

# 枠なし手書き文字列認識における誤認識訂正インタフェース

坂 東 宏 和<sup>†</sup> 福 島 貴 弘<sup>†</sup>  
加 藤 直 樹<sup>†</sup> 中 川 正 樹<sup>†</sup>

本論文は、表示一体型タブレット上での文字記入枠を用いない枠なし手書き認識方式における誤認識の訂正インタフェース、および、認識結果の表示インタフェースを提案する。枠なしの手書き認識方式は、筆者に自然で円滑な手書き入力を提供する反面、文字の誤認識は枠あり認識より起こりやすく、文字間の区切り判定でも誤りが発生する危険がある。そこで本論文では、各文字の誤認識訂正方法、文字区切り位置の訂正方法、および、認識結果の表示方法を複数提案し、それらの比較実験について報告する。実験の結果、各文字の誤認識訂正は訂正候補を手書き文字の上にメニュー表示し選択させる方法が適切であること、文字の分割操作は区切りジェスチャ、文字の結合操作は文字が小さく書かれている場合には囲みジェスチャ、文字が大きい場合には接続ジェスチャが適切であることが示された。また、認識結果の表示は、文字が小さいなどの理由により読みにくくなる場合には離れた場所にまとめて表示する方法が、結果と手書き文字が読みにくくならなければ手書き文字の上に認識結果を重ねる方法が適切であることが示された。

## User Interfaces for Correcting Errors in Writing-box-free Recognition of Handwritten Text

HIROKAZU BANDO<sup>†</sup>, TAKAHIRO FUKUSHIMA<sup>†</sup>, NAOKI KATO<sup>†</sup>  
and MASAKI NAKAGAWA<sup>†</sup>

This paper presents user interfaces for writing-box-free handwriting input on a display-integrated tablet. The writing-box-free handwriting recognition is natural and smooth to input text, but character recognition errors may happen more frequently than the writing-boxed recognition and character segmentation errors may also occur. Therefore, this paper proposes various user interfaces for correcting character recognition and segmentation errors as well as those for displaying recognition results, and then compares them. In consequence of comparative experiments, the followings are suggested: for changing character recognition results, showing recognition candidates above each handwritten character pattern and accepting selection is efficient; for correcting mis-divided character segmentation into one, cutting pen-gesture is superior; for correcting mis-combined character patterns into two or more, encircling pen-gesture is preferred when small characters are written densely, while connecting pen-gesture is effective when large characters are written sparsely; for displaying character recognition results, showing resulting text apart from handwritten patterns is preferred when characters are written small and densely, while overlaying the recognition result on top of each handwritten pattern is preferred when characters are written large and sparsely.

### 1. はじめに

急速な技術の進歩による計算機の高速度と低価格化により、各家庭に計算機が浸透し、そのユーザ層も急速に拡大している。従来の計算機は使いやすさよりも速さが優先されてきたが、近年のこのような状況の中で、速さよりも使いやすさが重視されるようになって

きた。

タブレットと電子ペン（スタイラス）による計算機への入力は、キーボードやマウスのように習熟を必要としない、まさに紙と鉛筆のように使える手軽なユーザインタフェース（以下インタフェースと記す）として研究が進められている。電子ペンによるテキスト入力の研究としては、仮想キーボード<sup>1),2)</sup>や手書き文字入力<sup>3),4)</sup>がある。手書きで文字を書くことは、慣れ親しんだ動作であるので初心者にも入力しやすいという利点がある。しかし、手書き文字認識における誤認識の発生や漢字入力時の画数の多さなどから、キーボー

<sup>†</sup> 東京農工大学工学部

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

ドよりも入力効率が劣るという問題もある。そこで、それを補う方法として、すでに入力された文字列からユーザが入力したい文字列を予測し、それをユーザに提示することによって入力の効率を向上させる研究<sup>5)</sup>などが行われている。

また、従来の手書きによる文字入力では、記入枠を設け、その枠の中に1文字ずつ文字を記入する方式が一般的である(枠あり手書き文字認識、以下枠あり認識と記す)。この方式の場合、記入枠のない状態よりも誤認識は減少するが、筆記者に対し枠の中に書かなければならないという制約を与えることになる。

そこで、筆者の所属する研究室では、この制約をなくし、筆記者が自由な筆記スタイルで手書き文字入力を行えることを目的とした、記入枠のない環境で書かれた手書き文字列を認識する技術(枠なし手書き文字列認識、以下枠なし認識と記す)を研究している<sup>6)</sup>。同様の研究としては、仙田らが行った切り出し・認識・言語の確信度を統合した枠なし文字認識手法の研究<sup>7)</sup>がある。自由な筆記スタイルで手書き文字を入力できるようにすることで、筆記者は紙と鉛筆による筆記と同様に、特別な意識を払うことなく自然で円滑な文字入力を行うことが可能になると期待される。

