

文字認識における「歩み寄りインタフェース」の試作と検証

篠原 祐樹[†]

宮下 芳明[†]

概要：本稿では、認識器が認識に苦勞する部分についてユーザが妥協し意識的に認識し易い入力を与えるという「歩み寄りインタフェース」を導入した手書き文字認識システムの試作および検証を行った。評価実験の結果、認識率の向上および入力時間の短縮が観測され、歩み寄りインタフェースの導入により入力の効率が上がる事が確認された。

"The Compromise Interface" in Handwriting Recognition

Yuki Shinohara[†] Homei Miyashita[†]

Abstract: In this paper, we developed an interface for compromise between handwriting recognizer (the computer) and recognizee (the user), and conduct the experimental tests. As a result we found that it contributes to the growth in recognition rate and in entry speed.

1. はじめに

コンピュータに対し直感的に情報を入力する手段として手書き文字や音声、ジェスチャなどが挙げられるが、このような自然情報はコンピュータにとって曖昧な物が多いうえ、入力のブレが大きいため認識が困難な場合がある。

認識率を改善するため種々のアルゴリズム等が研究されており、本稿で扱う文字認識の分野においても HMM などを用いた認識アルゴリズムの改良 [1]、くせ字の学習データの収集技術 [2] など枚挙にいとまがない。しかし、依然として誤認識が多いのが現状であり、キーボードやマウス等の入力手段に代わるほどの速度や信頼性を達成できていない。そのような背景から、「コンピュータが 100% の認識を実現する事は不可能」という前提に立ち、認識誤りを訂正しながら入力を行うインタフェース [3] や、ストロークを描くたびに列挙される認識候補リストの中から入力したい字や語句を選択する手書き文字入力インタフェース [4] などの研究も行われている。その一方で、キーボードが搭載できないような携帯端末においては Palm で実装された Graffiti [5] のように手書き文字入力が普及した例もある。これは、T を 7 のように、K を ㄨ のように書くなど文字の形状自体を変えてしまうという発想により、「U」と「V」など紛らわしい入力を回避したほか、全ての文字を一画で表現したため素早い入力を可能にし、速度と信頼性の両面において優れた入力機構を実現した。ところが、一部の文字については実際の文字とかけ離れた形状や筆順をした物も存在し、全ての文字について書き方を覚えスムーズに入力出来るようになるためには訓練を行わなければならない。ユーザの負担が大きくなってしまっていた。特に文字数の多い日本語で上記のようなシステムを実装した場合、この問題はいっそう深刻になる。

もっとも、後藤が「熟練者だけが使える音声イン

タフェース」について指摘しているように [6]、「もしそれが十分魅力的で慣れると使いやすければ、そのインタフェースの熟練者が使っている様子を見た人達が、自分も使いたく自発的に練習するようになり、次第に普及していく可能性がある」。しかしその一方で、入力方式が直感的であればあるほど、ユーザ側に練習するという心理は働かなくなるという現状もある。これは、タイピング練習ソフトが数多く存在するのに対し、手書き文字入力や音声入力の練習ソフトが普及していない事を考えれば明らかであろう。だからこそ、後藤が指摘するように、ユーザに熟練を求めることは「一見時代に逆行」して見えるのであり、このような視座に立った研究は未だ少ない。

こうした背景から、コンピュータの認識機構において現在必要とされているのは、ユーザが少ない労力を負担することで高い認識結果を得ることが出来る「妥協案」ではないかと筆者らは考えている。そこで本稿では、コンピュータが認識に苦勞するところだけ人間側が妥協できるような着地点を探す「歩み寄りインタフェース」を実装する手書き文字認識システムを試作し、その検証を行う。

2. システム概要

本システムは、以下のようなプロセスを念頭に置き、最終的にはコンピュータが認識に苦勞する文字について、ユーザが自分の文字の書き方を変更しそれに適応することで、紛らわしい手書き文字の入力を回避出来るようになるように設計されている。

