

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

坂東 宏和, 福島 貴弘, 加藤 直樹, 中川 正樹
東京農工大学工学部

本稿では、文字記入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。枠なし文字認識では、文字間の区切り判定と文字認識の両方で誤りが発生する危険がある。そこで、本システムでは、認識結果として、各文字の区切りを示す外接矩形と、その外接矩形内に入力パタンに重ねて認識候補文字を表示することにより、ユーザの確認と訂正を仰ぐ。各文字の認識結果を訂正するためには、文字区切り内をタップして認識候補一覧から選択するか、そこに文字を再入力することにより行い、区切りの訂正には、囲みまたは区切りジェスチャを用いる。予備評価の結果、文字が小さく書かれた場合や、文字間を詰めて書いた場合に、認識結果が読みにくい、囲みと区切りの操作がしにくいという問題が明らかになった。

User interfaces for correcting errors in writing-box-free recognition of handwritten text.

Hirokazu Bandoh, Takahiro Fukushima, Naoki Kato, Masaki Nakagawa
Dept. of Computer Science, Tokyo Univ. of Agriculture and Technology.

This paper presents a design and its preliminary implementation of user interfaces for correcting errors in writing-box-free recognition of handwritten text. In the writing-box-free recognition, errors may occur in character segmentation and character recognition. Therefore, our system shows the segmentation result by displaying a bounding box for each segmented character pattern and shows the recognition result by overlaying an answer on top of the pattern in each box. Users confirm the results and correct them if necessary. Correction of character recognition is made by selecting a right answer from the candidates list shown by tapping in each bounding box or by rewriting a pattern there. On the other hand, correction of segmentation is performed by the combining or cutting pen gestures. A preliminary evaluation has revealed that recognition results are hard to read and the pen gestures are difficult to write when characters are written in small sizes or when they are written very closely to neighbors.

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。

電子ペン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、まさに鉛筆と紙のように使える手軽なインタフェースとして研究が進められてきた¹⁾。電子ペンによりテキストを入力する方法としては、仮想キーボード²⁾や手書き文字入力³⁾⁴⁾がある。手書きによる文字入力は、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

ードよりも入力効率が劣る。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⁹⁾などが行われている。

また、従来の手書きによる文字入力、認識率を向上させるために記入枠を設け、その記入枠の中に1文字ずつ手書き文字を入力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)。この場合、記入枠のない状態よりも誤認識は減少し、結果的に入力効率は向上する。しかし、筆記者に対し、枠の中に書かなければならないという余計な負担を与えてしまう。

そこで筆者の所属する研究室では、この余計な負担をなくし、筆記者が自由な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、78%程度の認識率を得ている¹⁰⁾。自由な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同様、特別な意識を払うことなく、自然なスタイルによりスムーズ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、手書き文字を認識するときに誤認識が発生し、その誤認識の訂正作業が必要になる。したがってスムーズ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。

また、過去に報告されたオンラインで手書きにより文字を入力するシステムのいくつかは、オンライン入力のリアルタイム性に着目し、入力されたパターンを即座に認識する方法を採用している。しかし我々は、リアルタイムに認識する方法は、ユーザに思考の中断を強いると考え、入力パターンを即座には認識しない遅延認識を提

案してきた⁷⁾。

そこで本稿では、それらを踏まえ、枠なし認識における誤認識を簡便に訂正し、スムーズな文字入力を行えるようにすることを目的とした誤認識訂正インタフェースの設計とその試作について述べる。

本稿の2章で枠なし認識における誤認識訂正インタフェースについて検討し、3章において、2章で提案したインタフェースを実現するための具体的なアルゴリズムを示す。4章で文書入力ツールの試作を行い、5章で予備評価とその結果を報告し、6章で結論を述べる。

2. 誤認識訂正インタフェース設計

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イメージ図を図1に示す。

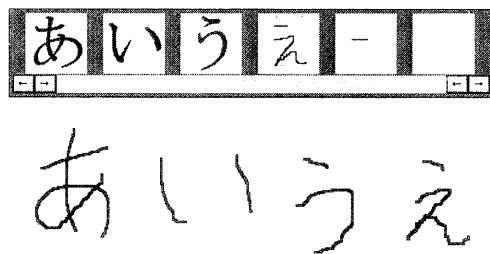


図1 枠あり認識(上段)と枠なし認識(下段)のイメージ図

枠あり認識は、基本的に一文字毎に認識を行うため、誤認識の種類は、各文字の認識結果が異なる場合だけである。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実現すればよい。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異なる。
- (2)各文字の区切り位置が異なる。