しかし、現状では手書き文字列を認識するときには枠あり認識以上に誤認識が発生し、その誤認識の訂正作業が必要になる。したがって、円滑な文字入力を実現するためには、認識率の向上に加え、簡便に誤認識を訂正できるインタフェースの実現が不可欠である。

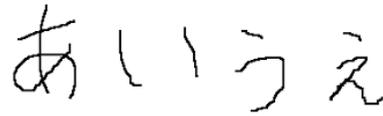
手書き文字認識を用いた日本語の文章入力の研究としては、荒井らの行った文字入力方法の研究<sup>8)</sup>がある。この研究では、手書きによる文字入力とテキスト編集が繰り返される場合の操作性について検討している。また、中島らの行った研究<sup>9)</sup>では、枠あり認識を用いたFEPのインタフェースを検討している。しかし、これらの研究は枠あり認識を対象としたものであり、区切り位置の誤認識訂正インタフェースなど、枠なし認識特有の問題については言及していない。

文章の誤入力の訂正方法としては、Huerstらが行った研究がある<sup>10)</sup>。この研究では、手書き文字認識エンジンの修正をほとんど行わずに文章の挿入、削除、上書きなどの誤入力訂正を実現する方法を提案している。しかし、具体的な誤認識訂正インタフェースについては言及していない。

そこで本論文では、枠なし認識における誤認識を簡便に訂正し、円滑な文字列入力を行えるインタフェースの基本設計を明らかにすることを目的とし、枠なし認識における誤認識の訂正インタフェース、および、



(1)枠あり認識



(2)枠なし認識

図1 枠あり認識と枠なし認識の典型的な入力画面

Fig. 1 Handwriting input forms with and without writing boxes.

認識結果の表示インタフェースを提案する。さらに、それらのインタフェースの比較評価を行った結果について報告する。

本論文の2章で枠なし認識における誤認識訂正インタフェースおよび認識結果表示インタフェースについて検討し、3章でそれらのインタフェースを実装した簡易文書入力ツールの実現を行い、4章で比較評価とその結果を報告し、5章で結論を述べる。

## 2. 誤認識訂正および認識結果表示インタフェースの設計

本章では、枠なし認識における誤認識訂正インタフェースおよび認識結果表示インタフェースの設計について述べる。

### 2.1 枠あり認識と枠なし認識の違い

枠あり認識と枠なし認識の典型的な入力画面を図1に示す。枠あり認識は、基本的に1文字ごとに認識を行うため、誤認識の種類は、各文字の認識結果が違う場合だけである。したがって、誤認識訂正インタフェースは、各文字の認識結果を訂正する方法だけを提供すればよい。それに対して、枠なし認識は文字列を一括して認識するため、次の誤認識の可能性が考えられる。

- (1) 各文字の認識結果が違う。
- (2) 文字の区切り位置が違う。

したがって誤認識の訂正インタフェースとしては、各文字の認識結果の訂正に加え、文字の区切り位置を訂正するインタフェースも必要となる。

### 2.2 各文字の誤認識訂正

各文字の誤認識訂正は、文字認識エンジンの返す第2位以下の候補(以下訂正候補と記す)の中から選択する方法と、訂正候補に正解文字が含まれていない場合に、その文字だけを書き直す方法の2種類用意する必要がある。

訂正候補の中から選択する場合のインタフェースと



(1) 訂正候補一覧を手書き文字から離れた場所に表示



(2) タップすることで次の候補に変更



(3) メニュー形式で訂正候補一覧を表示

図 2 誤認識訂正方法

Fig. 2 Methods of correcting recognition errors.

しては、次の方法が考えられる(図2)。

- (1) 枠内をタップすると訂正候補一覧が手書き文字から離れた場所に表示され、その一覧から選択する方法。
- (2) 枠内をタップすることで次の訂正候補に変更される方法。
- (3) 枠内をタップすると訂正候補一覧が手書き文字の上にメニュー形式で表示され、その一覧から選択する方法。

訂正候補を離れた場所に表示する方法(1)の場合、訂正のために視線が手書き文字と訂正候補間を行き来することになり、ユーザが疲れる可能性がある。一方、方法(2)の場合、正解文字が訂正候補の上位に入っている場合には素早く訂正できるが、下位に入っている場合や訂正候補に入っていなかった場合に何回もタップする必要がある。また、方法(3)は、正解文字が何番目に入っていたとしても、タップ回数はメニューの表示と候補選択の2回となるが、正解候補が上位に入る確率が高い場合には、方法(2)よりも手間がかかる。そこで本論文では、方法(1)~方法(3)を試作し、評価実験による比較を行う。

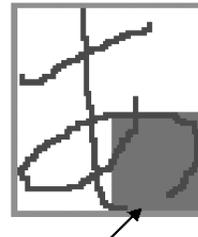
また、図3に示すように、訂正候補に正解文字が含まれていなかった場合については、枠内から新たに手書き文字を書き直すことで、その文字だけ修正できるようにする。

なお、ペン入力環境でタッチインタフェースを用いた場合には、マウスの場合と異なりペンダウンしてからペンアップするまでの間にペン先が移動してしまう



図 3 文字の再入力方法

Fig. 3 Methods of re-inputting character.