(1) コンピュータがユーザの筆跡を学習

ユーザの入力した手書き文字を保存しておき、認識の際に用いる。

(2) ユーザが書き方を修正

コンピュータが認識に苦勞する文字をユーザが把握し、ユーザ側がそれに対する妥協案として自分の文字の書き方を修正する事を宣言す

[†] 明治大学理工学部情報科学科

[†] Department of Computer Science, Meiji University

る。ここで修正された書き方はユーザの筆跡データベースに反映され、次回以降の認識に生かされるようになる。

(3) 練習により修正した書き方に慣れる

変更した筆跡をもとに文字認識が行われるようになるので、ユーザもそれに適応しなければならない。本システムではこのための練習を行う仕組みを用意している。

2-1. くせ字セット

本システムでは、入力された文字が何であるか認識する際に必要なサンプルの文字として、ユーザがあらかじめ手書き入力によって書いた文字を使用している。これらの手書き文字の集合を本システムでは「くせ字セット」と呼んでいる。

文字の認識は、入力された手書き文字とくせ字セット内の各文字との非類似度をそれぞれ求める事により行う。この計算はストロークの位置座標列および、それから得られる曲がり方向ベクトル列の比較により行う。手書き入力される文字は最終的な形状こそ似ているが、入力の度にその経過にはバラツキがあり各ストロークの座標系列には時間的な伸縮が生じてしまう。このようなブレを吸収するために、比較にはD P 照合法[7]を用いている。

これらの計算の結果、くせ字セットに登録された文字の中で、入力された文字との非類似度が最も小さくなるものが入力された字であると解釈される。そして、非類似度が2番目に小さい文字は、誤認識する可能性の一番高い字であるということができる。

2-2. 誤認識指数の算出

前節でも述べたように、コンピュータはくせ字セットに登録された書きぐせの文字と入力された文字を比較することで手書き文字の認識を行うため、くせ字セットが決まれば誤認識の起きやすさもある程度類推できる。そこで、「誤認識指数」を以下のように定義し、ユーザの書きぐせにおいて誤認識がどの程度起きるかの指標とした。「誤認識指数」が低ければ低いほど、認識しやすい書きぐせであるといえる。文字 C_i と C_j の非類似度が a_{ij} で与えられるとき、誤認識指数 S を以下のように逆二乗の総和を用いて求める。(n はくせ字セットの文字数、 K_G は比例定数)

$$S = \sum_{i=1}^n \left(\sum_{j=1}^n K_G \frac{1}{a_{ij}^2} \right) / n$$

紛らわしい書きぐせをもつ文字対が存在していた場合、それらの非類似度は小さくなるため、誤認識指数 S は大きく跳ね上がる。このため、 S を小さくするためには、似ている文字ができるだけ似ないように書きぐせを変えて非類似度の小さな文字対を排除する必要がある。このように誤認識指数 S を小さくしていく作業は、本システムの目標である「似て

いる文字を書かなくする」と結びついており、その目標の達成度を表すひとつの指標となる。

2-3. くせ字マップ

コンピュータが文字認識における曖昧さを可視化し、ユーザがどのように歩み寄るべきか提示する手法として、本システムでは「くせ字マップ」なる機能を導入した。

「くせ字マップ」では、くせ字セットに登録されている文字同士の非類似度を空間における距離に見立て、似ている文字同士がより近くに、似ていない文字同士は遠くに配置し表示する。これによって、各文字同士の類似関係を視覚的に理解できるようにする(図1)。

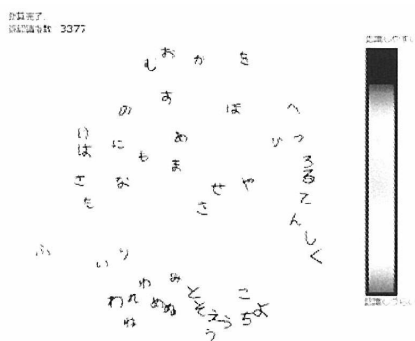


図1. くせ字マップ

各文字の配置には多次元尺度構成法[8]を用いている。これは、複数のオブジェクトおよびそれらの間の類似度が与えられたとき、各オブジェクトを多次元空間内の点として表し、点間の距離が与えられた類似度と最もよく一致するような点の位置を定める手法である。多次元尺度構成法には様々な手法があるが、本システムでは代表的な手法であるKruskalの方法を用いている。

