

歩み寄りインタフェースの提案

篠原 祐樹^{†1} 宮下 芳明^{†1}

認識器における認識率改善の努力がなされる一方で、「100%の認識を実現することは不可能である」という前提に立った研究も始まっている。本論文では、認識器が認識に苦勞する箇所についてユーザ側が妥協をする、という思想をもとに、可能な限り少ないユーザの負担で高い認識結果を得ることのできる着地点を探すための「歩み寄りインタフェース」を提案する。また、文字認識においてこの「歩み寄りインタフェース」の実装を行ったシステムを試作し、その評価を行うことで検証を行っている。

An Interface for Compromise between Recognizer and Recognizee

YUKI SHINOHARA^{†1} and HOMEI MIYASHITA^{†1}

Although researchers still make efforts to lift the recognition rate of recognizer, recently there's been an increase of research on condition that it is impossible to realize a recognizer of 100% recognition rate. In this paper, we propose an interface design philosophy for compromise, in that the user (recognizee) compromises with the computer (recognizer) on the point that it strain to recognize, and the user finally gets great results with less work. We made a trial model in handwriting recognition, and from that evaluation we discussed the potency of this interface design philosophy.

1. はじめに

コンピュータを道具として利用するためには、我々は自分自身の思想や意志をコンピュータに理解させなければならない。コンピュータがそのような人間の意志や思想を理解する

ためには、音声認識¹⁾、歌声認識¹⁾、手書き認識²⁾、表情²⁾ やジェスチャ認識³⁾ などの技術が必要であるが、それらが直感的な入力形態であるほど認識精度が達成されておらず、それを向上させるべくアルゴリズムを改良する研究が進められている。

こうした認識率改善の努力がなされる一方で、「100%の認識を実現することは不可能である」という前提に立った研究も始まっている。緒方らは、人間同士のやりとりのような柔軟さを計算機が持ち合わせていないことを指摘し、音声認識における認識誤りが起こることを前提として、ユーザがそれを訂正するためのインタフェースを提案している⁴⁾。認識率がどこかで「頭打ち」になるのであれば、そこからの突破口を提供するのはインタフェースの観点からのアプローチではないだろうか。コンピュータによる認識が100%にならないのであれば、その残りの部分を向上させるために人間が入力の仕方を変えればよいのではないだろうか。筆者らはそう考えている。コンピュータの柔軟性や学習能力は近年の改良により向上しているものの、柔軟性や学習能力についてはやはり人間のほうが勝っている部分も多い。

たとえば、読み上げた数字を音声認識するシステムを考える。「7」を「シチ」と発音したときに「イチ」と誤認識されることが多いことが分かった場合、その誤認識を減らすために「7」を「ナナ」と読み上げることにしてもよいのではないかと、思うユーザは多いと考えられる。すなわち、認識率を向上させるために人間が入力の仕方を受協するのである。しかし、認識率を劇的に向上させる結果になるとユーザが理解していても、コンピュータが完全に識別できる新しい呼び方ですべての数字を読み上げる「全面的な妥協」を行うにはユーザの負担が大きすぎるであろう。

ただし、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⁵⁾、「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たちが、自分も使いたくて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」。我々がキーボードの早打ちにあこがれるような心理が、他の入力インタフェースにおいても起こりうるのである。しかしながら、とっつきにくい印象を最初に与えるキーボードとは異なり、より直感的な入力ができるような気にさせられてしまう音声認識や手書き認識においては、練習しようという意欲は起きないのではないだろうか。その証拠としてキーボードのタイピング練習ソフトは数多くあるが、音声認識や手書き認識における練習ソフトはほとんど存在しない。入力方式が直感的であればあるほど、ユーザ側に妥協するという心理は働かなくなるのである。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、こうした視座に立った研究はいまだ少ない。

^{†1} 明治大学理工学部情報科学科

Department of Computer Science, Meiji University

以上から、おそらく現在必要とされているのは、少ない労力で高い認識結果を得ることができる「妥協案」ではないかと筆者らは考えている。そこで本論文では、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ための「歩み寄りインタフェース」を提案する。本論文ではさらに、文字認識においてこの「歩み寄りインタフェース」の実装を行ったシステムを試作し、その評価を行うことで検証を行っている。

