

# 筆順画数同時フリーを実現する 楔文字モデルによるオンライン日本文字認識方式

横田 登志美<sup>†</sup> 葛貫 壮四郎<sup>†</sup> 郡 司 圭 子<sup>†</sup>  
桂 晃 洋<sup>††</sup> 浜 田 長 晴<sup>†††</sup> 福 永 泰<sup>†</sup>

ペン入力インタフェースによるコンピュータや携帯用情報機器が一般的になり、操作性向上のため、筆順の間違いや続け書きの文字を認識できるオンライン文字認識が望まれている。特に、筆順違いをありがちなものに限定せず、筆順を間違えて続け書きした文字までも認識するために「楔文字モデル方式」と呼ぶ認識技術を開発し評価したので報告する。本方式は(1)文字の続け書きを分断するように楔状に分割すること(2)楔の分割数の違いを考慮し筆順によらずに辞書パターンと入力パターンの距離値の算出を行うこと、を特徴にしている。評価した結果、第1候補の認識率83.9%、辞書容量371Kbyte、処理時間0.15秒/文字(Pentium 300MHz使用)を得た。

## An On-line Cuneiform Modeled Handwritten Japanese Character Recognition Method Free from Both the Number and Order of Character Strokes

TOSHIMI YOKOTA,<sup>†</sup> SOUSHIRO KUZUNUKI,<sup>†</sup> KEIKO GUNJI,<sup>†</sup>  
KOUYOU KATSURA,<sup>††</sup> NAGAHARU HAMADA<sup>†††</sup>  
and YASUSHI FUKUNAGA<sup>†</sup>

A new handwritten character recognition algorithm is developed in order to provide a user-friendly pen-input computer system. This is capable of recognizing order-free character strokes and continuously handwritten characters. The features of the algorithm are: i) newly developed cuneiform model which divides continuous handwritten strokes into segments of lines (cuneiform) and tries to match the cuneiform model to the dictionary; ii) calculation of the distance taking both the number of cuneiform and increase in the number of cuneiform segments. We evaluated the algorithm by the TUAT Nakagawa Lab.HANDS-kuchibue\_d-96-02 benchmark database of the Tokyo University of Agriculture and Technology. We obtained the results of the recognition ratio to be 83.9%, the required dictionary space is only 371 Kbyte, and processing time is 150ms/character on a Pentium PC running at 300 MHz.

### 1. はじめに

コンピュータが小型化するにつれ、キーボードに優る携帯性やより自然な紙と鉛筆の操作性が好まれ、ペン入力コンピュータやペン入力を用いた携帯用情報機器の市場が拡大しつつある。このように個人使用が前提の比較的小規模なコンピュータリソース環境では、より快適なユーザインタフェース提供のため、筆順の

間違い、続け書き、また、個人的な癖の表れた文字の認識が可能なこと、そして、認識用の辞書容量が小さく小型機器への組み込みが容易なことが望まれている<sup>1)</sup>。

従来から、このような認識ニーズへの対応は継続的に研究されており、たとえば、続け書きによる変形文字を認識するための画数フリー文字認識方式<sup>2)~5)</sup>や、筆順の制限を緩和する筆順フリー文字認識方式<sup>6),7)</sup>などが報告されている。また、筆順違いと続け書きの両者に有効な筆順画数同時フリー文字認識方式<sup>8)</sup>について、以下のような報告がある。

(1)画対決定方式<sup>8)~10)</sup>

本方式は、まず、距離値が最小になるように、入力パターンと辞書パターンの画の対を決定し、対のない

<sup>†</sup> 株式会社日立製作所日立研究所  
Hitachi Research Laboratory, Hitachi, Ltd.

<sup>††</sup> 株式会社日立製作所研究開発本部  
Research & Development Group, Hitachi Ltd.

<sup>†††</sup> 倉敷芸術科学大学  
Kurashiki University of Science and the Arts

辞書パターン画は前後の画に続け書きされたとして、再度、入力パターンとの対応を取る方式である。

これは、続け書きが1, 2画程度である場合を対象にした方式であり、画数が大幅に減少するような続け書き文字の認識は困難である。

## (2) 要素のコード化方式

本方式は、認識対象を構成線分の要素に分解して、その形状をコード化し、入力パターンと辞書パターンの比較によって認識する方式であり、線分や単一の弧など要素が比較的単純な形状の図形認識<sup>11)~13)</sup>、漢字の認識<sup>14)</sup>で使われている。

この方式によると、あらゆる筆順違いの文字を認識でき、辞書も小さくできるが、筆順画数同時フリー方式として日本語の認識に適用した報告はない。日本語で多用されるひらがなが、形状変動が大きく、かつ、微妙なため本方式で認識精度を上げることが難しいためと考えられる。

## (3) オフライン方式<sup>15)~17)</sup>

本方式は、筆順の影響を無くすために、オンライン文字を画素データに展開してオフライン文字認識(OCR)と同じようなパターンマッチングを行う方式である。演算性能やメモリ容量などコンピューティングパワーの向上にともない、近年になってオンライン認識への適用が研究されるようになってきている。しかし、文字パターンを画素に展開するため、前述の他方式に比べ辞書容量が大きくなる。

以上述べたように、従来報告されている筆順画数同時フリー方式には、省略できる画数やひらがなへの対応などの点で制限があり、改善すべき課題が残っている。

さらに、画数フリーな日本文字認識のための折れ線近似方式は多数報告されているが<sup>2)~19)</sup>、文字の続け書きを分断するという観点から折れ線近似を行うという方式の報告は見あたらない。

従来報告されている、筆跡の屈曲点を拾っていく近似方式では、滑らかな曲線形状に変形した続け書きを正確に分断することが難しく、局部的な筆方向に着目して筆跡を量子化した方向ベクトルに分類する近似方式では、量子化のしきい値前後の筆方向の判定が不安定になるなどの課題が残されている。また、筆跡と近似線分で囲まれる面積が一定値となるように近似線分を決定する面積法ではカーブでの近似が細かすぎ入力と辞書とのマッチング処理が重くなるという欠点が指摘されている<sup>2)</sup>。

上記のような課題の解決を目指し、①形状の変動が大きいひらがなを認識でき、②ありがちな筆順画数違

いに制限せずに筆順画数同時フリーな認識が可能で、あわせて③辞書の小型化を実現できる「楔文字モデル方式」と呼ぶオンライン文字認識方式を開発した<sup>20)</sup>。本論文では、提案方式の認識処理の概要と、開発した認識プログラムの評価結果について報告する。