なお、画面右側のゲージには前節で定義した誤認識指数をもとにユーザの各文字がコンピュータにどれだけ認識されやすいかを視覚的に表示している。

2-4. 書きぐせの修正

前節のくせ字マップにより、どの文字の書きぐせをどのように改善すればよいか把握し、実際に修正を行っていく。

くせ字マップ上で文字をクリックすると修正ウィンドウが開き、その文字を修正することができる。たとえば、くせ字マップにおいて図2(左)のような配置を得た場合、「や」と「せ」は類似度が高く、誤認識を招きやすいが、この「や」を図3のように丸みがかった形状として修正すると、くせ字マップ上で「せ」と「や」の距離は広がり、図2(右)のような配置に改善される。つまり、似ている書きぐせの

文字が減り認識されやすくなったといえる。このように、書きぐせを変更すると即座にくせ字マップの再描画が行われ、また誤認識指数に従って右側のゲージが上下動する。これにより、書きぐせの修正方法を実際に試しながらコンピュータの認識しやすい書き方を見つけていく事が出来る。また、画面右側のゲージにより書きぐせの改善をしていく際に自分の行った修正がどの程度認識率の向上に貢献したかを知る事が出来る。

以降の文字認識においては、修正後の書きぐせがサンプルとして用いられるようになる。

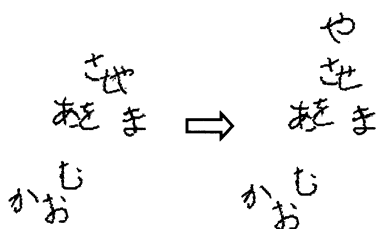


図2. 書き方の修正例とくせ字マップの変化

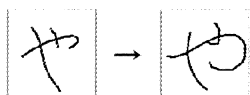


図3. 「や」の書き方の修正例

2-5. 修正した書きぐせの練習

コンピュータ側が修正した書きぐせに対応しても、ユーザがそれに適応していなければ認識率の向上は望めない。本システムでは修正後の書きぐせに適応し、コンピュータにとって認識し易い入力の方法を練習出来る機能を搭載した。

このような練習システムにおいては、ユーザの学習を促すため入力に対し何らかのフィードバックを与え、それを受けてユーザが入力の方法を修正していく、という反復練習を行うのが一般的である。本システムにおいてもこのような反復練習の手法を導入しているが、次章で述べる思想をもとに、可能な限りユーザが直感的かつ少ない学習量で理想的な入力の方法を習得できるようなフィードバックを与えている。

3. ユーザの効率的な「歩み寄り方」

前章のような文字認識システムにおいて最もユーザの負担が大きくなるのは「修正した書きぐせの練習」部であり、この負担を減らす事が歩み寄りインタフェースを実現する上で重要な課題となる。本システムでは、ユーザが少ない学習量で良い入力の方法を習得出来るよう以下のような施策を導入している。

3-1. 練習システムのコンセプト

漢字ドリルなどの一般的な文字練習システムでは「美しい字」をお手本として、その文字をなぞるように練習し、一貫してその美しい字が書けるように練習をする。このような文字学習システムは多く、最近ではゲームソフトとして普及している例もある[9]。こうしたシステムではいかにお手本に近い文字を書く事が出来るかに着目しており、それに基づいて入力文字を添削してくれたり、お手本文字との類似度を得点化したりしている。

しかし、認識率を上げるための練習においてはこのような施策はあまり意味をなさないと筆者らは考える。重要なのは「いかにお手本に近い文字を書けるか」ではなく、「いかに他の文字と差異をつけて書く事が出来るか」である。どんなに汚くお手本とかけ離れた文字でも、あるいは書く文字に一貫性がなかったとしても、これらが「他の文字に似ていない」ならば、その入力に対し高い評価を与えるべきなのである。