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、各文字の区切り位置の訂正インタフェースも必要となる。

2.2 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、次のような方法が考えられる。

- (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 (図2 左)
- (2)手書き文字から離れた位置に認識結果を表示する方法 (図2 中)
- (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 (図2 右)

手書き文字を消去してしまう(1)の方法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を忘れてしまう可能性がある。また、文字の区切り位置が間違っって認識された場合に、正しい区切り位置を示すために元ボタンを再表示する操作が必要になり面倒である。一方(2)は、SHARP 社の電子手帳や Microsoft 社の IME パッドなどで採用されている方法であるが、確認や訂正のために、視線が手書き文字ボタンと認識結果間を行き来することになり、ユーザが疲れるという問題がある。そこで本研究では(3)を採用する。(3)の欠点としては、元ボタンが見にくくなる可能性が考えられるが、その欠点は、元ボタンと認識結果の表示色、および、線の太さを調節することで改善できると考える。また、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3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。

枠なし文字認識では、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。

- (1)元ボタンを、認識結果 1 文字分ごとに色を変えて表示する (図4上)。
- (2)枠を表示する (図4下)。

色を変えて表示する(1)の方法の場合、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には、一部の色が見にくくなる可能性がある。またカラー表示ができない端末には適用しにくい。そこで本研究では、(2)の方法を採用する。

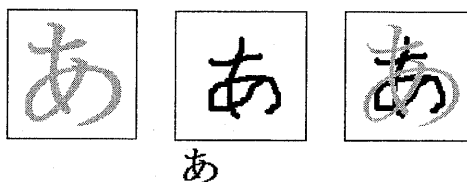


図2 認識結果の表示例

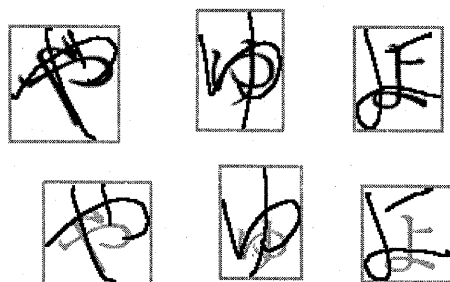


図3 認識結果の表示 (小さい文字が下段)

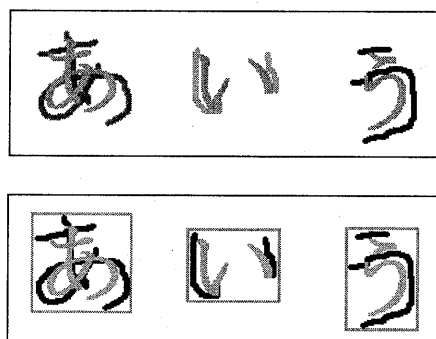


図4 区切り位置の表示例

2.3 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。

認識候補の中から選択する場合のインタフェースとしては、次の方法が考えられる。

- (1) 候補一覧を離れた場所に表示し、その一覧から選択する(図5左上)。
- (2) 枠内をタップすると、次の候補に変更される(図5下)。
- (3) メニュー形式で候補一覧を表示し、選択する(図5右上)。

候補を離れた場所に表示する(1)の方法の場合、認識結果の表示でも述べたように、訂正のために視線が手書き文字ボタンと訂正候補間を行き来することになり、ユーザが疲れるという問題がある。(2)の場合、正解文字が候補の上位に入っている場合には良いが、下位に入っている場合や候補に入っていない場合何回もタップする必要があり面倒である。そこで本研究では、(3)を採用する。(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回となり、タップ回数は(2)より減少する。

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、図6のように、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。

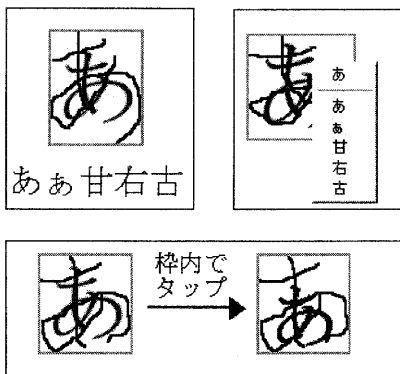


図5 各文字の訂正候補の表示



図6 各文字の再入力

2.4 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) 文字の結合：例えば「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。
- (2) 文字の分割：例えば「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。