## 2. 楔文字モデルによる文字認識

### 2.1 原理と特長

筆順画数同時フリーの対象は、楷書書き文字のほか、図1に示す続け書き文字例(12画の文字を4画で書いた場合)のように画数が大幅に減少する過度の続け書き、また、図2に示す筆順違いでかつ続け書きで書かれた文字などで、ここでは、筆方向違いでの筆記文字は含めない。提案方式では、これらのさまざまな筆順・続け書きで書かれた文字を、新たに辞書パターンを登録することなしに認識する。

本提案の楔文字モデルによる文字認識の処理フローを図3に示す。第1ステップ(STEP1)で入力パターンの続け書きを分断するように折れ線近似を行い、第2ステップ(STEP2)で分断後の部分的なストロークの筆順を入れ替わる対応も許しながら、入力パターンと辞書パターンとの筆順によらないマッチングを行う。

本方式は、続け書きを分断したうえで、筆順によらないマッチングをすることで、図1や図2のような、画数が大幅に減少する続け書きや筆順違いの続け書きを1つの楷書書き辞書パターンにより認識できる。このとき、部分的なストロークを要素のコード化方式のようにコード化せず、座標点列のまま扱うことで、形状の変動が大きいひらがなや変形文字でも認識精度を上げることができる。また、オフライン方式で画素に



本来 12 画の文字を 4 画で筆記

図 1 画数が大幅に減少する続け書き文字

Fig. 1 A very cursive character.



本来 5 画の文字の筆順を違えて 3 画で筆記

図 2 筆順違いの続け書き文字

Fig. 2 A cursive character whose segments are not in order with dictionary patterns.

展開する方式と比較し、1文字あたりの辞書容量を従来のオンライン文字認識の容量程度に抑えることができる。さらに、辞書登録方式と比較しても、複数の筆順のパターンを登録する必要がなくなるため、辞書全体の容量を小さくできる利点がある。

楔文字モデルの名称は、折れ線の近似が古代メソポタミアなどで用いられていた楔形文字<sup>21)</sup>を連想したことから名づけている。図4は、楔形文字の例であるが、元は不規則な線からなる象形文字が、粘土板に筆で刻むため、次第に、まっすぐな線の字画からなる形

に変形していったとされる。

以下、折れ線近似した文字パターンを楔文字モデル、折れ線をなす線分を楔と称する。

2.2 楔文字モデルへの変換処理

楔文字モデルに変換するための折れ線近似方式は、続け書きを分断できる細かさで、しかも、筆順に依存しないマッチング処理を実時間に処理できる大まかさで近似する必要がある。以下、図3 STEP1の折れ線近似方式について説明する。

入力文字は、その筆跡を単位時間ごとのタブレット上座標点列として入力するが、タブレットのどの位置にどの大きさで書いた文字をも認識するために、位置と大きさを正規化する。1文字として切り出した座標点列の重心が原点にくるように平行移動して位置を正規化し、その後、各座標点の原点からの距離の平均が一定値  $R = 30$  になるように大きさを正規化する。

図5に、変換処理の例を示す。(1)は、正規化後の筆跡を示している。正規化後、筆跡の始終点、および、筆方向がx軸・y軸方向の正負が反転する筆方向変換点を抽出し、(2)のように近似点  $P1 \sim P6$  を置く。この処理のみであると、点  $P3, P4$  間の筆跡のように、緩やかな曲線を近似できないため、楔  $P(i)P(i+1)$  からの元の筆跡の最も高い位置までの高さ  $h$  を調べ、高さ  $h$  がしきい値  $Th_h = 16$  以上であれば、図5(3)のように、高さの位置に近似点  $Pn$  を置く。

最終段階では、ペンはねなどのノイズを除去するため、長さがしきい値  $Th_l$  以下の楔  $P(i)P(i+1)$  を削除する。さらに、ごく緩やかなたわみであってもx軸・y軸方向の正負が反転した位置に近似点を置いて縦線や横線が2本の楔に分割されてしまうことを防ぐため、連続する楔  $P(i)P(i+1)$  と楔  $P(i+1)P(i+2)$  の角度差  $\theta$  を調べ、角度差  $\theta$  がしきい値  $Th_\theta$  以下の楔を統合する。これらの処理により、(4)の近似点列を得る。

これらの処理の各々は、従来から、それぞれ単独には用いられていたが<sup>22),23)</sup>、上述の順で組み合わせることにより続け書き部分をより正確に分断できるようにしている。

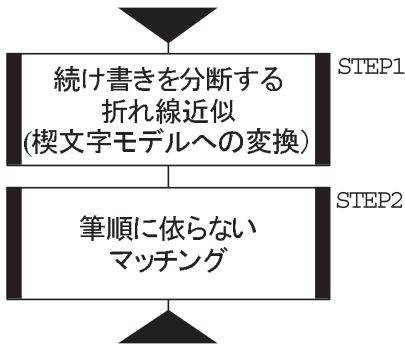


図3 楔文字モデルによる文字認識方式  
Fig.3 Handwritten character recognition method.



Microsoft(R) Encarta(R) 97 Encyclopedia.  
(C) 1993-1997 Microsoft Corporation. All rights reserved

図4 楔形文字  
Fig.4 Cuneiform.

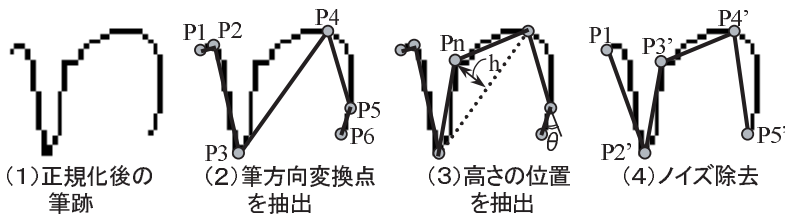


図5 楔文字モデルへの変換  
Fig.5 Conversion to cuneiform model.

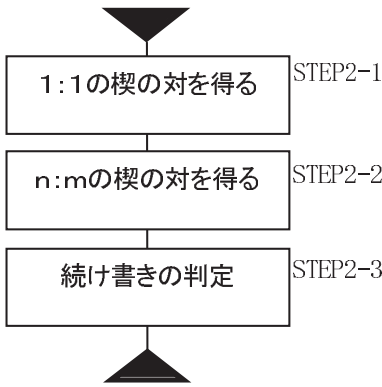


図6 マッチング処理フロー  
Fig.6 Matching process flow.