候補選択領域

図 4 候補選択領域

Fig. 4 Tapping region to select a correct answer.

ことが多く、ユーザがタップのつもりで入力したにもかかわらず手書き文字の再入力として処理されてしまう危険性がある。そこで、図4のように、認識結果の右下に特別な領域を表示し、その中での入力を無条件でタップと見なす(以下この領域を候補選択領域と記す)。

### 2.3 文字区切り位置の訂正

文字区切り位置の訂正は、次の2通りの場合がある。

- (1) 文字の分割: たとえば「女子」を1文字として認識し「好」となったときに、それを「女」と「子」に分割する場合。
- (2) 文字の結合: たとえば「晴」を「日」と「青」に分割して認識されたときに、それを1文字であると指示し「晴」に訂正する場合。

分割位置の訂正方法としては、囲みジェスチャ<sup>11),12)</sup>(図5(a),(b)),区切りジェスチャ<sup>11)</sup>(図5(c)),接続ジェスチャ(図5(d))が考えられる。

囲みジェスチャは場合(1),(2)の両方に対応できる。一方区切りジェスチャは場合(1),接続ジェスチャは、場合(2)にしか対応できないが、囲みジェスチャよりも入力すべきストローク長が短い。そこで本論文では、囲みジェスチャ、区切りジェスチャ、接続ジェスチャについて、文字の分割を行う場合と結合を行う場合それぞれについて比較評価を行う。

### 2.4 誤認識の訂正範囲

誤認識を訂正する場合に、ユーザはランダムな順番で修正していくのではなく文章の先頭から順番に修正していくと考えられる。また、枠なし認識では、誤認



(a) 囲みジェスチャによる文字の分割



(b) 囲みジェスチャによる文字の結合



(c) 区切りジェスチャによる文字の分割



(d) 接続ジェスチャによる文字の結合

区切り位置の間違い(左側)を,ジェスチャ(中央)により訂正できる(右側).

図5 区切り誤りの訂正方法

Fig. 5 Methods of correcting segmentation errors.

識の訂正が行われたときに文脈処理をやり直すことで, 1カ所訂正するだけで複数個所の誤認識を自動的に訂正できる可能性がある.そこで,2.2節と2.3節で述べた訂正が行われたときに,図6のように訂正部分より前の文字認識は正しいと仮定し,訂正文字以降の手書き文章を一括して再認識することで,1回の訂正で複数の誤認識を訂正する.なお,一度訂正された部分については,再認識の対象としない.

### 2.5 枠なし認識における認識結果の表示

枠あり認識における認識結果の表示方法から考察すると,各文字の認識結果の表示は,次の方法が考えられる(図7).

- (1) 手書き文字を消去し,その場所に認識結果を表示する方法.
- (2) 手書き文字から離れた位置に,まとめて認識結果を表示する方法.
- (3) 手書き文字を表示したまま,その上に認識結果を表示する方法.

手書き文字を消去してしまう方法(1)の場合,誤認識が発生した場合に,ユーザが何の文字を書いたのか

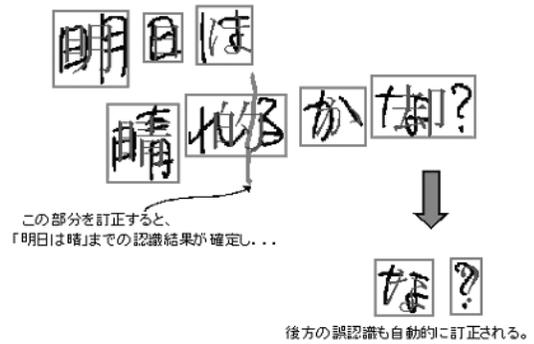


図6 誤認識の文脈による自動訂正

Fig. 6 Automatic correction of recognition errors according to context.



図7 認識結果の表示方法

Fig. 7 Methods of displaying results.

を忘れてしまう危険性がある.また,文字の区切り位置が間違っ認識された場合,正しい区切り位置を示すために手書き文字を再表示する操作が必要になり面倒である.

Microsoft社のIMEパッドなどで採用されている方法(2)は,確認や訂正のために,視線が手書き文字と認識結果間を行き来することになり,ユーザが疲れる危険性がある.一方,方法(3)は,視線が行き来するという欠点は改善されるが,重ねて表示するために,認識結果や手書き文字が見にくくなる危険性がある.そこで本論文では,方法(2)および方法(3)を試作し,評価実験による比較を行う.

なお,促音や拗音などの小さい文字と通常の文字との判別を容易にするために,図8のように小さい文字と通常の文字の色を変えて表示するものとする.