分割位置の訂正方法としては、囲みジェスチャおよび区切りジェスチャが考えられる(図7、図8)。

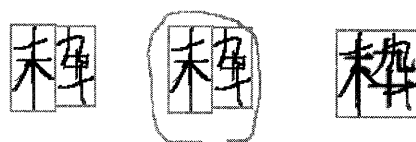


図7 囲みによる区切り位置訂正



図8 区切りによる区切り位置訂正

囲みジェスチャは、(1)(2)両方の場合に対応できる。一方区切りジェスチャは、(2)の場合にしか対応できないが、(2)の場合には囲みジェスチャよりも入力すべきストローク量が少ない。そこで本研究では、囲みジェスチャと区切りジェスチャの両方のインタフェースを実装する。

2.5 認識結果の訂正範囲

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。そこで、2.2節から2.4節で述べた訂正が行われたときに、図9のように、訂正部分よりも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だけを認識することで再認識を効率化し、さらに1回の訂正で複数の誤認識を訂正する。また、一度訂正された部分については、再認識の対象としない。

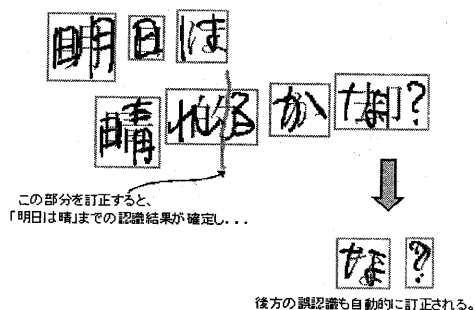


図9 誤認識の自動訂正

3. アルゴリズム

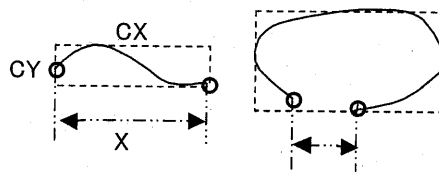
3.1 各文字の訂正とベンジェスチャの判別

入力が、2.3節で述べた各文字の訂正と、2.4節で述べた区切り位置訂正のどちらであるかを判別するために、再入力は区切り位置を示す外接矩形の中から、区切り位置訂正ジェスチャは、外接矩形の外側から始めるものとする。

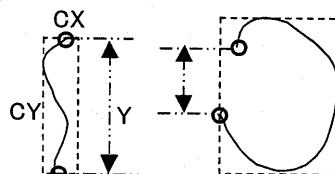
入力が各文字の訂正であった場合、それが、訂正候補表示と再入力のどちらであるかの判別は、ペンのダウン位置からペンアップ位置までの距離により判別する。

また、入力が区切り位置訂正のためのベンジェスチャであった場合、入力されたベンジェスチャが、区切りジェスチャと囲みジェスチャのどちらであるかの判別は、次の通り行う(図10)。

直線: 囲み、または区切り線
 破線: 囲み、または区切り線の外接矩形(CX,CY)
 2点鎖線: 始点と終点の差(XorY)
 円: 始点・終点



CX > CY の時
 $CX \times n < X$: 区切り
 $CX \times n \geq X$: 囲み



CX ≤ CY の時
 $CY \times n < Y$: 区切り
 $CY \times n \geq Y$: 囲み

n は $0 < n \leq 1$ で、0.8 程度の値を指定

図10 囲みと区切りの判別処理

- (1) 区切りまたは囲み線の外接矩形について、外接矩形の横幅が縦幅より大きい場合には、区切りまたは囲み線の始点と終点の X 座標の差と、外接矩形の横幅 $\times n$ ($0 < n \leq 1$) を比較する。その結果、始点と終点の X 座標の差の方が小さい場合には囲み、大きい場合には区切りと判定する。
- (2) 区切りまたは囲み線の外接矩形について、外接矩形の縦幅が横幅より大きい場合には、区切りまたは囲み線の始点と終点の Y 座標の差と、外接矩形の縦幅 $\times n$ ($0 < n \leq 1$) を比較する。その結果、始点と終点の Y 座標の差の方が小さい場合には囲み、大きい場合には区切りと判定する。

3.2 囲みと区切りのアルゴリズム

囲みの判別は、手書き文字列の各線を構成す

る点が、囲み線の中にある一定の割合以上含まれている場合、囲まれていると判定する⁸⁾。

区切り位置の判別は、認識結果 1 文字分ごとに、区切り線に追加線を加えて囲みに変換し、囲みと同様の処理を行うことで実現する。具体的には、次の処理を行う。

- (1) 認識結果の外接矩形領域と区切り線の外接矩形が重なるかどうかを判別し、重ならない場合には(2)以下の処理を省略して区切られていないものとする(図 11)。
- (2) 区切り線の外接矩形の横幅と縦幅を比較し、縦幅の方が長い場合には縦方向の区切り、横幅の方が長い場合には横方向の区切りとする(図 12)。
- (3) 区切り線に、図 13のように点を追加し、囲みに変換する。なお図 13は、上から下、または左から右へ区切り線が書かれた場合の例である。下から上、または右から左へ区切り線が書かれた場合には、点列を逆にして同様の処理を行う。
- (4) 囲みと同様の判定を行う。