このような変換処理では、直線や角は安定して抽出できるが、筆跡に曲がりやたわみがあると、1つの楔で近似されるか複数の楔で近似されるかが不安定になりがちである。この問題は、次節に述べるマッチング処理の中で対策している。

ここでは、各々のしきい値を以下のように定めた。大きさの正規化の基準である原点からの距離の平均

$$R = 30$$

高さの位置に近似点を置くしきい値  $Th_h = 16$

ノイズを除去するしきい値  $Th_L^2 = 20$

統合する楔の角度差しきい値  $Th_\theta = 30^\circ$

### 2.3 筆順によらないマッチング処理

楔文字モデルによる認識方式におけるマッチング処理(図3:STEP2)では、あらかじめ楔文字モデルに変換して辞書に定義してある辞書パターンと入力パターンとのマッチングを、入力パターンの楔(以下、入力楔)と辞書パターンの楔(以下、辞書楔)の距離に基づいて、距離の総和が最も小さくなる楔間対応を探索することで行う。この探索は、筆順に依存せずに行い、筆順を間違えて入力した文字は、筆順を入れ替えて辞書とマッチングすることになる。

マッチング処理で各々の辞書パターンとの距離値を求めた後、距離値が最も小さい辞書パターンを認識結果として出力する。楔文字モデルによれば、同じ文字はほぼ等しい楔数になると考えられるため、マッチングの対象とする辞書パターンを、楔数が入力パターンの楔数  $\pm N$  以内のものに絞ることで、マッチング処理の高速化を図ることができる。ここでは、 $N = 4$  を用いた。

図6にマッチング処理の流れを示す。処理は、STEP2-1~STEP2-3の3段階に分かれ、以下、各処理について説明する。

#### 2.3.1 1:1の楔の対の決定

マッチング処理の第1ステップ(図6:STEP2-1)では、楔列からなる入力パターン  $V$  と辞書パターン  $U$  の1:1の楔の対を、対になる楔間の距離値  $d(v(i), u(j))$  の和が小さくなるように決定する。また、対からはずれた入力楔・辞書楔は、その長さをペナルティとしてパターン間の距離値  $D(V, U)$  に加算する。

$$\text{入力パターン } V = \{v(1), \dots, v(N)\} \quad (1)$$

$$\text{辞書パターン } U = \{u(1), \dots, u(M)\} \quad (2)$$

$$d(1:1 \text{ の楔の対}) = \sum d(v(i), u(j)) \quad (3)$$

ただし、 $i = 1 \sim N, j = \text{Pair}(i) = 1 \sim M,$

$i1 \neq i2$  ならば  $\text{Pair}(i1) \neq \text{Pair}(i2)$

$$d(\text{対からはずれた楔}) = \sum L(v(i)) + \sum L(u(j)) \quad (4)$$

ただし、 $i = 1 \sim N, \text{Pair}(i)$  はなし、

$j = 1 \sim M : j \neq \text{Pair}(i)$

パターン間の距離値  $D(V, U)$

$$= d(1:1 \text{ の楔の対}) + d(\text{対からはずれた楔}) \quad (5)$$

理想的には、式(3)で表す楔間の距離値  $d(v(i), u(j))$  :  $j = \text{Pair}(i)$  の総和を最小とするが、実時間内で処理を行うため、安定な結婚の問題<sup>24)</sup>と呼ばれる簡便な解法により、対を決定する。すなわち、入力楔  $v(i)$  と辞書楔  $u(j)$  で、楔間の距離値  $d(v(i), u(j))$  を求めておき、 $i = 1 \sim N$  の順に、入力楔  $v(i)$  について最小値  $\min(d(v(i), u(j)) : j = 1 \sim M)$  をとる  $j = j1$  を選び  $\text{Pair}(i) = j1$  とする。このとき、もし、辞書楔  $j1$  がすでに他の入力楔  $i1$  に選ばれていたら、以下のように  $\text{Pair}(i), \text{Pair}(i1)$  を定める。

$$d(v(i), u(j1)) < d(v(i1), u(j1)) : \text{Pair}(i) = j1$$

とし  $\text{Pair}(i1)$  を  $j \neq j1$  から求めなおす。

$$d(v(i), u(j1)) > d(v(i1), u(j1)) : \text{Pair}(i1) = j1$$

とし  $\text{Pair}(i)$  を  $j \neq j1$  から求めなおす。(6)

この処理を順次繰り返す。この解法によれば、 $O(NM)$  ステップで1:1の楔の対の決定が終了する。

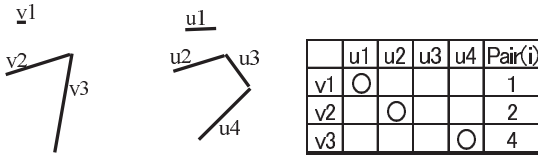
楔間の距離値  $d(v(i), u(j)) = d(v, u)$  は、始点・終点間の距離、および、方向ベクトルの距離の和を用いる。楔  $v, u$  間の距離  $d(v, u)$ 、および、楔の長さ  $L(v)$  を次式のように定義する。

$$d(v, u) = |v(sx) - u(sx)| + |v(sy) - u(sy)|$$

$$+ |v(ex) - u(ex)| + |v(ey) - u(ey)|$$

$$+ |(v(ex) - v(sx)) - (u(ex) - u(sx))|$$

$$+ |(v(ey) - v(sy)) - (u(ey) - u(sy))| \quad (7)$$



1) 入力パターン (2) 辞書パターン (3) 1:1の楔の対

図7 楔の対の決定

Fig. 7 Determining matches.

$$L(v) = |v(ex) - v(sx)| + |v(ey) - v(sy)| \quad (8)$$

ただし、楔 v の始点 (v(sx), v(sy))、

終点 (v(ex), v(ey))

楔 u の始点 (u(sx), u(sy))、

終点 (u(ex), u(ey))

入力・辞書パターン両方に、別々の位置にペンはねなどのノイズが含まれているような場合、これらに対とすると楔間の距離値が大きくなる。そこで、対が決定した  $v(i), u(j)$  間で以下のノイズ判定を行い、パターン間の距離値  $D(V, U)$  を修正する。

すなわち、対と解釈する距離値  $d(v(i), u(j))$  と、対ではなくノイズと解釈する距離値  $L(v(i)) + L(u(j))$  とを比較して、後者の方が小さければ、 $v(i), u(j)$  間は対ではなく、ノイズと判定して各々対からはずれた楔として扱い、式 (5) 第 1 項から距離値  $d(v(i), u(j))$  を減算し、第 2 項に  $L(v(i)) + L(u(j))$  を加算し、パターン間の距離値  $D(V, U)$  を計算しなおす。