枠なし認識では,各文字の認識結果に加え,文字の区切り位置を示すことも必要である.区切り位置を示す方法としては,次の方法が考えられる(図9).

- (1) 手書き文字を,認識結果1文字分ごとに色を変えて表示する.
- (2) 文字の外接枠を表示する.

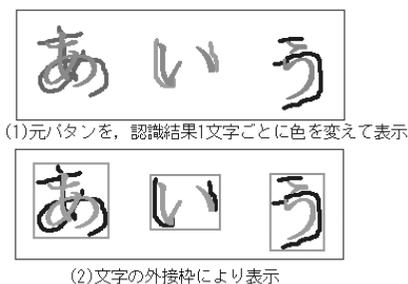
色を変えて表示する方法(1)の場合,各文字の誤認識を訂正するときや接続ジェスチャを行うときに,ど



通常の文字(上側)と促音や拗音などの小さい文字(下側)を、色を変えて表示する。

図8 小さい文字の表示方法

Fig. 8 Methods of displaying recognition results for small characters.



(1)元パターンを、認識結果1文字ごとに色を変えて表示

(2)文字の外接枠により表示

図9 区切り位置の表示方法

Fig. 9 Methods of displaying segmentation.

の部分から入力を始めればよいか分かりにくい。また、大画面に表示するためにプロジェクタで投影して用いる場合や、モノクロ表示の場合には、一部の色が見にくくなる可能性がある。そこで本論文では、方法(2)を採用する。

### 3. 簡易文章入力ツールの実現

2章で述べた誤認識訂正および認識結果表示インタフェースの比較評価を行うために、2章で提案した複数のインタフェースの中から、任意の組合せを選択して入力できる、簡易文章入力ツールを実現した。本ツールを利用すると、任意のテキストエディタやワープロソフト、その他テキストの入力が可能な部分に、手書きで文章を入力できる。

本ツールを起動した後に、文書入力したい部分でダブルタップを行うと、入力画面が起動する。ユーザは任意の文章を手書きで入力する。文章は複数行にわたって書くことができる。また入力スペースが不足した場合には、スクロールさせることも可能である。書き間違えた場合には、最後に書いたストロークから逆順に1画ずつ取り消すことができる。

「認識」ボタンを押すと、手書きで書かれた文字列

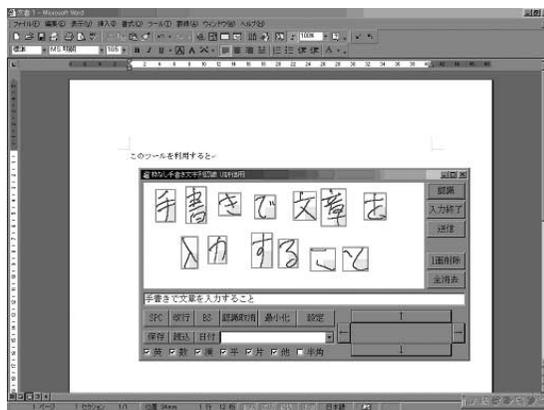


図10 簡易文章入力ツール

Fig. 10 Tentative application of text inputting.

本稿では枠のない入力面に筆記される手書き文字列を認識する際に発生する誤認識を訂正するためのユーザインタフェースの設計とその試作について述べる。

図11 訂正時間の測定に用いた文章

Fig. 11 Text to evaluate the time required for correction.

を認識し、図10のように認識結果を表示する。認識結果が正しい場合には、「送信」ボタンを押すと、先に指定したアプリケーションに文字列が入力される。

認識結果の表示方法や誤認識が発生した場合の訂正方法は、ユーザが任意の方法を設定、選択できる。

### 4. 誤認識訂正および認識結果表示インタフェースの比較評価

2.2節、2.3節で提案した誤認識訂正インタフェースと2.5節で提案した認識結果の表示方法について、それぞれの方法が適切かを評価するために、3章で述べたツールを用いた比較評価実験を行った。

#### 4.1 実験方針

各誤認識訂正インタフェースの比較は、意図的に含ませた7カ所の誤認識を訂正するのにかかる時間の測定結果と、実験後に行ったアンケートによる主観評価の結果により行う。認識結果表示インタフェースについては、アンケートによる主観評価のみ行う。

訂正時間の測定に用いた文章を図11に示す。なお、測定に用いた文章は1通りであり、すべての実験・被験者において同じである。ただし、誤認識の場所は実験を繰り返すたびに異なる。

#### 4.2 実験環境

実験環境を表1に、実験の様子を図12に示す。被

表 1 実験環境

Table 1 An environment for experiments.