また、字によって練習すべき量は異なってくる。書き方の似ている字を多く持つ字は、ある程度お手本に近い文字を意識して書かないと認識されないが、似ている字が全くないような字の場合、書きぐせにバラツキが出てても正確に認識される。さらに、ひとつの文字の中でも意識すべき部分とそうでない部分がある。例えば、図4のように「け」と「は」の場合、認識の際に重要となる部分は白抜きで示した部分であり、ここを正確に書かないと誤認識のおそれが生じる。



図4. 認識の際に重要となる箇所

逆に考えると網掛けで示した部分は「け」と「は」の差異をつける要因とならないため特に意識して書く必要は無く、ある程度の形状を再現できればよい。つまり、白抜きの部分のように認識の際に重要となる部分は練習の必要があるが、それ以外の部分は特に意識的に練習する必要はないのである。

このような差を理解し、意識して書く必要の無い部分で「効率的に手を抜く」ことによってユーザは変更した書きぐせを覚えたり練習したりする過程でその負担を軽減する事が出来る。

3-2. 認識状況モニタ

前節で述べたように、文字を書く中で意識すべき部分とそうでない部分があるが、それらが文字の中のどの部分であるかを判別するのは難しい場合もある。また、意識して書くべき部分について、どのように書けばより認識し易くなるのか試行錯誤を行いたい場合がある。そこで、入力している文字がど

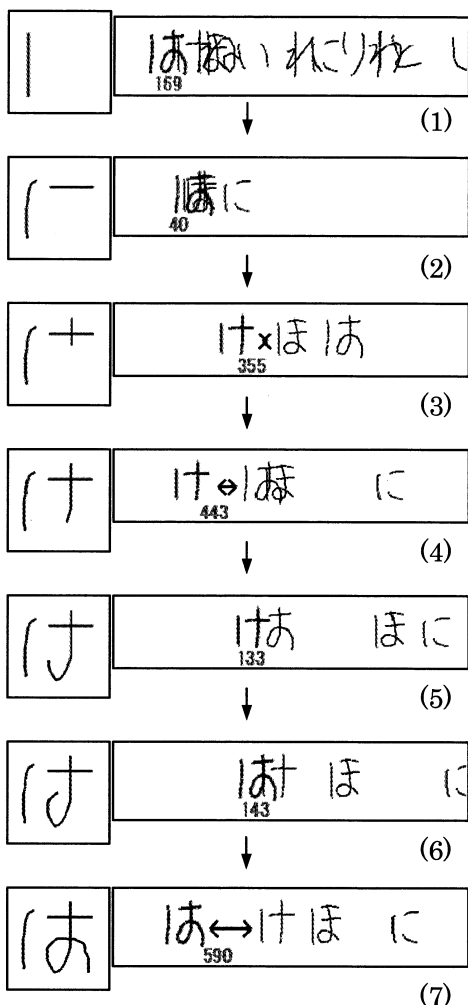


図5. 認識状況モニタ

の文字にどの程度似ているのかをリアルタイムに表示する「認識状況モニタ」を導入した。

「認識状況モニタ」は、現在入力している文字に似ている文字を一覧で表示する。単に列挙するだけでなく、入力文字との非類似度を元に、似ている文字ほどモニタ領域の左端に近づくように配置させる。たとえば図5(1)では、「は」という文字を入力しようとしているが、他にも「け」などの文字の候補がある。ここで、入力している文字に似ているほどモニタ領域の左側に近づくように各文字を配置しているので「は」が第一候補、「け」が第二候補とわかる。図5(2)~(5)のように入力を続けていると、「け」が第一候補となってしまいますが、図5(6)のように3画目の最後の曲げ部では再び「は」が第一候補となっている。このような認識状況の推移を視覚化することにより、ユーザは3画目の曲げ部が「は」を認識させ

るために重要な要素であることを理解することができる。つまり、今後「は」を書くときには、3画目の最後をよく曲げるように特に気をつけて書けば「け」と誤認識されることを避けられるというわけである。さらに、もっと早期の段階で「は」が確実に第一候補となるようにしたいのであれば、くせ字マップ上で1画目を「け」のそれとは異なるものとして修正すればよい。