図 7 に示した [う] の例では、図 7 (3) に示す 1:1 の楔の対を決定する。u3 は、対からはずれるため、その長さをパターン間の距離値  $D(V, U)$  に加算する。

### 2.3.2 n:m の楔の対の決定

楔文字モデルへの変換では、直線や角は安定して抽出できるが、図 7 の例のように筆跡に曲がりやたわみがあると、1 つの楔で近似されるか複数の楔で近似されるか不安定になりやすい。このような問題点の解決策として、マッチング処理の第 2 ステップ (図 6: STEP2-2) では、前段での 1 対 1 の対に基づいて、 $n:m$  の対を得るような処理を行う。

図 7 に示した [う] の例では、前段での 1:1 の対の決定により、 $(v1, u1), (v2, u2), (v3, u4)$  の対を決定し、u3 は対からはずれる。このように対からはずれた楔については、曲がりやたわみによる楔の数の違いによるものかを以下のように判定し、 $n:m$  の対を得る。

対からはずれた楔については、たとえば、u3 のように前後に対ができていない楔 u2 や u4 があれば、次の 3 通りの解釈での距離値を算出し、距離値が最も小さくなるものを選ぶ。

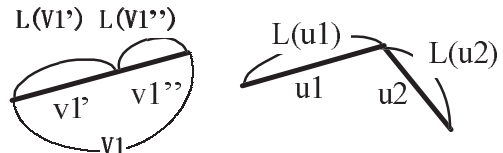
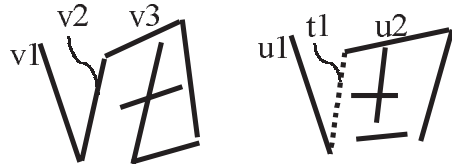


図8 1:2の対での距離値算出

Fig. 8 1:2 Matching distance.



(1) 入力パターン (2) 辞書パターン

図9 続け書きの例

Fig. 9 Example of a cursive character.

- (i)  $(v2, u2), (v3, u4)$  が 1:1 の対、u3 は対からはずれる
- (ii)  $(v2, u2 + u3)$  が 1:2 の対、 $(v3, u4)$  が 1:1 の対
- (iii)  $(v2, u2)$  が 1:1 の対、 $(v3, u3 + u4)$  が 1:2 の対

(ii), (iii) の 1:2 の楔の対応による距離値の算出について、図 8 を用いて説明する。図 8 は、 $(v1, u1+u2)$  が 1:2 の対になる対を示している。この場合、 $v1$  を、 $v1'$  と  $v1''$  に分割するが、 $v1'$  の長さ  $L(v1')$  と  $v1''$  の長さ  $L(v1'')$  の比が、 $u1$  の長さ  $L(u1)$  と  $u2$  の長さ  $L(u2)$  の比になるよう分割し、 $v1'$  と  $u1, v1''$  と  $u2$  の 1:1 の対として式 (7) に従い距離値を算出する。

入力パターンに対からはずれる楔がある場合も、同様に 2:1 の対を決定していく。

対からはずれた楔が複数ある場合、順次この処理を行うことで、 $n:m$  の対を得ることもできる。たとえば、図 7 に示した [う] の例で、辞書パターン楔 u4 の後ろに、さらに対からはずれた楔 u5 がある場合を考えると、u3 の処理で  $(v3, u3 + u4)$  が 1:2 の対と判定した後、u5 の処理で  $(v3, (u3 + u4) + u5)$  が 1:2 の対と判定されれば、1:3 の対となる。

### 2.3.3 続け書きの判定

図 9 に示すように、入力パターンが続け書きの場合、続け書き部分の楔 v2 が辞書パターンよりも多くなり、前段までの処理での楔の対からはずれてしまい、これをペナルティとして距離値に加算すると距離値が大きくなり誤認識してしまう。

そこで、続け書きにより派生した楔かどうかを、マッチング処理の第 3 ステップ (図 6: STEP2-3) で判定

し、パターン間距離値を求めなおす。

前段までの処理で対からはずれた楔について、図9(1)に示す入力パターン  $v_2$  のように前後に対ができていいる楔  $v_1$  や  $v_3$  があれば、次の2通りの解釈での距離値を算出し、距離値が小さい方を選ぶ。

- (i)  $v_2$  は対からはずれる
- (ii)  $(v_2, t_1)$  が 1:1 の対:ただし、 $t_1$  は  $u_1$  の終点と  $u_3$  の始点の間のベクトル

以上の3ステップからなる処理で楔の対を得たのち、パターン間の距離値  $D(V, U)$  として、楔の対の距離値の総和  $d(n:m$  の楔の対)に、最終的に対からはずれた入力・辞書パターンの楔の長さを加算した値  $d$  (対からはずれた楔)を用いる。

$$\begin{aligned} & \text{パターン間の距離値 } D(V, U) \\ & = d(n:m \text{ の楔の対}) + d(\text{対からはずれた楔}) \end{aligned} \quad (9)$$

### 3. 認識性能の評価実験とその結果

#### 3.1 辞書用データ

辞書に定義する辞書パターンは、あらかじめ楔文字モデルに変換し、辞書に登録しておく。本方式は、楷書書きの文字を1文字ずつ登録してあれば、続け書きも筆順違いの文字も認識できるが、いろいろな人が書く文字パターンに対応できるように、複数の字体の辞書パターンに登録してマルチプレート化を図ることで、より精度の良い認識ができる。

ここでは、筆順画数同時フリーの効果を見るために、筆順画数違いが多く筆記されると考えられる漢字について、登録パターン数を変えた2種の辞書を作成した。

- (a) 高精度版 漢字・非漢字とも:複数の字体の辞書パターンに登録。
- (b) 小型版 漢字:1パターンずつを登録、非漢字:複数の字体の辞書パターンに登録。

漢字以外の字種については、漢字と比べて1パターンあたりのデータサイズが小さく、複数字体に登録しても辞書容量があまり増えないことから、認識率を優先して、2種の辞書のどちらにも複数の字体の辞書パターンに登録した。登録する字体数は、文字ごとに異なるが、(a)高精度版では、平均して1文字あたり4字体登録している。

辞書パターンは、手書き文字認識用データベース TUAT Nakagawa Lab. HANDS- nakayosi\_t-98-02<sup>25)</sup>の nk0001 から nk0100 までの100人分と、これまで我々が辞書作成に用いてきたデータ1人分を用い、文字ごとに、101個の文字パターンをクラスタリングして各クラスタの平均パターンを辞書に登録した。ク