計算機	PC/AT 互換機(ノートパソコン) Pentium III 600Mhz. メモリ 192MB
OS	Microsoft Windows Me
表示一体型タブレット	MUTOH Video Tablet MVT-14 画面サイズ: 285.696×214.272mm. (1024×768 ピクセル)
簡易文章入力ツールの 筆記面のサイズ	261.423×171.864mm (937×616 ピクセル)
実験に使用した手書き 文字のサイズ	261.423×119.97mm(937×430 ピクセル)の範囲に 71 文字 平均的な文字のサイズ: 10.044×11.718mm(36×42 ピクセル)



図 12 評価実験の様子

Fig. 12 A scene of experiment.

験者は筆者の所属する研究室の学生・職員あわせて 10 名であり、全員コンピュータ、および、電子ペンの操作に習熟していた。

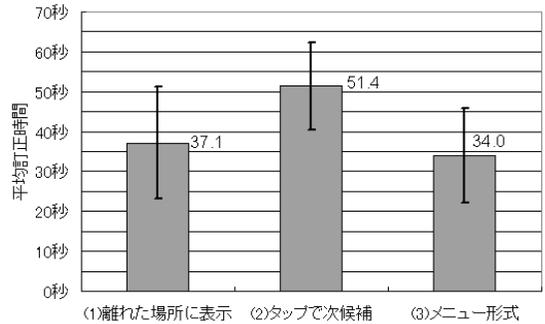
#### 4.3 実験手順

評価実験の前に簡易文章入力ツールの簡単な説明を行い、ツールの基本的な使い方に慣れてもらうために、各被験者に 15 分程度自由に利用してもらった。ただし、認識結果の表示方法については、4.6 節で後述する主観評価の関係から、すべての表示方法を試してもらった。

また、誤認識訂正時間の測定は、次のような手順で行った。

- (1) 正しい認識結果を被験者に示す。
- (2) 7カ所の誤認識を意図的に加え、被験者にその位置を憶えてもらう。
- (3) 指定した訂正インタフェースだけを用いて文章を訂正してもらい、その訂正時間を測定する。

以上の測定をすべてのインタフェースについて行った。時間測定中は各文字の再入力機能と、訂正文字以降を自動的に訂正する機能を無効とした。また、時間測定中の認識結果表示は、手書き文字から離れた位置にまとめて認識結果を表示する方法と、手書き文字を



縦棒は平均訂正時間、線分は平均訂正時間 ± 標準偏差の幅を示す。

図 13 文字の誤認識を訂正するための所要時間の比較評価

Fig. 13 Comparative evaluation on the time required for correcting character recognition errors.

表示したままその上に認識結果を表示する方法の両方を併用した。

また、順序効果をなくすために、各インタフェースの実験順を変えて被験者ごとに比較対象数だけ繰り返した。たとえば比較対象のインタフェースが 3 種類であった場合、インタフェース 1, 2, 3 と測定した後、誤認識の位置を変えてインタフェース 2, 3, 1 と測定し、さらに誤認識の位置を変えてインタフェース 3, 1, 2 の順番で測定する。

すべての誤認識訂正時間の測定が終了した後、被験者に簡単なアンケートに回答してもらった。

#### 4.4 各文字の誤認識訂正インタフェースの比較

##### 4.4.1 実験目的

文字の認識結果が違っていた場合の訂正インタフェースとして、2.2 節で提案した次の訂正インタフェースの中でどのインタフェースが一番使いやすいかを調べるために、比較評価実験を行った。

- (1) 枠内をタップすると訂正候補一覧が手書き文字から離れた場所に表示され、その一覧から選択する方法(離れた場所に表示)。
- (2) 枠内をタップすることで次の訂正候補に変更される方法(タップで次候補)。
- (3) 枠内をタップすると訂正候補一覧が手書き文字の上にメニュー形式で表示され、その一覧から選択する方法(メニュー形式)。

##### 4.4.2 実験結果

各訂正インタフェースを用いて、7カ所の誤認識を訂正した場合の平均訂正時間を図 13 に示す。図 13 は、平均時間を縦棒、平均値 ± 標準偏差の幅を線分で示している。また、危険率 5% の t 検定を行い、各項目間の差を検討した結果、項目 (1) と項目 (2)、および、項目 (2) と項目 (3) の間で有意差が認められた。

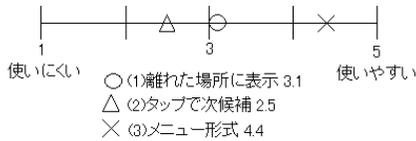


図 14 文字の誤認識訂正に対する主観評価

Fig. 14 Subjective evaluation on correcting character recognition errors.