2. 歩み寄りインタフェースモデル

前述の音声訂正インタフェース⁴⁾で画期的な点の1つは、通常の認識結果とともに競合候補も表示し、しかも音声認識器にとって曖昧で自信がない箇所ほど多数の候補を表示する点である。このように認識する側がいわば弱点を見せてくれてはじめて、ユーザ側でも対処法を考えることができる。本論文で提案する歩み寄りインタフェースにおいても、まず認識器における曖昧さの可視化が必要であると考えている。

可視化によってコンピュータが認識に苦勞するところが明らかになれば、ユーザは妥協できるポイントを発見し、妥協案を思いめぐらすことができる。ただし、その妥協案を適用することでコンピュータ側の認識率が本当に向上するのか確認する必要がある。前章での数字の音声認識のたとえを再掲するならば、「シチ」が「イチ」と誤認識されないように「シ」を強調する読み方にしてみたら今度は「4」と認識されるようなことがあるかもしれない。つまり、実際に効果があるかどうかはユーザの予想の範疇を越えるところとなるので、シミュレーションが必要であり、試行錯誤の末に初めて良い妥協案をデザインすることが可能なのではないかと考えている。

最後に、ユーザが自らデザインしたその妥協案を実践できるかも考慮すべきであると考えている。そのときは許容範囲だと考えていた妥協案が、実践してみると思いのほか負担が大きかったり、1日経ったらもとのやり方に戻ってしまったりしては効果がない。そこで、ユーザが練習を行うことで、自ら設定した入力方法に慣れることが肝要である。この練習機構を設計する際には、認識率を上げるために不要な労力をできるだけかけさせない工夫が必要である。

まとめると、認識器における曖昧さの可視化、妥協案による効果のシミュレーション、そしてユーザがそのやり方で練習できる機構が必要なのであり、ユーザにかける負担ができるだけ少なくなるように設計するべきである。筆者らはこれを歩み寄りインタフェースモデルとして定義したい。

ユーザ側がコンピュータに歩み寄りという発想自体は決して新規なものではない。前述の

音声訂正インタフェース⁴⁾ではその訂正作業がユーザ側からの歩み寄りであると解釈することができる。あるいは、北山らによる音声スポット⁶⁾の研究においては、通常会話で不自然であるがコンピュータにとっては識別しやすい発声を行うという、ユーザからの歩み寄りを実現している。これらは、ユーザとコンピュータがお互いに歩み寄りという思想に基づいて設計されたインタフェースであるといえる。一方で、本論文で提案しているのは「ユーザとコンピュータがお互いに歩み寄るためのインタフェース」であり、どの程度歩み寄るか、そしてどのように歩み寄るかをユーザが自由に工夫して設定することができるうえ、自らの設定に合わせてユーザが練習可能な統合的な枠組みを提供しているところに新規性が存すると考えている。

3. 文字認識における歩み寄りインタフェースの設計

さて、前章で行った歩み寄りインタフェースの思想を文字認識の分野に適用させてシステムを試作した。今日、スタイラスで書き込んだ文字をテキストデータに変換する手書き文字認識技術は一般化しつつある。「くせ字登録」などのカスタマイズもできるようになり、HMMなどを用いて認識アルゴリズムも改良され⁷⁾、くせ字の学習データの収集技術⁸⁾、1ストローク描くごとに動的に候補が検索・表示されるインタフェース技術⁹⁾などによって日々進歩しているものの、依然として誤認識が多いのが現状であり、速度と信頼性においてキーボードより劣っていると考えられている。この背景は1章で述べた認識技術の現状と符合している。

そもそも我々が使用する文字の中には、「ぬ」と「め」など似た文字が存在しており、根本的な誤認識の要因となっている。これを回避すべく、逆に文字の形状自体を変えてしまう発想がPalmのGraffiti¹⁰⁾やXEROXのUnistrokes¹¹⁾である。これは、Tを∟のように、Kを×のように書くことで誤認識を減らすことに成功している。しかし、これらの文字の一部は認識精度の高さと引き替えに憶えにくい形状をしており、ユーザ側の負荷は高く、まさに「熟練者だけが使えるインタフェース」といえるのではないだろうか。本来ユーザごとに筆跡は異なるので、コンピュータが認識に苦勞する文字だけについてユーザが特徴的に書くようにすれば解決するはずである。そこで、歩み寄りインタフェースモデルの考え方に従ってシステムを構築したとき、文字認識においての「歩み寄り」がどの程度効果的にはたらくのかを検証した。

4. 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、最終的にはユーザが書きぐせ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できるようになるよう設計されている。