ラスタリングは、文字を楔文字モデルに変換した後、階層クラスタリング方式により行った<sup>26)</sup>。101個の文字パターンをそれぞれの平均パターンとする101のクラスタから処理を開始し、平均パターン間の距離値が最も小さい2クラスタを統合していくことを繰り返す。距離値は2.3節で述べたマッチング処理により算出し、統合後の平均パターンは、マッチング処理により対と判定した2つの楔の平均値を得て作成する。各クラスタの平均パターン間の距離値が、すべてしきい値  $Th\_D$  より大きくなればクラスタリングは収束したとして、処理を終了する。収束条件に用いるしきい値  $Th\_D$  は、クラスタリング前の辞書で文字認識を行い、正解であった文字データの距離値の平均を用いた。

#### 3.2 評価用データ

認識対象データは手書き文字認識用データベース kuchibue\_d-96-02 の mdb0001 から mdb0080 までの80人分を用いた。これは、日本の新聞記事を例文として、80人の筆記者から漢字、仮名、英数字、記号などのオンライン手書きパターンを採集・収録したもので、各セットは3,356字種、11,962文字パターンから構成される。各字種のパターン数内訳は、漢字5,643字、英数字116字、記号1,085字、平仮名4,492字、片仮名576字となっている。

辞書用データの nakayosi および評価用データの kuchibue は、ともに、東京農工大学中川研究室で構築されたオンライン手書き文字パターンデータベースである。

#### 3.3 認識評価

楔文字モデルによる筆順画数同時フリー文字認識方式を、2種の辞書を用いて、認識率、辞書容量、処理時間について評価した。また、認識率の比較を容易にするため、1文字だけでは大きさや形状の判別が困難な図10の58組の文字対は、互いへの誤認識は正解として認識率を計算している。

ああ; いい; うう; ええ; おお; つつ; やや; ゆゆ; よよ;  
わわ; アア; イイ; ウウ; エエ; オオ; ッツ; ヤヤ;  
ユユ; ヨヨ; ヲワ; カカ; ケケ; cC; 。oO; pP;  
sS; vV; wW; xX; zZ; エ工; カ力; トト; タタ;  
又又; ハハ; て乙; え之; リリ; へへ; 7ク; 3ろ;  
hん; ち千; t七; テ千; ノ1; ×X; +十; べべ;  
べべ; ニ二; □口口; <<; ±士士; ○O0; 一;

2Z

図10 同一パターンとする文字  
Fig.10 Identical characters.

### 3.3.1 認識率

図 11 に評価データに対する第 1 候補から第 10 候補の累積認識率を示す。第 1 候補の認識率は、(a) 高精度版 83.9%、(b) 小型版 82.2%、第 10 候補の認識率は、(a) 高精度版 95.6%、(b) 小型版 94.1%であった。

引例<sup>4)</sup>の手法による同評価データ認識率では、図 10 に示す文字の互いへの誤認識は誤認識のままカウントする条件で、第 1 候補の認識率は 77.61%、第 10 候補の認識率は 94.8%である。認識率のカウント条件の違いを避けるために第 10 候補の認識率で比較すると、従来方式では 94.8%のところ、本方式では (a) 高精度版 95.6%、(b) 小型版 94.1%であり、本方式 (a) 高精度版では 0.8%向上できた。

図 12 に、評価データのうち、漢字データ、ひらがなデータの字種別認識率を示す。

図 12 の左の図は、漢字に対する認識率を示しており、辞書パターンを各文字 1 パターンしか持たない (b) 小型版に比べ、マルチテンプレート化を図っている

(a) 高精度版のほうが、3%程度高い認識率となっている。

図 12 の右の図は、非漢字の中で最もデータ数の多いひらがなの認識率を示しているが、ひらがなの辞書は、両方とも同じくマルチテンプレート化を図っているため、(a) 高精度版も、(b) 小型版も、ほぼ同様の認識率になっている。

### 3.3.2 筆順画数誤り文字の認識例

図 1、図 2 に示した筆順画数誤りの入力文字例は、(a) 高精度版、(b) 小型版とも認識できた。また、図 13 に、漢字を 1 パターンのみ登録している (b) 小型版において、認識できた文字の例を示す。[慮] は、正しい筆順の 1 画と 2 画が入れ替わり、また、3 画と 4 画が入れ替わって続け書きされている。[務] は、正しい筆順の 4 画と 5 画が入れ替わって続け書きされ、正しくは 11 画の画数が 4 画で書かれ画数が大幅に減少している。

### 3.3.3 誤認識した文字例とその原因

図 14、15 に、(a) 高精度版、(b) 小型版とも誤認識した文字の例を示す。図から、誤認識の原因として、以下が考えられる。

#### (1) 緩いカーブ

[な] は、[な] の点とループを続け書きで筆記した S

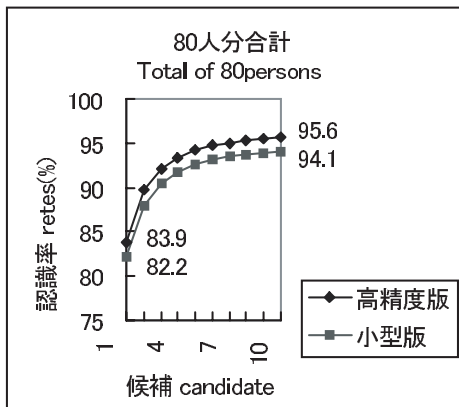


図 11 認識率  
Fig. 11 Recognition ratio.



図 13 認識できた文字の例  
Fig. 13 Correctly recognized samples.

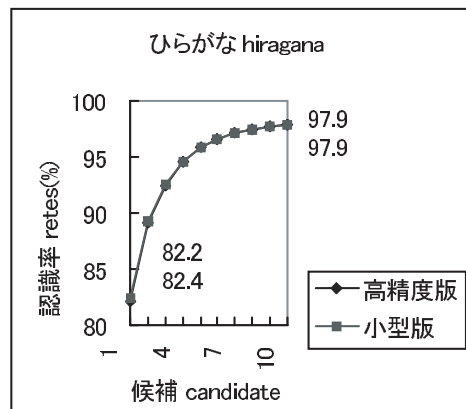
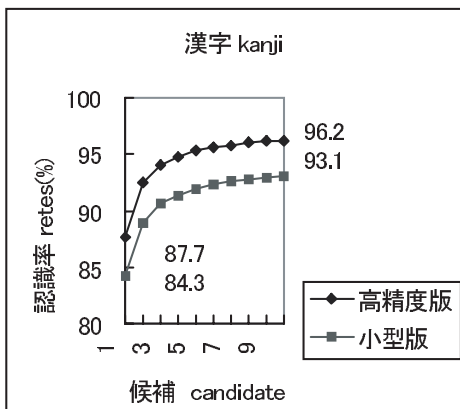


図 12 字種別認識率  
Fig. 12 Recognition ratio by character types.

