できない
- ・続けて複数の演算子が入力できない
- ・数式の記述開始時に『+』『-』『(』以外の演算子は入力できない

## 6. むすび

本論文は、「空中に描かれた数式計算システム」について述べたものであり、試作したシステムを用いた認識実験を行い、得られた結果を示した。被験者数は10名と小規模な実験ではあるが、かなり良好な結果が得られた。入力総パターン数584に対して認識率97.6%、認識精度0.163であり、認識された数式の計算も正確に行われた。システムの認識部では記述者の個人性に起因する手書き歪みへの対処が必要であることが明らかとなったが、システムへの入力方法として指ではなく豆電球を用いる、および一般の記述順序とは異なる記述ルールが存在するという前提条件の下では、基本的なシステムは完成したと考えている。今後は5章で述べた課題に取り組み、環境や入力パターンの変動にロバストなシステムに改良していきたい。

## 参考文献

- [1]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—”, 第59回情処全大論文集(4), pp. 39-40, 1999.
- [2]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.
- [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]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.
- [5]櫻井和之, 許 昭倫, 岡崎信一郎: “IMAP-VISIONを用いた道路画像からの白線検出”, 信学技報, PRMU99-66, 1999.
- [6]能見 正, 松井俊弘, 山下郁夫, 若原 徹, 吉室 誠: “手書き数字認識における誤読・リジェクトパターンの分析—第2回文字認識技術コンテストの実施結果より—”, 信学技報, PRU93-46, 1993.



## A Calculation System for Mathematical Expressions Drawn in the Air

Mitsuru OHKURA, Hiroyuki MORITA<sup>\*</sup>, Takashi HAMAMOTO, Hideyuki SHIMADA,

Yasuhiro SHIMADA and Mitsuru SHIONO

*Department of Information & Computer Engineering,*

*Faculty of Engineering,*

*Okayama University of Science,*

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

*<sup>\*</sup>Canopus Co., Ltd.*

(Received November 1, 2001)

Humans can view the path of a moving object, and give a meaning to the path. For example, when a sign or character is drawn in the air by a finger, humans recognize the drawn sign or character by the path. This is due to the human's ability to perform dynamic scene recognition. In this paper, we attempt to give computer this important human ability. We examined the recognition of letters (categories) limited only to the numerals and operators, and made a trial system which recognizes and calculates a mathematical expression drawn in the air by a miniature bulb. We completed making the overall system, though we still need to improve the recognition rate, and increase the number of operators.