- (1) 筆跡の保存
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) 書き方の修正
この段階は、曖昧さの可視化と妥協案の効果のシミュレーションに該当する部分である。各文字の書き方を変えて、筆跡を修正することをコンピュータに宣言する。ここで変更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ぬ。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。

4.1 くせ字セット

作成したシステムでは、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。

入力された手書き文字を「くせ字セット」に追加する際、以下の手順に従ってデータの作成・登録が行われる。

- (1) 代表点の選出
入力された各ストロークについて、それぞれ 20 個の代表点を選出する。ストロークが n 個の座標点から構成されていた場合 $ni/19$ ($i = 0, 1, \dots, 19$) 番目の点を代表点として採用する。ただし、 n が 20 未満の場合、 $n \geq 10$ ならすべての点を代表点とし、 $n < 10$ ならその入力は破棄する。
- (2) 曲がり方向の算出
各代表点間においてストロークがどちらの方向に曲がっているか、計算を行う。点 (x_1, y_1) 、 (x_2, y_2) 間の曲がり方向 d は以下の式によって求める。

$$d = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right)$$

(3) 登録

以上の代表点および各代表点間の曲がり方向列をデータとして「くせ字セット」に登録する。

文字の認識は、入力された手書き文字と「くせ字セット」内の各サンプル文字との非類似度をそれぞれ求めることにより行う。非類似度は、2 つの文字における代表点の系列および曲がり方向の系列の差異を DP 照合法¹²⁾を用いて求め、それらを足し合わせることで算出している。なお、代表点間の差異はユークリッド距離、曲がり方向の差異は角度差である。これらの計算の結果、非類似度の最も小さくなる文字が入力された文字と解釈される。また、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文字であるということが出来る。

4.2 誤認識指数の算出

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、ユーザの書きぐせであるくせ字セットにおいて誤認識がどのように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持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」と結び付いており、その目標の達成度を表す 1 つの指標となる。

4.3 くせ字マップ

歩み寄りインタフェースでは、ユーザに対し「どのように歩み寄ってほしいか」をコンピュータ側が提示することが重要になるが、その手法として本システムでは、「くせ字マップ」なるインタフェースを導入した。

ここでは、くせ字セットに登録されている文字どうしの非類似度を空間における距離に見

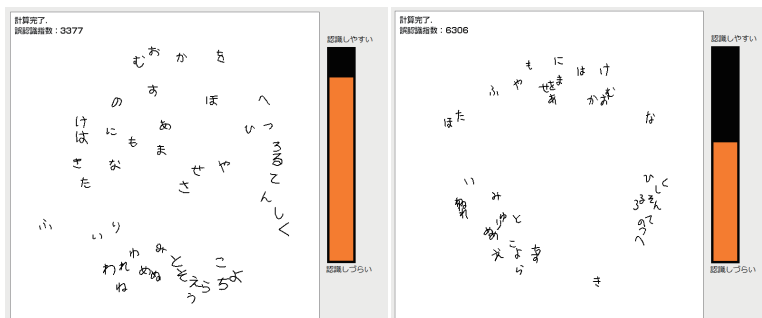


図 1 くせ字マップ
Fig.1 The Handwriting Map.

立て、似ている文字どうしがより近くに、似ていない文字どうしは遠くに配置して表示する(図1)。これによって、各文字どうしの類似関係を視覚的に理解できるようになる。実際の描画については動画を参照されたい¹³⁾。

本システムではコンピュータが認識に苦労する文字についてユーザ側が書きぐせを修正することができるが、この「くせ字マップ」により自分の書きぐせをどのように改善すべきかを把握し、またその修正の仕方を試すことができる。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書く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おり、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目安となる。

各文字の配置には多次元尺度構成法を用いている。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¹⁴⁾。

4.4 書きぐせの修正

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(図2(a))。たとえば、くせ字マップにおいて図3(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2(b)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3(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。

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、誤認識指数に従って右側

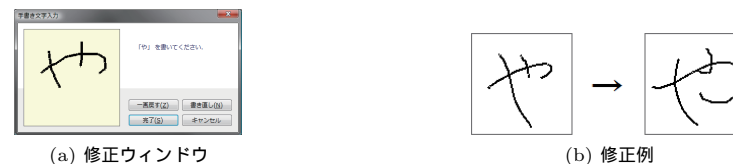


図 2 文字修正ウィンドウと修正例
Fig.2 Modify Window and example of modification.