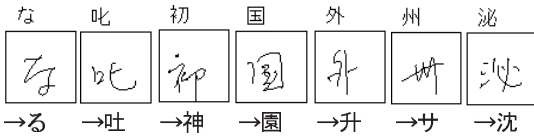


図 14 誤認識した例  
Fig. 14 Not correctly recognized samples.

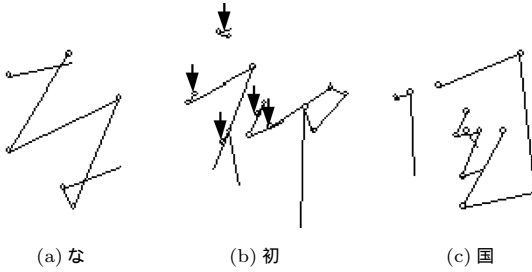


図 15 誤認識した文字の楔文字モデル

Fig. 15 Examples of cuneiform models which are not recognized as correct candidates.

型のカーブが緩く、楔文字モデルに変換した結果、図 15 (a) に示すパターンとなり、[る] に誤認識した。

(2) 筆方向違い

[叱] は、横棒の向きが右方向に筆記されているが、辞書パターンには左方向のみが登録されていた。式 (7) に表したように、楔間の距離値  $d(v, u)$  は、始点・終点間の距離、および、方向ベクトルの距離の和を用いるため、筆方向が左右逆転した場合は、距離値が大きくなり、誤認識の原因となる。

(3) ペンはねなどのノイズ

[初] は、楔文字モデルに変換した結果、図 15 (b) に示すパターンとなり、図中に矢印で示したペンはねのノイズから楔数が多くなり、正しい辞書パターンがマッチング対象外となった。また逆に、[国] は、楔として抽出したい横棒や点を、ノイズとして省いてしまい、図 15 (c) のように続け書き部分のみが残っている。

(4) 切り書きへの誤認識

[外 升] は、[升] の横棒を切り書きしたものと誤認識した。

(5) 楔の対の決定ミス

[州] は、短い 3 点を一続きに筆記しているため、図 5 (P2-1) の距離和最小の 1 : 1 の楔の対を決定できなかった。また、[泌] は、辞書パターンとくらべて点の位置が異なることから、距離値が大きくなり、マッチングによる楔の対の決定が、人が判断するのとは違っている。

3.4 楔数の分布

楔文字モデルによれば、同じ文字はほぼ等しい楔数

になると考え、マッチングの対象とする辞書パターンを、楔数が入力パターンの楔数  $\pm N$  以内のものに絞ることで、処理時間を短縮している。この絞込みが妥当かどうかを調べるため、入力・辞書パターンの楔数の分布を調べた。

入力・辞書パターンの楔数の関係を、図 16, 図 17 に示す。続け書きにより、入力パターンの楔数は増加すると考えられるため、続け書きでないデータを図 16 に、続け書きしたデータを図 17 に、楔の減少数 (辞書楔数 - 入力楔数) の頻度分布を示す。認識辞書は (a) 高精度版を用い、マルチテンプレート化により辞書パターンが複数登録されている文字は、入力パターンとの距離値が最小の辞書パターンを選んでいる。また、辞書画数より入力画数が大きい切り書きデータ、および、小文字や一部の辞書パターンがない文字データは省いている。データは、評価に用いた 80 人分データの中から比較的楷書の多い mdb0006 (続け書き文字率 24%) と、続け書きの多い mdb0001 (続け書き文字率 46%) の 2 人分について調べた。

続け書きでないデータについては、2 人どちらのデータでも、楔の減少数 = 0 をピークに  $\pm 4$  の範囲に分布している。mdb0006 は、やや入力楔数が多い傾向が見られるが、これは、辞書パターンにはペンはねなどのノイズが少ないものを登録しているため、これと比較して、入力パターンにははねによる楔数の増加が多かったためと考えられる。

続け書きしたデータについては、図 16 に、続け書き回数 (辞書画数 - 入力画数) 別に楔の減少数 (辞書楔数 - 入力楔数) を示しているが、mdb0001 は、楔の減少数 = 0 をピークに分布しており、mdb0006 は、楔の減少数 = -1 をピークに入力楔数が多い傾向にある。続け書きによって入力楔数はやや増加し、これは、入力パターンの続け部分に、辞書パターンにない楔が発生するためと考えられる。

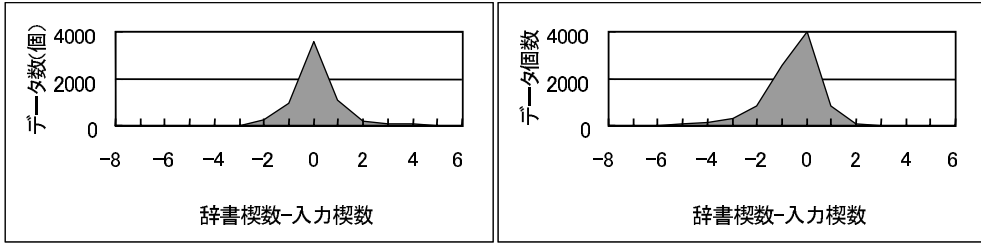
また、続け書きのあるなしを総合して、同じ文字は  $\pm N$  の楔数に分布しており、楔数によるマッチング対象とする辞書パターンの絞込みは妥当と考えられる。

3.5 辞書容量と処理時間

図 18 に本評価に用いた辞書容量と 1 文字平均処理時間 (Pentium 300 MHz 使用) を示す。辞書容量は、漢字・非漢字についてマルチテンプレート化を図った (a) 高精度版は、371 Kbyte、非漢字のみマルチテンプレート化を図った (b) 小型版はさらに小さく、189 Kbyte であった。

1 文字平均処理時間は、経験から、0.6 sec 以下であれば実用上問題がないと考えているが、評価に用いた



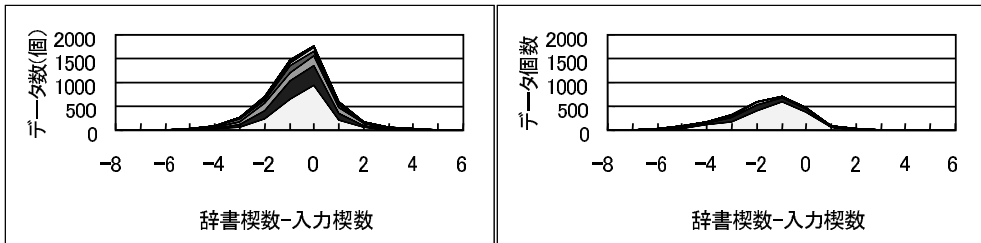


(1) mdb0001

(2) mdb0006

図 16 楔差分の分布：続け書き無し

Fig. 16 Distribution of no. of kusabi: Non-cursive characters.