3-3. 文字の認識しやすさの得点化

ユーザが練習の際に入力した文字の認識しやすさの指標となるスコアを導入する。このスコアはユーザにとって練習の成果や達成の度合いを表す数値となり、効率的な練習やユーザのモチベーション維持を支援するものとなる。

スコア p は以下の式を用いて算出する。

$$p = a_2 - a_1$$

(ただし、 a_1 は入力文字とその認識結果の第一候補となる文字との非類似度、 a_2 は入力文字とその認識結果の第二候補となる文字との非類似度を表す。)

このスコアは「いかに第二候補を突き放すことができたか」を意味しており、「いかに他の文字と差異をつけて書く事が出来るか」に着目するシステムの設計思想を反映した物となる。

これまでの漢字ドリルの文字練習機構の場合、入力している文字がサンプルの文字にどれだけ類似しているかを評価の対象としてきた。これらのシステムで図6(a)(b)における二つの「は」を比べると、(a)の「は」のほうがサンプルに近く「よい文字」であると判断される。この文字はお手本に似た「きれいな文字」であるということもできよう。しかしながら、この(a)は第二候補である「け」と誤認識されるリスクが相対的に高く、実は(b)のほうが正しく認識されやすいのである。また、スコアも(b)のほうが高い値となり、前述したスコアの算出方法が文字の認識しやすさを表す指標として有効であると考えられる。

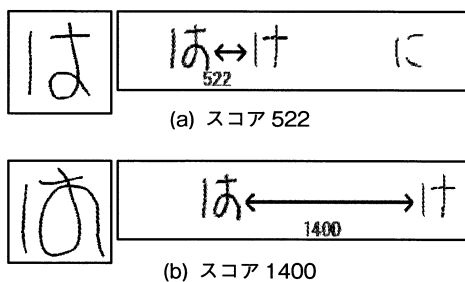


図6. 練習機構における得点

このように、前章で宣言した新しい書き方を忠実に再現することが求められているのではなく、その新しい書き方の中で「どの部分に注意を払えば他の

文字と差をつけることが出来るか」という、いわばその文字のアイデンティティとなる部分をユーザが理解することが重要なのである。そして、その部分を正確に書くことができれば、後の部分は手を抜いて書いても良いのである。「くせ字マップ」を元に書き方を修正して、似ている書きぐせの文字を減らす作業は、手を抜いて書ける文字を多くする作業であるということもできる。

4. 実験

本システムの有効性を評価するために、20代の被験者5名を対象として、液晶ペンタブレット wacom Cintiq12WX を用いて実験を行った。被験者はすべてタブレット等を用いた手書き文字入力の実験経験がほとんどなく、その認識精度を向上させることに特段の欲求はもっていなかった。

4-1. 実験手順

はじめに、「あ」から「ん」までのひらがな46文字を書いてもらい、それらをくせ字セットに登録した。次に、ひらがなの全文字が2度ずつ出現するランダムな文字列を入力してもらい、その認識成功率・入力時間・スコアを測定した（以下測定とよぶ。各種数値は被験者には提示しない）。くせ字マップを使用した書きぐせの修正、および練習フェーズを用いた練習を自由に行い、間で適宜測定を行ってもらうこととした。被験時間は45分程度とした。実験終了後、被験者からその疲労や感想についてインタビューを行った。

5. 考察・検討

5-1. 書きぐせの修正

表1は、実験の最初に登録されたくせ字マップにおいて、その非類似度が小さい文字ペアを上位5ペア抜き出したもの（二人分）である。これを見ると、例えば表1(a)における「し」「く」のペア、あるいは表1(b)における「え」「う」のペアは他の被験者の上位には見られず、個人の書きぐせの傾向が異なることがわかる。一方で、「ぬ」「め」のペアは被験者に共通で見られる傾向であった。修正事例