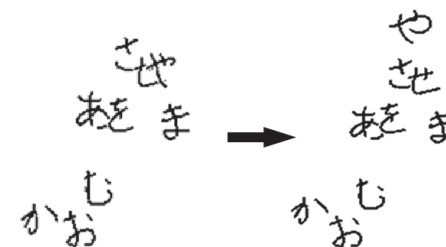


図 3 くせ字マップの修正
Fig.3 Modification of the Handwriting Map.

のゲージが上下動する。以降の文字認識においては、変更後の書きぐせがサンプルとして用いられるようになる。

4.5 認識率向上のためのユーザの歩み寄り方

前節の修正によって、ユーザはコンピュータに書きぐせの変更を宣言することができるが、今度はユーザがその自分で宣言した書きぐせに適應する必要が出てくる。これはユーザ側が歩み寄る部分において最も負担が大きな要素である。

このため本システムでは、自分が登録・修正した書きぐせを効率的に練習するための機構を用意している。これは、いわゆる漢字ドリル的なものとはかなり異なる。ドリル的な練習では、「美しい字」をお手本として、ときにはその文字をなぞるように練習し、一貫してその美しい文字が書けるように練習する。このような文字学習支援システムも多く¹⁵⁾、最近では「DS 美文字トレーニング」など^{16),17)}のソフトとして普及し始めている。こうしたシステムでは、いかにお手本に近い文字を書くことができ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。柔軟性を追求した武居らの漢字学習システム¹⁸⁾も、手本や評価箇所をカスタマイズできるのにとどまっており基本的なコンセプトは同じである。

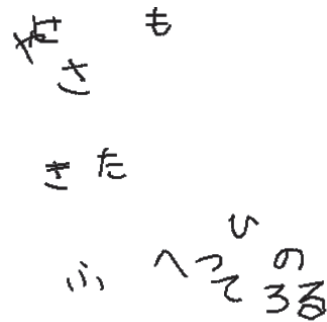


図4 くせ字マップの例
Fig. 4 Example of Handwriting Map.

しかし認識率をあげるための練習の場合、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ことができるか」である。文字の美しさも文字認識の観点からは弊害となることがあり、文字どうしの差異が広がるならむしろ字を汚くするほうが良いのである。

また、書く文字に一貫性がなくともよい。似ている字が多い文字のときは一貫性のない文字を書くと誤認識を引き起こす可能性があるが、ほかに似ている字の少ない文字については多少書きぐせにばらつきが出ても認識できるためである。この差を理解すれば、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ことができる。

たとえば、くせ字マップ上で図4のような配置を得た場合、以下のようなことが読み取れる（なお、ここでいう「手を抜く」とは、お手本として登録されている書きぐせと同じように書こうとする努力を怠ることを指す）。

- 「ふ」は似ている文字があまりなく、「ふ」にある程度近い形をした文字を入力すれば正確に認識される、すなわち手を抜いてもよい文字である。
- 「つ」は書き方によっては「へ」に似てしまうが、逆に「へ」と似ないように書こうとした結果「て」や「ひ」に近くなる可能性があり、手を抜くことのできない文字といえる。
- 「さ」は書き方によっては「せ」や「や」に近い文字となるが、それ以外に似ている文字が少ないので、この2文字と似ないようにさえすれば、後の部分で手を抜くことができる。

このように、文字によって勝手が違うため、必ずしもすべての文字の書き方を正確に習得する必要はなく、ユーザが最小限の負担で認識率の向上を図るためには「効率的に手を抜

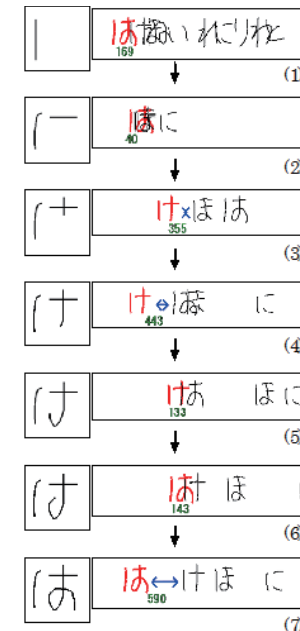


図5 認識状況モニタ
Fig. 5 Recognition Monitor.