直線: 区切り線、2点鎖線: 区切り線の外接矩形
点線: 認識結果各文字の外接矩形領域

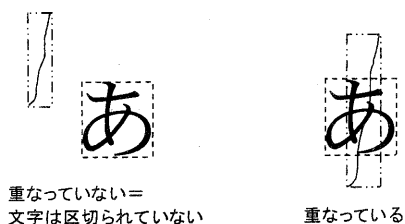


図11 外接矩形による判別

直線: 区切り線、2点鎖線: 区切り線の外接矩形

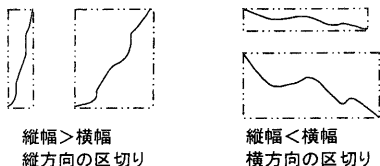


図12 区切り方向の判別

直線: 区切り線
2点鎖線: 区切り線の外接矩形
点線: 認識結果各文字の外接矩形領域
太線: 追加線

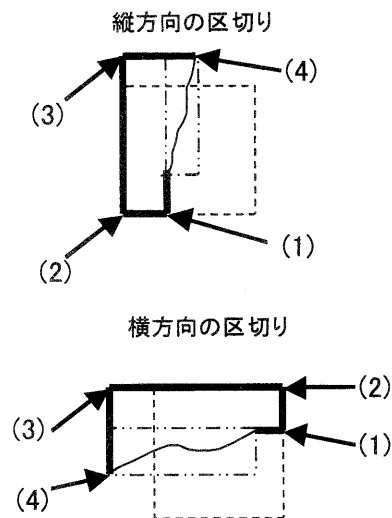


図13 区切りと囲みの変換

4. 誤認識訂正インターフェースの実現

2章で述べた誤認識訂正インターフェースの利点や問題点を明らかにするため、2章で述べた設計に従い、枠なし手書き文字認識を用いた簡易文章入力ツールを実現した。簡易文章入力ツールは、任意のテキストエディタや Microsoft Word、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するためのツールである。

簡易文章入力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、図 14のように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一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を認識し、図 15 のように認識結果を表示する。認識結果は、区切り位置を外接矩形で示し、認識結果はその外接矩形内に表示する。なお認識結果の文字サイズは、外接矩形の大きさに合わせて自動的に調節される。

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(図 16)。

誤認識が発生した場合には、2章で述べた方法により修正することができる。

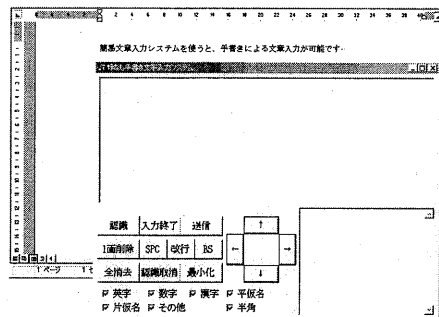


図16 文字列の送信結果

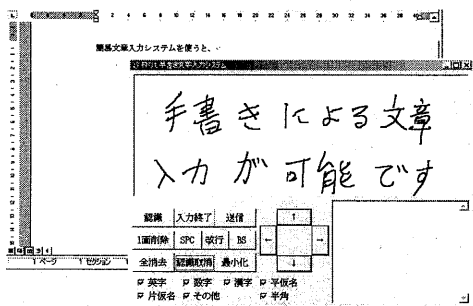


図14 手書きによる文字入力例

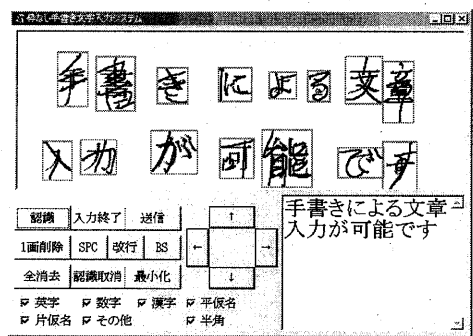


図15 認識結果の表示

5. 予備評価

5.1 試作ツールの試用

試作した簡易文章入力ツールを、本学の学生3名に試用してもらった。試用後、被験者から次のような意見を得られた。

- ・ 枠あり認識よりも、自然に入力できる。
- ・ 自由な筆記スタイルで素早く入力できる。
- ・ 文字が小さいと、表示される結果が読みにくい。
- ・ 訂正候補を表示しようとして、再入力になってしまうことがあった。
- ・ メニューによる訂正候補選択の時に、キャンセルがほしい。
- ・ 再入力後の認識速度が遅い。
- ・ 文字間が詰まると区切りや囲みの操作がしづらい。