各訂正インタフェースの使いやすさを 5 段階で評価してもらった結果を図 14 に示す。また、自由に意見を記述してもらった結果、次の意見が得られた。

- タップすると次候補に変更される方法は、前の候補に逆戻りできないと不便。
- 視線とペンの移動は少ない方が使いやすい。

#### 4.4.3 実験の考察

実験の結果から、タップすると次候補に変更される方法は、他の 2 つの方法に比べて訂正時間が多くかかることが示された。主観的な評価も同様に、やや使いにくいと評価されている。これは、正解候補を見つけるまで何回もタップしなければならないためであると思われる。また、被験者の意見にもあったが、現在の方法では前の候補に戻る手段が用意されていないため、誤って正しい候補を通り過ぎてしまったときに、また何回もタップしなおす必要があることも原因であると思われる。何らかの方法で前の候補に戻るインタフェースが用意されれば、もっと良い評価になる可能性がある。

離れた場所に候補を表示する方法とメニュー形式で表示する方法では、有意差はないが、メニュー形式の方が若干訂正時間が短い。主観評価でも、離れた場所に表示する方法よりもメニュー形式の方が使いやすいと評価されている。これは、他の場所に表示する方法と比較してメニュー形式の方が視線の移動が少ないからであると思われる。

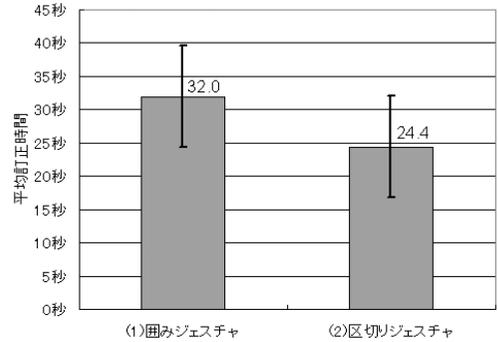
以上の結果から、今回提案した 3 つの方法の中では、メニュー形式で表示する方法が一番適切であると考えられる。

### 4.5 文字区切り位置の訂正インタフェースの比較

#### 4.5.1 実験目的

文字の区切り位置が違っていった場合の訂正インタフェースとして、2.3 節で提案した次の訂正インタフェースの中でどのインタフェースが一番使いやすいかを調べるために、文字を分割する場合と結合する場合のそれぞれについて比較評価実験を行った。

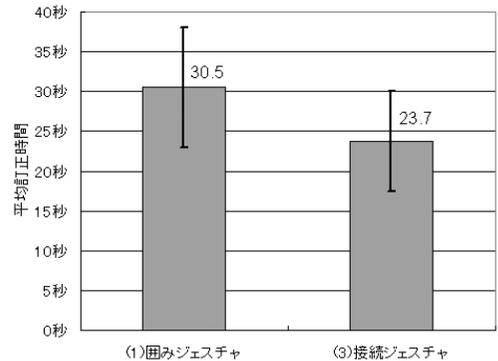
- (1) 囲みジェスチャ
- (2) 区切りジェスチャ(文字を分割する場合だけ)



縦棒は平均訂正時間、線分は平均訂正時間 ± 標準偏差の幅を示す。

図 15 文字を分割するための所要時間の比較評価

Fig. 15 Comparative evaluation on the time required for separating characters.



縦棒は平均訂正時間、線分は平均訂正時間 ± 標準偏差の幅を示す。

図 16 文字を結合するための所要時間の比較評価

Fig. 16 Comparative evaluation on the time required for merging character segments.

### (3) 接続ジェスチャ(文字を結合する場合だけ)

#### 4.5.2 実験結果

各訂正インタフェースを用いて、7カ所の誤った区切り位置を分割訂正した場合の平均訂正時間を図 15 に、結合訂正した場合の平均訂正時間を図 16 に示す。図 15 と図 16 は、平均時間を縦棒、平均値 ± 標準偏差の幅を線分で示している。また、危険率 5% の t 検定を行い、項目間の差を検討した結果、分割訂正した場合の項目 (1) と項目 (2)、および、結合訂正した場合の項目 (1) と項目 (3) の間で有意差が認められた。

各訂正インタフェースの使いやすさを、文字が小さく書かれている場合と大きく書かれている場合について 5 段階で評価してもらった結果を図 17、図 18 に示す。なお、本論文では、小さく書かれている場合として、文字の縦幅または横幅が 3 mm (11 ピクセル程度) 以下の場合 (“1” や “一” など) や、画数の多い文字

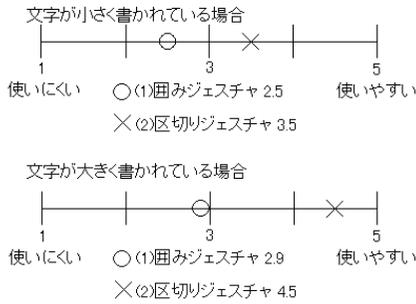


図 17 文字の分割指示に対する主観評価

Fig. 17 Subjective evaluation on separating characters.