く」必要がある。しかし、どの部分に関して手を抜けるの否かをユーザが判断するのは困難であるといえる。そこで我々は、入力している文字がど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。ただし、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側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、ほかにも「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1候補、「け」が第2候補と分かる。図5(2)–(5)のように入力を続けていると、「け」が第1候補となってしまうが、図5(6)のように3画目の最後の曲げ部では再び「は」が第1候補となっている。このような認識状況の推移を見せ



図 6 練習機構における得点
Fig. 6 Scoring in Practicing Phase.

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 1 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。

さらに、この練習機構では、第 1 候補と第 2 候補の非類似度（距離）をスコアとして算出し、二者間に矢印とともにリアルタイム表示している。このスコアはすなわち「いかに第 2 候補を突き放すことができたか」を意味している。

これまでの漢字ドリル的な文字練習機構の場合、入力している文字がターゲットとしている文字にどれだけ類似しているかを「良い文字」として評価してきた。図 6 (a), (b) における 2 つの「は」をこれらのシステムで比べると、(a) の「は」のほうがターゲットに近く「良い文字」とであると判断される。この文字はお手本に近い「きれいな文字」とであるということもできよう。しかしながら、この (a) は第 2 候補である「け」と誤認識されるリスクが相対的に高く、正しく認識されやすい文字は実は (b) なのである。前述のとおり、本システムでは第 1 候補と第 2 候補との非類似度が大きいほど認識されやすい文字として評価しているため、(b) のほうが多くの得点を表示している。

このように、前節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できるか」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よいのである。「くせ字マップ」をもと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。

以上のような作業によって、習得すべきポイントを最小限に抑え、歩み寄りによるユーザの負担を減らすことができる。ユーザは、書きぐせを変更しその特徴を理解することで紛らわしい文字の入力を可能な限り回避するよう努力する。コンピュータ側はその努力による

負担を最小限にするため、様々な手法でユーザの書きぐせ変更をサポートする効果的な情報や練習システムを提供する。このような相互の「歩み寄り」によって精度の高い文字認識に到達できるようなデザインを行うことができた。

5. 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 人を対象として、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などを用いた手書き文字入力の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持っていなかった。

5.1 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、実験ごとに異なるひらがな全 46 文字が 2 度ずつランダムに出現する意味をなさない計 96 文字の文字列を 1 つ画面に表示し被験者にその文字列を手書き文字で入力してもらい、その認識率（96 文字中何文字が正確に認識されたか）、入力に要した時間、4.2 節で述べた計算手法による誤認識指数を測定した（以下測定とよぶ）。なお、実験中これらの各種数値は被験者には提示しなかった。

その後、くせ字マップを使用した筆跡の修正および、4.5 節で解説した練習機構を利用し文字入力の練習を自由に行わせた。また、その間で適宜測定を行わせた。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。

6. 結果と考察

6.1 くせ字マップ

図 7 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（2 人分）である。これを見てみると、たとえば図 7 (a) における「し」「く」のペア、あるいは図 7 (b) における「え」「う」のペアはそれぞれ他方の被験者の上位に見られず、個人の書きぐせの傾向が異なることが分かる。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。

修正事例を見てみると、たとえば図 7 (b) の「ち」のように、類似してしまっている文字「す」に似ないように変形したり、図 7 (a) の「ち」や「れ」のように丸みを強調したりしているほか、角張った文字に変えたり、「ぬ」の最後部分を 2 度巻きに変えるといった形状変化だけでなく、書き順を変えたり、同じ線でも書く方向を変えるなど、多彩な修正が見られた。

	書きくせ修正前		最後の書きくせ修正後	
	文字対	非類似度	文字対	非類似度
1	ぬ ぬ	34.5	ぬ ぬ	329.9
2	ち す	48.9	ち す	675.6
3	し く	53.4	し く	53.4
4	われ	56.7	われ	108.9
5	ら ち	61.5	ら ち	161.1

(a) 被験者 1

	書きくせ修正前		最後の書きくせ修正後	
	文字対	非類似度	文字対	非類似度
1	われ	52.8	われ	65.1
2	ぬ ぬ	63.6	ぬ ぬ	73.9
3	え う	75.1	え う	75.1
4	ち す	75.7	ち す	93.1
5	る る	79.6	る る	112.4

(b) 被験者 2

図 7 非類似度が小さい文字ペア上位 5 ペア
Fig. 7 Top 5 pairs in the degree of similarity.

