

追加・平均化併用方式による候補内文字選択時学習機能付き オンライン手書き文字認識システムの開発と評価

横田登志美[†] 葛貫壮四郎^{††} 浜田 長晴^{†††} 桂 晃洋[†]
中川 正樹^{††††}

An On-line Handwritten Character Recognition System with Candidate-Selection
Triggered Learning by the Add and Average Method

Toshimi YOKOTA[†], Soushiro KUZUNUKI^{††}, Nagaharu HAMADA^{†††},
Koyo KATSURA[†], and Masaki NAKAGAWA^{††††}

あらまし オンライン手書き文字認識の認識率を向上させるため、ユーザの文字パターンの学習機能を開発・評価した。学習操作として、ユーザが誤認識時に 10 候補の中から正しい候補を選択した際、自動的に文字パターンを学習する方式を採用し、辞書サイズの増加を避ける工夫として、既存の認識用辞書パターンと学習する文字パターンを比較して追加または平均していく追加・平均化併用方式を開発した。約 1 万字の文章からなる評価用データ kuchibue から 3 人分を選んで評価した結果、学習機能なしでは認識率 82.0%、辞書サイズ 373 kByte であるのに対し、学習機能ありでは、認識率 87.2%、辞書サイズ 399.3 kByte と、26.3 kByte 辞書サイズが増加するにとどまり、認識率を 5.2 ポイント向上できた。また、辞書サイズの増加を抑える効果として、従来の 1 字種の登録パターン数に上限を設ける方式では、認識率 86.6、辞書サイズ増加分 27.6 kByte であり、提案方式の方が認識率は高く、辞書サイズの増加を抑えることができた。

キーワード 文字認識、オンライン手書き文字認識、筆者適応、学習アルゴリズム

1. ま え が き

コンピュータが小型化するにつれ、キーボードに勝る携帯性と操作性が好まれ、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。そこで、快適なユーザインタフェースを提供するため、ペンで書いた文字を認識するオンライン手書き文字認識の高度化が求められており、筆順の間違い、続け書き、また、個人的な癖の現れた文字などの認識率向上が重要な課題となっている。また、リソースに

制約がある小型システムへの組み込みを容易とするためには、処理が軽く、プログラムと字体表現辞書（以下、辞書）に要するサイズが小さいことが望まれる。

文字認識において認識率を向上させるためには、いろいろな文字の変形パターンを辞書に登録する必要があるが [1] ~ [3]、特定ユーザによる使用であれば、そのユーザが筆記する個人パターンのみ辞書に登録することで、小型辞書サイズながら認識率を向上できる [4] ~ [7]。多くの製品では、ユーザが意図的に文字パターンを辞書に登録する方式を採用しているが [8]、辞書に登録しようとするパターンを表示してユーザに確認を求めるために操作が煩わしく、そのためにユーザは追加をしたがらず、そのことが辞書サイズの増加に歯止めをかける結果になっている。本来なら、この煩わしさを排除し、かつ、辞書サイズの増加を抑える方式が望まれる。この点に着目した研究では、一部の字種に対する実験結果から、個人パターンの辞書への登録が有用な手法であることが報告されている [4]。個人パターンを手当たり次第に追加した場合に、辞書サイズが増

[†] (株)日立製作所日立研究所, 日立市
Hitachi Research Laboratory, Hitachi, Ltd., 1-1 Omika-cho
7-chome, Hitachi-shi, 319-1292 Japan

^{††} アイティードクターコーポレーション, ひたちなか市
IT-DOCTOR CORPORATION, 3600-150 Ooaza Nakane,
Hitachinaka-shi, 312-0011 Japan

^{†††} 倉敷芸術科学大学, 倉敷市
Kurashiki University of Science and the Arts, 2640 Nishinoura,
Tsurajima-cho, Kurashiki-shi, 712-8505 Japan

^{††††} 東京農工大学, 小金井市
Tokyo University of Agriculture and Technology, 2-24-16
Naka-cho, Koganei-shi, 184-8588 Japan

加し、ひいては処理性能に影響を及ぼす懸念があるが、文献 [4] では字種ごとの登録上限を設けることで、この問題を回避する提案をしている。また、文献 [5] では、字種ごとに制限を設けるのではなく、全体で登録上限を設け、登録上限数に達した後は、使用頻度の小さいものを削除して辞書の肥大化を防いでいる。上限を設ける方式以外に辞書サイズの増加を抑える方法として、特定ユーザの筆記した文字を平均化して登録する平均化方式が考えられ、少ない辞書で効率良く認識率を向上できると考えられる。文献 [7] は、OCR 文字認識において、標準辞書と個人パターンとの加重平均パターンを辞書として用いた方式を提案しており、この方法によれば、標準辞書を用いるより、また、個人パターンをそのまま用いるより認識率を向上できると述べている。

これらの報告から、追加方式、平均化方式とも、認識率向上に効果的と考えられるが、両者を比較した報告はない。また、平均化するパターン間があまりにも異なる場合、平均化することで無意味なパターンを生成して辞書化してしまうが、このことに考慮した方式は検討されていない。また、個人パターンを学習するための操作として、意図的に辞書に登録する操作ではなく、ユーザが候補文字の中から正しい認識結果を選択する操作を利用して学習する方法が自然でよいと考えられる。この方法では、ユーザに付加的な指示や操作を求めずに学習できるが、候補に含まれない誤認識は学習できない制約がある。こうした条件下で、学習効果がどの程度期待できるのかは、まだ報告されていない。