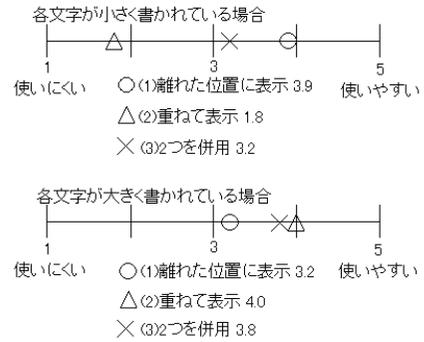


図 19 認識結果の表示方法に対する主観評価

Fig. 19 Subjective evaluation on displaying recognition results.

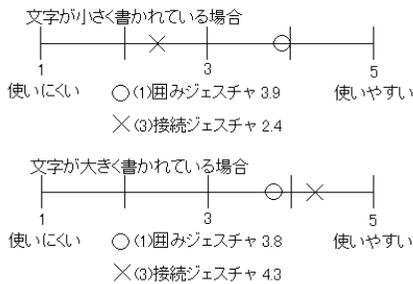


図 18 文字の結合指示に対する主観評価

Fig. 18 Subjective evaluation on merging character segments.

(“識”など)が 10 mm (36 ピクセル程度) × 10 mm 以下の大きさで書かれた場合などを想定している。また、自由に意見を記述してもらった結果、次の意見が得られた。

- 囲みジェスチャは、囲みたい文字の形が複雑だと面倒である。
- 小さい文字の場合、接続ジェスチャは失敗が多い。

#### 4.5.3 実験の考察

文字を分割するためのジェスチャは、訂正時間、主観評価両方とも、囲みジェスチャよりも区切りジェスチャの方が使いやすいという結果になった。これは、区切りジェスチャの方が囲みジェスチャよりも入力すべきストローク長が短いためであると思われる。

文字を結合するためのジェスチャは、訂正時間では接続ジェスチャの方が有意に優れていたが、主観評価では、手書き文字が小さく書かれている場合には囲みジェスチャ、大きく書かれている場合には接続ジェスチャの方が使いやすいという結果になった。

これは、接続ジェスチャの方が入力すべきストローク長が短く便利であるが、文字が小さく書かれた場合は接続に失敗する危険性が高くなり、結果的に何度もやり直す必要が出てくるためであると思われる。なお、

今回の実験では時間測定するとき各文字の再入力を無効としたが、再入力機能と併用した場合には接続失敗が再入力と判定される危険性もある。

以上の結果から、文字の分割操作は区切りジェスチャ、文字の結合操作は、文字が小さく書かれている場合には囲みジェスチャ、文字が大きい場合には接続ジェスチャが適切であると考えられる。

## 4.6 認識結果の表示方法の比較

### 4.6.1 実験目的

認識結果の表示方法として、2.5 節で提案した方法の中でどの表示方法が一番適切かを調べるために、比較評価実験を行った。表示方法については誤認識訂正時間の測定を行わず、4.3 節で述べたように、簡易文章入力ツールに慣れてもらうための練習中に、次の 3 種類の表示方法すべてを試してもらい、それに基づき簡単なアンケートに回答してもらった。

- (1) 手書き文字から離れた位置にまとめて認識結果を表示 (離れた位置に表示)。
- (2) 手書き文字の上に認識結果を重ねて表示 (重ねて表示)。
- (3) 上記 2 つの方法を併用して表示 (2 つを併用)。

### 4.6.2 実験結果

各認識結果の表示方法について、その適切さを 5 段階で評価してもらった結果を図 19 に示す。また、自由に意見を記述してもらった結果、次の意見が得られた。

- 重ねて表示する方法は、色により読みやすさが異なる。
- 小さい文字に認識結果を重ねると読みにくい。
- 候補選択領域はあった方がよいが、認識結果や手書き文字が見にくくなる場合がある。

### 4.6.3 実験の考察

評価の結果、文字が小さく書かれている場合には離

れた場所に認識結果を表示する方法が、文字が大きく書かれている場合には重ねて表示する方法の方が適切であることが示された。これは、重ねて表示する方法の方が視線の移動量が少なくなるが、文字が小さくなった場合に結果と手書き文字が読みにくくなるという問題が発生するためであると考えられる。この問題は、結果を表示する際に手書き文字の色を薄い色で表示し、認識結果を濃い色で表示するなど、色と濃さを調整することである程度改善できると考える。

## 5. 終わりに

本論文は、枠なし認識における誤認識を簡便に訂正し、円滑な文字列入力を行えるインタフェースの基本設計を明らかにすることを目的とし、枠なし認識における誤認識の訂正インタフェース、および、認識結果の表示インタフェースを提案した。さらに、それらのインタフェースの比較評価を行った結果、次のことが明らかになった。

- 各文字の誤認識訂正は、訂正候補一覧が手書き文字の上にメニュー形式で表示され、その一覧から選択する方法が適切である。
- 文字の分割操作は、区切りジェスチャが適切である。
- 文字の結合操作は、文字が小さく書かれている場合には囲みジェスチャ、文字が大きい場合には接続ジェスチャが適切である。
- 認識結果の表示は、文字が小さいなどの理由により読みにくくなる場合には離れた場所にまとめて表示する方法が、結果と手書き文字が読みにくくならなければ手書き文字の上に認識結果を重ねる方法が適切である。