表1. 非類似度が小さい文字ペア上位5ペア

	書きぐせ修正前		最後の書きぐせ修正後			書きぐせ修正前		最後の書きぐせ修正後	
	文字対	非類似度	文字対	非類似度		文字対	非類似度	文字対	非類似度
1	めぬ	34.5	めぬ	329.9	1	われ	52.8	われ	65.1
2	ちす	48.9	ちす	675.6	2	めぬ	63.6	めぬ	73.9
3	しく	53.4	しく	53.4	3	えう	75.1	えう	75.1
4	われ	56.7	われ	108.9	4	ちす	75.7	ちす	93.1
5	らち	61.5	らち	161.1	5	るる	79.6	るる	112.4

(a) 被験者1

(b) 被験者2

5-2. 誤認識指数

図8は、横軸を時間に、縦軸をくせ字セットにおける誤認識指数にとったグラフである。どの被験者においても、その誤認識指数は単調に減少していることがわかる。最初にマップを修正してそのあとは練習に集中する被験者2のようなケースもあれば、被験者3のように後半でもマップの大幅な修正を行っているケースも見られる。こうした学習の仕方には個人ごとに多様性があるとみてとれる。なお、被験者3の誤認識指数は相対的に高いが、これは似ている文字が多かった事に加え、入力枠に対して文字のサイズが小さかった事などが挙げられる。

5-3. 学習効果

図9は、横軸を時間に、縦軸を文字の認識率としたものであり、図10は縦軸を文字の入力時間としたグラフである。認識率については、上下動しながらも着実に向上していることが伺える。

入力時間については3人の被験者が減少、2人の被験者が最初と最後を比較すると微増となっている。被験者4の入力時間が途中大きく変化しているため、このログについて調査したところ、この被験者は誤認識指数をできるだけ下げるために各文字の下に記号を付記して、それぞれの文字の差異を大きくしようとしていた。結果として、それを暗記しなければならなくなり、途中入力時間が大幅に増えてしまっている。そのあと、少しくせ字マップを修正しながら学習を続けたところ、認識率の大幅な上昇と入力時間の大幅な減少を導くことができたようである。この被験者のような「妥協案」は筆者らの想定とは異なったものの、全体の傾向として本システムは認識率の向上と入力時間の減少に貢献しているといえることができる。

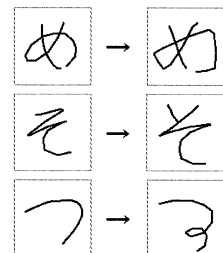


図7. 修正例

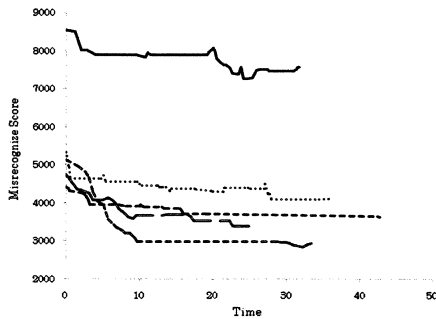


図 8. 誤認識指数の推移

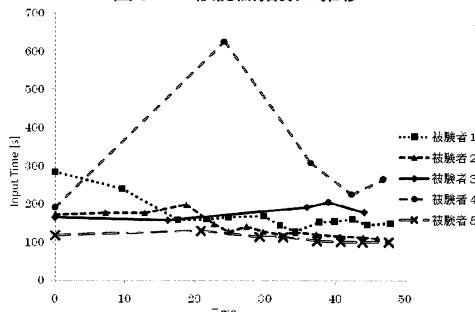


図 10. 入力時間の推移

5.4. 被験者の感想

被験者の意見として、暇があればゲームとしてやると面白いといった意見や、タブレットの練習にもなる、という意見が出た。また、難しい字以外は手書き入力をしたくない、という被験者もいた。自分の書きぐせの修正をしすぎて、時間内に全ての文字の特徴を覚えきれなかった被験者も複数いた。