(1) mdb0001

(2) mdb0006

図 17 楔差分の分布：続け書き有り

Fig. 17 Distribution of no. of kusabi: Cursive characters.

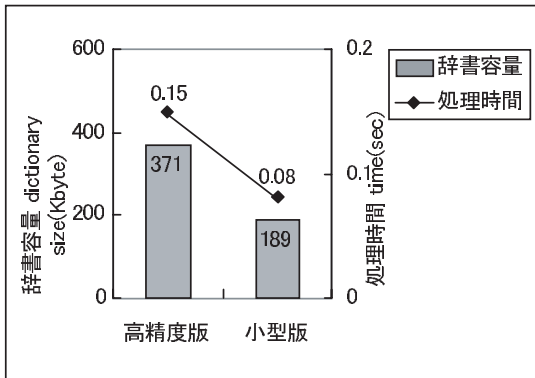


図 18 辞書容量と処理時間の評価

Fig. 18 Size of dictionary and process time.

Pentium 300 MHz では、(a) 高精度版でも 0.15 sec と実時間で処理ができています。適用システムによっては、CPU クロック数を低く抑える必要があるが、(b) 小型版は、0.08 sec であるので、約 1/7 倍の CPU クロック数でも実時間で処理ができると推測できる。

#### 4. 今後の課題

楔文字モデルによる文字認識方式により、筆順違いで、かつ、続け書き文字を認識できた。しかし、3.2.3 項に述べた誤認識の原因から、今後対策すべき点につ

いて述べる。

楔文字モデルへの変換処理（折れ線近似）は、続け書きを分断できる細かさで、しかも、筆順に依存しないマッチング処理を実時間で処理できる大まかさで近似しているが、今後、CPU 性能が向上してくれば、誤認識原因（1）の緩いカーブを近似しきれない問題を改善するために、高さの位置に近似点を置ききい値  $Th_h$ 、統合する楔の角度差しきい値  $Th_\theta$  を小さくすることで、筆順を入れ替える楔線分をさらに細かく近似し、また、楔の対の決定方式を現在の簡便な方式より正確に最適解を得られる方式にすることが考えられる。また（3）ペンはねなどのノイズについても、マッチングの対象とする辞書パターンを楔数から絞る範囲を広げることで対応できる。

マッチング処理に関しては（4）切り書きへの誤認識を対策するために、切り書きに対して、ペナルティを付加することが考えられる（2）筆方向違いは、マッチング処理で対応することも可能であるが、1:1 の楔間の距離値を求めるためには 4 倍の処理時間がかかることから、間違いやすい筆方向違い文字を辞書に登録しておく対応のほうが得策と考える。

また（5）楔の対の決定ミスについては、対の決定方式を工夫することも考えられるが、図 14[州] のよう

に、省略されたり過度の変形が生じる場合は、辞書パターンと入力パターンの楔間の距離が大きくなり対の決定ができない。辞書パターンを各文字1パターンしかもたない (b) 小型版に比べ、マルチプレート化を図っている (a) 高精度版のほうが、第10候補の認識率も高いことから、本方式で辞書を1パターンしか持たずに認識するには限界があり、変形パターンについては辞書に登録しておく必要がある。

## 5. おわりに

ペン入力コンピュータにおいて快適なユーザインタフェースを提供するため、続け書きの文字や筆順の間違いにも対応できる、楔文字モデルによる文字認識方式を開発した。基本方式として、①入力の続け書きを分断するように、文字を構成する線分である楔に分割し、②筆順に依存せずに楔の分割数の違いや続け書きによる楔の数の増加を考慮した距離値の算出を行う。

文字認識用データベース kuchibue\_d-96-02 を用いて評価した結果、本方式の高精度版では第1候補 83.9%、また、辞書を小型化した小型版では 82.2%と、実用の域にある認識結果を得ることができ、また、続け書きと筆順違いの文字認識への有効性を確認できた。

今後は、楔文字モデルへの変換において、緩いカーブ形状を考慮した方式に改善すること、また、マッチングの対象とする辞書を絞り込む方式を改善することで、認識率の向上と処理時間の短縮を図りたいと考えている。

謝辞 本研究にあたり、東京農工大工学部中川正樹教授および関係各位よりご指導いただきましたことを感謝いたします。

## 参考文献

- 1) 中野康明：文字認識・文書理解の最新動向 [III] オンライン手書き文字認識，信学誌，Vol.83, No.3 pp.228-231 (2000).
- 2) 葛貫壮四郎，横田登志美，郡司圭子，三浦正樹，桂晃 洋，石田正幸：一筆続け書き文字に対応可能な文字認識アルゴリズムの開発—より柔軟性のある文字認識を目指して(その2)，第55回情報処理学会全国大会，pp.2-190-191 (Sep. 1997).
- 3) 秋山勝彦，中川正樹：ストロークのつながりに寛容なオンライン手書き文字認識，画像の認識・理解シンポジウム MIRU'94 (1994).
- 4) 秋山勝彦，中川正樹：オンライン手書き日本語文字認識のための線形処理時間伸縮マッチングアルゴリズム，信学論(D)，Vol.J81-D-II, No.4, pp.651-659 (1998).
- 5) Kim, H.J. and Kim, P.K.: On-line recogni-

tion of cursive Korean characters using a set of extended primitive strokes and fuzzy functions, 1996 Elsevier Science B.V., Vol.17, No.1 (1996).