6.2 誤認識指数

図 8 は、横軸を実験開始からの経過時間に、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分かる。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 のように中盤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差があると見てとれる。

6.3 学習効果

図 9(a) は、横軸を実験開始からの経過時間に、縦軸を文字の認識率としたものであり、図 9(b) 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うかがえる。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番号を付記して、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、その数字を暗記しなければなくなり、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。

6.4 被験者の感想

被験者に対するインタビューの結果、5 人全員から試作システムの使用感は良好で、快適な文字入力が可能になったという回答を得た。

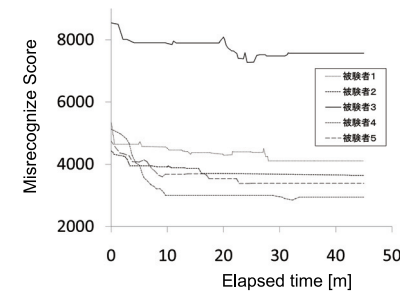


図 8 誤認識指数の推移
Fig. 8 Changes in the misrecognize score.

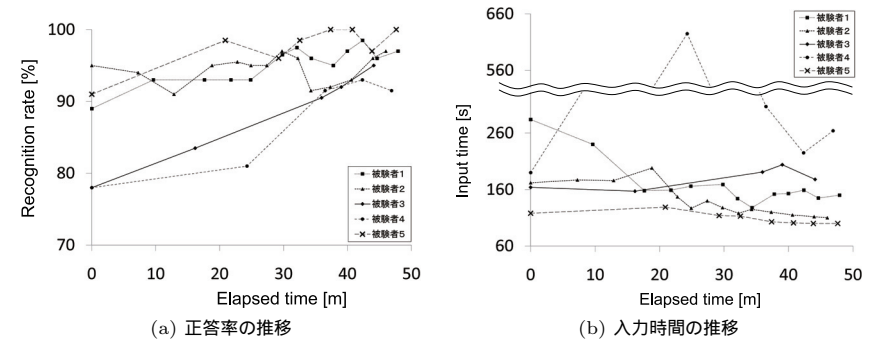


図 9 正答率と入力時間の推移
Fig. 9 Changes in the recognition rate and the input time.

それ以外の意見として、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いと思わない、という被験者も 1 人いた。自分の書きくせの修正をしすぎて、時間内にすべての文字の特徴を覚えきれなかった被験者も 2 人いた。

今回は 45 分前後で実験を打ち切ったが、疲労感について尋ねたところ、指が疲れたという意見が多かった。また、どの程度の時間連続して入力できそうか、という問いに対し 4 人の被験者が 1 時間前後と回答した。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 3 人から得られた。また普段の字が汚くなってしまわないか心配、というコメントもあった。

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いないにもかかわらず、「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の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とら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。

そういう意識の芽生えがあるのなら、たとえば我々が大人相手と子供相手とで話し方を変えるように、筆記においても書き分けができる可能性があるので、「字が汚くなってしまいかもしれない」という危惧も不要なのではないだろうか。ただし、これはあくまで予想であるため今後の実験によって明らかにしていきたい。1章の「熟練者だけが使えるインタフェース」の考え方に従えば、そういった字はむしろ「あこがれ」の対象として受け入れられる可能性も十分にある。

6.5 残された課題

試作システムでは50～100文字程度の比較的小規模な文字セットを適用範囲として想定しているが、文字数が大幅に増加した場合、「くせ字マップ」の表示や「くせ字セット」への手書き文字登録は困難となり、漢字など文字数が極端に多いものは事実上扱えない。これらの文字セットに対応するため、より高速かつ高精度な認識アルゴリズムの採用や、くせ字登録手法の変更、および「くせ字マップ」の表示手法の改良を行わなければならない。「くせ字マップ」は現在すべての文字を同じ条件の下で平面上に配置しているため、認識困難な箇所だけでなく認識が容易な部分まで表示していることになるが、認識しやすい文字についての情報は不要であり筆跡修正の効率を低下させる一因となる。このような情報を排除し筆跡修正の効率化を図るとともに、大規模文字セットの表示に対応するために、Focus+Context手法を用いた表示の導入を検討している。

実際の文字入力手法の面においては文字を最後まで入力しなくても、第1候補が意図した文字になった時点で次の文字入力に移れるようにしたり、認識候補を直接クリックすることで文字を入力できるようにしたりするなど、さらなる文字入力の効率化を行う余地がある。また筆跡の修正においても、どの文字をより優先して修正すればよいか、といった優先課題の提示や、どのように筆跡を修正すればよいか、コンピュータ側から修正案を提案してくれる機能を導入し、より効率的な歩み寄りを実現する必要がある。