表 1 を見ると、被験者 1・2 の両者において非類似度が改善されていない文字対がある。ここで列挙した文字はよく似ているため本来改善されるべきであるが、このように修正されないまま実験を終了しており、コンピュータが認識に苦勞する文字をユーザに伝え切れていないことになる。実際に被験者からも、「くせ字マップによる提示は直感的で文字の関係は分かりやすかったが、どの文字の修正から手をつけて良いか分からず、全部の文字を修正しなければならぬというような感覚を覚えた」という意見が出た。どの文字を優先的に修正すれば良いか提示するなど改良の必要があると感じた。また、書きぐせを変更する際に、どのように修正していけばよいかわからないので、コンピュータ側から修正案を提案して欲しい、と述べた被験者が多数いた。このような機能も今後実装していきたい。

今回は 45 分前後で実験を打ち切ったが、被験者に感想を聞いたところ、手が疲れたという意見が多かった。被験者の意見によれば連続して入力できるのは 1 時間くらいまでではないかという。ただし、視覚的に効果が見えるおかげで、学習に対するモチベーションが上がったという感想が多かった。

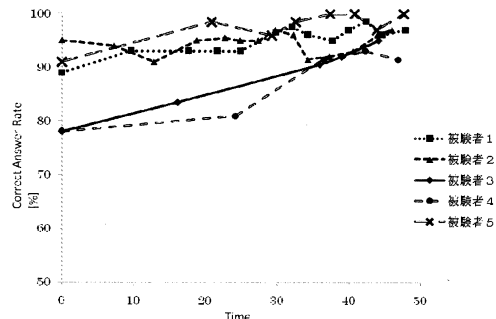


図 9. 認識成功率の推移

最後に、実験のインストラクションにおいて特に「歩み寄り」という言葉を使ってはいないにも関わらず、「コンピュータと対話しているような感覚をおぼえ、親近感がわいた」という意見があった。認識する側の立場を理解しそれに合わせようとする事が歩み寄りインタフェースの思想であったが、コンピュータによる認識作業を人間同士のコミュニケーションのように捉えるという意識の変化こそが、認識精度の向上に一番必要なものだったのではないかと筆者らは感じている。

6. おわりに

本稿では、認識器が認識に苦勞する部分についてユーザが妥協し意識的に認識し易い入力を与えると「歩み寄りインタフェース」を導入した手書き文字認識システムの試作および検証を行った。評価実験の結果、認識率の向上および入力時間の短縮が観測され、歩み寄りインタフェースの導入により入力の効率が上がる事が確認された。

手書き文字認識について歩み寄りインタフェースの有用性を検証してきたが、音声認識や自然言語認識など、その適用範囲は広いと考えている。今後は本稿で述べた文字認識システムの改良に加え、他の認識技術においても歩み寄りインタフェースが有効である事を実証していきたいと考えている。

参考文献

- [1] 奥村大樹, 内田誠一, 迫江博昭. オンライン手書き文字認識 HMM における座標情報と方向情報の利用法と効果, 電子情報通信学会技術研究報告, Vol.104, No.523(2004I209) pp. 25-30, 2004
- [2] 福居宏和. 手書き文字認識用辞書のネットワーク分散データ収集システム, 平成 14 年度末路ソフトウェア創造事業, 2002.
- [3] 緒方淳, 後藤真孝. 音声訂正: 認識誤りを選択操作だけで訂正ができる新たな音声入力インタフェース, 日本ソフトウェア科学会 第 12 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 (WISS 2004) 論文集, pp.47-52, 2004.
- [4] Toshiyuki Masui. Integrating Pen Operations for Composition by Example, Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'98), pp. 211-212, 1998.
- [5] <http://www.palm.com/>
- [6] 後藤真孝: "非言語情報を活用した音声インタフェース", 情報処理学会 音声言語情報処理研究会 研究報告 2004-SLP-52-7, Vol.2004, No.74, pp.41-46, 2004.
- [7] 舟久保登, 情報・電子入門シリーズ 11 パターン認識, 共立出版, pp.61-67, 1991
- [8] 斎藤貴幸, 宿久洋, 関連性データの解析法〜多次元尺度構成法とクラスター分析法〜, 共立出版, pp.97-123, 2006
- [9] DS 美文字トレーニング <http://www.nintendo.co.jp/ds/avmj/index.html>