5.2 考察

認識結果の表示に関しては、小さい文字を書いたときに、読みにくいという意見があった。これは、認識結果の表示サイズを、各文字の外接矩形の大きさに合わせて自動調節しているため、文字を細かく書いた場合に、非常に小さいフォントサイズで結果が表示されるためであると思われる。この問題の改善方法としては、フォントの最低サイズを設定し、外接矩形の大きさが小さい場合でも、そのサイズで表示する方

法が考えられる。しかし、この場合、文字間を詰めて書いた場合に、隣の認識結果と重なってしまい、余計に見にくくなる危険性がある。したがって、隣接する文字の認識結果の表示位置を考慮し、適切な位置と大きさで認識結果を表示することが必要であると思われる。

各文字の誤認識訂正に関する意見としては、訂正候補を表示しようとして、再入力になってしまうという問題が指摘された。これは、訂正候補表示と再入力のどちらであるかを、ペンダウン位置からペンアップ位置までの距離により判別しているが、この距離の敷居値が小さすぎたためであると考えられる。ペンによるタップの場合、マウスによるクリックと違いすべりやすいため、ペンアップ位置とペンダウン位置の距離が遠くなる傾向がある。そのため、マウスの時よりも大きな敷居値に設定する必要がある。

区切り位置の訂正に関しては、文字間が詰まると訂正しにくいという問題が指摘された。この問題については、書いた手書き文字の拡大表示機能などで対応できると思われる。

全体として、小さく文字が書かれた場合や、文字間が詰まった場合に、使いづらくなるのが分かった。これは、元の入力パタン、区切り位置を示す外接矩形、認識結果文字を同じ場所に表示しているため、結果が見にくくなったためであると考えられる。認識結果に関しては、表示色を工夫するなどの対策が必要であると考えられる。

6. 終わりに

本稿では、枠なし認識における誤認識を簡便に訂正し、スムーズな文字入力を行えるようにすることを目的とした誤認識訂正インタフェースの設計とその試作について述べた。本研究では、認識結果を、各文字の区切り位置を示す外接矩形と、その外接矩形の中に文字を表示することで示す方法を採用した。また、各文字の訂正は、訂正候補一覧から選択するか、各文字を

再入力することにより行い、区切り位置の訂正は、囲みまたは区切りジェスチャにより行うインタフェースを提案した。

そのインタフェース設計に基づいて試作した簡易文章入力ツールの予備評価の結果から、小さく文字が書かれた場合や、文字間を詰めて書いた場合に、認識結果が読みにくい、囲みと区切りの操作がしにくいといった問題が明らかになった。

今後は予備評価の結果を基にインタフェースを改良し、より詳細な評価実験を行いたい。

謝辞

評価実験に参加していただいたすべての方々へ感謝する。

本研究は、情報処理振興事業協会による高度情報処理ソフトウェアシーズ育成事業「手書きインターネットプラットホーム」の一部補助により行われた。

参考文献

- 1) 福永泰：ペン入力技術－紙の操作性を目指して－，情報処理，Vol.33 No.7，pp.820-827(1992).
- 2) 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).
- 3) 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，電子情報通信学会論文誌，D-II Vol.J81-D-II No.4，pp.651-659(1998).
- 4) 田中宏，中島健次，石垣一司，秋山勝彦，中川正樹：オンライン認識とオフライン認識の候補統合によるハイブリッド型ペン入力文字認識エンジン，信学技報，PRMU98-140，pp.31-38(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) 曾谷俊男，福島英洋，高橋延匡，中川正樹：遅延認識を用いた手書きユーザインタフェースの基本設計，情報処理学会論文誌，Vol.34 No.1，pp.158-166(1993).
- 8) 中川正樹，佐藤俊：表示一体型タブレット上でのペンの囲みに対する対象の包囲を判定する高速アルゴリズムの実現と評価，信学論，Vol.J77-D-II，No.8，pp.1630-1639(1994).