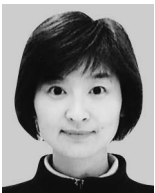
- 6) 小高和巳，若原 徹，増田 功：筆順に依存しないオンライン手書き文字認識アルゴリズム，信学論(D)，Vol.J65-D, No.6, pp.679-686 (1982).
- 7) 迫江博昭，慎 重弼：画順サーチによる筆順フリーなオンライン文字認識の検討，信学技報告 PRU95-59, pp.55-59 (July 1995).
- 8) 若原 徹，小高和巳，梅田三千雄：選択的ストローク結合による画数—筆順に依存しないオンライン文字認識，信学論(D)，Vol.J66-D, No.5, pp.593-600 (1983).
- 9) 広瀬斉志，田坂 滋，森田敏昭，堀井正博，猪田嗣朗：オンライン手書き漢字認識装置，シャープ技報，Vol.28, pp.69-74 (1984).
- 10) 若原 徹，梅田三千雄：ストローク結合規則を用いたオンラインくずし字分類，信学論(D)，Vol.J67-D, No.11, pp.1285-1292 (1984).
- 11) 正嶋 博，横山孝典，葛貫壮四郎，福永 泰：画数，筆順，回転，区切りに依存しないオンライン手書図形認識方式，情報処理学会論文誌，Vol.27, No.5, pp.492-498 (1986).
- 12) 村瀬 洋，若原 徹，梅田三千雄：接続ルールを導入した候補ラティス法によるオンライン手書き線図形認識，信学論，Vol.J67-D, No.3, pp.273-280 (1984).
- 13) 御船登志美，正嶋 博，葛貫壮四郎，福永 泰：オンライン手書き入力システム(2)—線分セグメント法オンライン手書き図形認識の評価，第31回情報処理学会全国大会。
- 14) Liu, Y.-J., Zhang, L.-Q. and Tai, J.: A New Approach to On-line Handwritten Chinese Character Recognition, *IEEE/2nd IC-DAR*, pp.192-195 (Oct. 1993).
- 15) Okamoto, M., Nakamura, A. and Yamamoto, K.: Direction-Change Features of Imaginary Strokes for On-line Handwriting Character Recognition, *Proc. 14th ICPR*, pp.1747-1751 (Aug. 1998).
- 16) Kawamura, A., Yura, K., Hayama, T., Minamikawa, T., Tanaka, A. and Masuda, S.: On-line recognition of freely handwritten Japanese characters using directional feature densities, *Proc. 11th ICPR*, II, pp.183-186 (Aug. 1992).
- 17) Hamanaka, M., Yamada, K. and Tsukumo, J.: On-line Japanese character recognition experiments by an off-line method, *Proc. 2nd ICDAR*, pp.204-207 (Oct. 1993).
- 18) 加藤 真，イッシャイ カモン：オンライン文字認識における文字検証の一手法，信学論(D-II)，Vol.J76-D-II, No.11, pp.2295-2303 (Nov. 1993).
- 19) 比田井，大井，中村，黒沢：パーソナルコンピュ

ータによるオンライン手書き漢字認識システム, 昭和 60 年信学総全大, No.1528 (1985).

- 20) 横田登志美, 葛貫壮四郎, 郡司圭子, 三浦雅樹, 桂晃 洋: 筆順画数フリー文字認識の開発—より柔軟性のある文字認識を目指して(その1), 第 55 回情報処理学会全国大会, pp.2-188-189 (Sep. 1997).
- 21) C. ウォーカー(著), 大城光正(訳): 大英博物館双書: 失われた文字を読む 1: 楔形文字(株)学芸書林, 東京(1995).
- 22) 石垣一司, 森下哲次: 誤読の少ない続け文字認識(特徴点逐次対応法), 第 33 回情報処理学会全国大会, pp.1633-1634 (Oct. 1986).
- 23) Ishigaki, K. and Morishita, T.: A top-down online handwritten character recognition method via the denotation of variation, *Proc. 1988 Int'l Conf. on Computer Processing on Chinese and Oriental Languages*, Toronto, pp.141-145 (Aug. 1988).
- 24) 奥村晴彦: C 言語による最新アルゴリズム辞典, 株式会社技術評論社, 東京(1991).
- 25) 中川正樹: 文章形式字体制限なしオンライン手書き文字パターンの収集と利用, 信学技報 PRU95-110, pp.43-48 (Sep. 1995).
- 26) 横田登志美, 葛貫壮四郎, 郡司圭子, 浜田長晴, 桂晃 洋: 特徴点近似オンライン文字に適用可能なクラスタリング手法及びパターン平均化手法の開発と評価, 信学論(D-II), Vol.J85-D-II, No.6, pp.997-1005 (2002).

(平成 13 年 5 月 31 日受付)

(平成 15 年 1 月 7 日採録)



横田登志美(正会員)

昭和 59 年茨城大学工学部情報工学科卒業。同年(株)日立製作所入社。日立研究所にて, オンライン文字認識・図形認識, ペン入力インタフェースの研究に従事。電子情報通信学会会員。

信学会会員。



葛貫壮四郎(正会員)

昭和 53 年茨城大学短期大学部電気学科卒業。平成 13 年放送大学卒業。昭和 37 年(株)日立製作所入社。日立研究所にて, 電力システムの安定度向上の研究, エレベータ群制御

の研究, ペン入力インタフェースの研究, 車載情報システムの研究に従事。



郡司 圭子(正会員)

平成 4 年東京理科大学理工学部情報科学科卒業。同年(株)日立製作所入社。日立研究所にて, ペン入力インタフェース, 日本語処理, 現在, 電力情報制御システムの研究に従事。

電子情報通信学会会員。



桂 晃洋(正会員)

昭和 52 年京都大学工学部電気工学科卒業。昭和 54 年同大学院修士課程修了。同年(株)日立製作所入社。マイクロコンピュータ, グラフィックス, LSI, ヒューマンインタフェースの研究に従事。工学博士。電気学会, IEEE, ACM 各会員。



浜田 長晴(正会員)

昭和 40 年鹿児島大学工学部電気工学科卒業。同年(株)日立製作所入社。日立研究所にて, ディスプレイ, グラフィックス, FAX, イメージ処理, マルチメディアシステム等

の研究に従事。平成 13 年より倉敷芸術科学大学教授。工学博士。電子情報通信学会, 画像電子学会, IEEE, SID 各会員。



福永 泰(正会員)

昭和 48 年京都大学工学部電気工学科卒業。昭和 50 年同大学院修士課程修了。同年(株)日立製作所入社。同研究所にて, ワークステーション, グラフィックス, ペン入力インター

フェース, ITS の研究に従事。電子情報通信, 電気学会各会員。