また、本論文では次のインタフェースを提案した。

- 枠の中から手書き文字を書き直すことで、各文字を再入力できるインタフェース
  - 訂正位置より後方の認識結果を自動的に訂正
  - 候補選択領域の表示
  - 促音や拗音などの小さい文字を、色を変えて表示
- 今後は比較評価の結果に基づき、より使いやすい誤認識訂正インタフェースを実現するとともに、枠なし手書き文字列認識をより身近なものにしていきたい。

謝辞 本論文の執筆にあたり、多大なご助言をいただいた葎田まりさんに心から感謝する。また、評価実験に参加していただいたすべての方々へ深く感謝する。本研究の一部は基盤研究(B)(2)11558031の補助による。

## 参考文献

- 1) 橋本美奈子, 長嶋雲兵, 富樫雅文, 細矢治夫: ペン入力のための楕円形仮想キーボードとベクトル入力法, 情報処理学会論文誌, Vol.37, No.11, pp.2105-2115 (1996).
- 2) 増井俊之: ペンを用いた高速文章入力手法, インタラクティブシステムとソフトウェア IV, 日本ソフトウェア科学会 WISS'96, 近代科学社, pp.51-60 (1996)
- 3) 情報処理学会(編): 新版情報処理ハンドブック, 第10編 4章 4.5節, pp.1171-1174, オーム社 (1995).
- 4) 田村 博(編): ヒューマンインタフェースハンドブック, pp.189-193, オーム社 (1998).
- 5) 福島俊一, 山田洋志: 予測ペン入力インタフェースとその手書き操作削減効果, 情報処理学会論文誌, Vol.37, No.1, pp.23-30 (1996).
- 6) 福島貴弘, 中川正樹: 確率モデルに基づくオンライン枠なし手書き文字列認識, 信学技報, PRMU98-139, pp.25-30 (1998).
- 7) 仙田修司, 濱中雅彦, 山田敬嗣: 切り出し・認識・言語の確信度を統合した枠なしオンライン文字列認識手法, 信学技報, PRMU98-138, pp.17-24 (1998).
- 8) 荒井俊史, 正嶋 博, 福永 泰: 手書きユーザインタフェースにおける文字入力方法, 情報処理学会論文誌, Vol.35, No.3, pp.478-487 (1994).
- 9) 中島直樹, 宮原末治, 若原 徹, 小高和己: マルチメディア端末用手書き入力インタフェースの設計とその応用, 電子情報通信学会論文誌 D-II, Vol.J79-D-II, No.4, pp.592-599 (1996).
- 10) Huerst, W., Yang, J. and Waibel, A.: Interactive Error Repair for an Online Handwriting Interface, *Proc. 1998 ACM SIGCHI Conference on Human Factors in Computing Systems, Student Posters*, pp.353-354 (1998).
- 11) 坂東宏和, 福島貴弘, 加藤直樹, 中川正樹: 枠なし手書き文字列認識における誤認識訂正インタフェースの研究, 情報処理学会研究報告, 2000-HI-89, pp.81-88 (2000.7).
- 12) 加藤直樹, 田中 宏, 中川正樹: 公開インクフォーマットの設計と手書き電子メール環境の開発, 電子情報通信学会論文誌 D-I, Vol.J84-D-I, No.2, pp.203-212 (2001).

(平成 13 年 6 月 1 日受付)

(平成 14 年 3 月 14 日採録)



坂東 宏和(正会員)

1975年生。2002年東京農工大学大学院工学研究科電子情報工学専攻博士後期課程修了。対話型電子白板を用いた一斉授業を支援する教育ソフトウェア, および教育ソフトウェア全般に興味を持つ。工学博士。

工学博士。



福島 貴弘(学生会員)

1972年生。1998年東京農工大学大学院工学研究科電子情報工学専攻博士前期課程修了。現在, 同学博士後期課程に在学中。手書き文字認識の研究に興味を持つ。電子情報通信学会会員。

学会会員。



加藤 直樹(正会員)

1969年生。1998年東京農工大学大学院電子情報工学専攻博士後期課程修了。1997年より日本学術振興会特別研究員を経て, 1999年より東京農工大学工学部助手。手書きユーザ

インタフェースの研究・教育に従事。情報教育, 教育の情報化に興味を持つ。工学博士。ACM, ヒューマンインタフェース学会, 電子情報通信学会各会員。



中川 正樹(正会員)

1954年生。1979年東京大学大学院理学系修士課程修了。同在学中, 英国 Essex 大学留学 (M. Sc. in Computer Studies)。1979年東京農工大学工学部助手。現在, 教授。共同研究開発センター長併任。パタン認識, 手書きインタ

フェース, 情報教育等の研究・教育に従事。理学博士。