本試作システムは膨大な文字種を扱う日本語の入力に十分に耐えうるものではなかったが、上記のような課題を解決することで本格的な実用化も可能になると考えられる。

7. おわりに

本論文では、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「歩み寄りインタフェース」を提案し、文字認識においてシステムを試作して評価を行った。筆者らが本論文で目的としているのは、文字認識の精度をあげるインタフェースを提案したいのではなく、あらゆる認識技術を向上させる可能性のある思想としてその背景にある考え方を理解してもらおうことである。本論文中で比喩として多く取り上げた音声認識をはじめ、「歩み寄りインタフェース」の適用範囲は広いと筆者らは考える。今回の文字認識システムを実用化へ向け改良していくほか、他の認識技術においてもこの「歩み寄りインタフェース」が有効であることを証明していきたいと考えている。

参考文献

- 1) 園田智也, 後藤真孝, 村岡洋一: WWW 上での歌声による曲検索システム, 電子情報通信学会論文誌 D-II, Vol. J82-D-II, No.4, pp.721-731 (1999).
- 2) 赤松 茂: 人間とコンピュータによる顔表情の認識 [I]: コミュニケーションにおける表情とコンピュータによるその自動解析, 電子情報通信学会誌, Vol.85, No.9, pp.680-685 (2002).
- 3) 入江耕太, 若村直弘, 梅田和昇: ジェスチャ認識に基づくインテリジェントルームの構築, 日本機械学会論文集 C 編, Vol.73, No.725, pp.258-265 (2007).
- 4) 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).
- 5) 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).
- 6) 北山広治, 後藤真孝, 伊藤克亘, 小林哲則: 音声スポット: 人間同士の会話中に音声認識が利用できる新たな音声インタフェース, 日本ソフトウェア科学会第 11 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 2003) 論文集, pp.9-18 (2003).
- 7) 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, pp.25-30 (2004).
- 8) 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 (2002). 平成 14 年度未踏ソフトウェア創造事業.
- 9) Masui, T.: Integrating Pen Operations for Composition by Example, *Proc. ACM Symposium on User Interface Software and Technology (UIST'98)*, pp.211-212

(1998).

- 10) Graffiti. <http://www.palm.com/>
- 11) Goldberg, D. and Richardson, C.: Touch-Typing With a Stylus, *Proc. ACM INTERCHI'93 Conference on Human Factors in Computing Systems (CHI'93)*, pp.80-87 (1993).
- 12) 舟久保登: 情報・電子入門シリーズ 11 パターン認識, pp.61-67, 共立出版 (1991).
- 13) Movie. <http://miyashita.com/publications/>
- 14) 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, pp.97-123, 共立出版 (2006).
- 15) 稲見 望, 富永浩之, 松原行宏, 山崎敏範: 筆記具の動きを学ぶ体感型書き方学習システム, 電子情報通信学会論文誌 D-I, Vol.J87-D-I, No.12, pp.1128-1135 (2004).
- 16) DS 美文字トレーニング. <http://www.nintendo.co.jp/ds/avmj/index.html>
- 17) 芳野可奈子, 高田雅美, 天白成一, 城 和貴: ニンテンドー DS を用いた書字学習トレーニングソフトの開発, 情報処理学会研究報告 MPS, 数理モデル化と問題解決研究報, No.19, pp.81-84 (2007).
- 18) 武居典子, 持田桂介, 未代誠仁, 中川正樹: 字形の評価箇所を指示できる手書き漢字学習システム, 情報処理学会研究報告 2005-CE-78, Vol.2005, No.15, pp.15-22 (2005).

(平成 20 年 3 月 25 日受付)

(平成 20 年 9 月 10 日採録)



篠原 祐樹

1986 年生まれ。現在, 明治大学理工学部情報科学科在学中。ヒューマンコンピュータインタラクションの研究に従事。インタラクション 2008 インタラクティブ発表賞受賞。



宮下 芳明 (正会員)

1976 年生まれ。2006 年北陸先端科学技術大学院大学知識科学研究科博士後期課程修了。同大学科学技術開発戦略センター研究員を経て, 2007 年より明治大学理工学部情報科学科専任講師, 2008 年より明治大学大学院理工学研究科新領域創造専攻デジタルコンテンツ系にも所属, 現在に至る。博士 (知識科学)。