そこで、我々は、候補文字選択時に個人パターンを学習する候補内文字選択時学習機能付き文字認識を試作した [9]。本論文では、辞書サイズの増加を回避するための方式として、平均化するパターン間が大きく異なる場合を考慮した追加・平均化併用方式を提案し、従来方式としての追加方式、置換え方式との比較評価を示す。また、学習の操作上、誤認識すべてを学習するのではなく、正解が候補内に含まれる場合のみを学習するという制約条件のもとで、学習効果の評価を示す。評価は、一人分約 1 万 2000 字からなり、データ規模の大きい東京農工大学（以下、農工大）のオンライン s 手書き文字パターンデータベース TUAT Nakagawa Lab. HANDS- kuchibue_d - 96 - 02 [11]（以下、kuchibue）を用いて認識辞書の増加と認識率向上の関係を評価実験した。

以下、2. では提案する学習機能を提示し、3. で本学習機能を組み込む文字認識方式について述べる。4. では評価方法を説明し、5. でその評価結果を示す。そして 6. で考察を述べ、7. でまとめる。

2. 学習機能

2.1 候補内文字選択時学習機能

我々が開発した候補内文字選択時学習機能付き文字認識では、ユーザが候補文字の中から正しい認識結果を選択し直した際、そのパターンを学習する。例えば、図 1 (a) 学習前では、折れ曲がりの浅い“く”の文字を筆記した結果として“(”と誤認識しているが、ユーザが、候補文字の中から“く”を選択すると、この文字パターンを辞書に登録する。ユーザは、候補文字を選択することで、煩わしい確認操作なしで学習させることができる。その後は、図 1 (b) 学習後のように、折れ曲がりの浅い“く”の文字が筆記されても、正しく認識することができる。

図 1 の文字入力パレット及び文字認識は、Windows 98[®] で動作する。ペン入力 PC (Pentium 300 MHz 使用) での文字認識時間は、後述の高精度辞書を用いた場合でも 1 文字当り 0.15 s であり [10]、辞書に登録する処理時間はほとんど感じられない。この操作によれば、操作の煩わしさはないが、辞書を簡単に登録できるために、ユーザの知らぬ間に、辞書サイズが大きく増加してしまうため、辞書サイズをなるべく増加させない方式を検討する必要がある。また、ユーザが文字認識結果を候補文字の中から選択できる誤認識のみ学習するため、候補に含まない誤認識は学習できないが、この学習の操作上の制約条件で、学習効果がどの

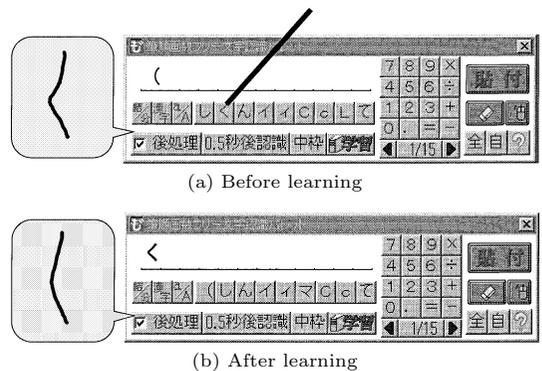


図 1 自動学習機能の操作

Fig. 1 Automatic learning operation.

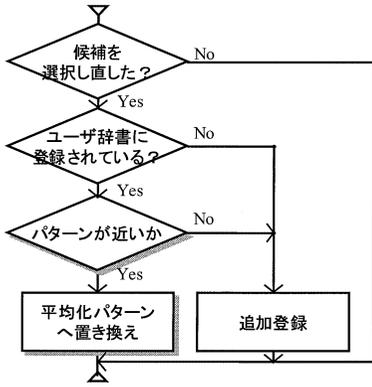


図2 追加・平均化併用方式
Fig.2 Add and average method.

程度期待できるのかを評価する必要がある。

2.2 追加・平均化併用方式

候補内文字選択時の学習方式の課題である辞書サイズの増加を回避するための手法として、追加・平均化併用方式を提案する。ここでは、学習前の標準辞書は、ROM (Read Only Memory) に搭載することが多いことから編集はできないものとして、辞書パターンを置き換える対象をユーザ辞書のみとした。提案する追加・平均化併用方式による学習処理フローを図2に示す。

まず、ユーザが候補を選択し直した際に、同じ文字がすでにユーザ辞書に登録されているかを調べる。もし登録されていれば、それらのパターンとこれから辞書に登録しようとするパターンとの間で、後述の文字認識による両パターンの特徴点間の距離値を求める。これがしきい値より小さい場合は、両者の平均パターンを生成して、既に登録されている辞書パターンをこれに置き換える。すなわち、平均化パターンへの置換えを行う。平均化を行うことで、ハネがある/ないなどパターンの変動を平均化してユーザの文字の特徴的な形状を辞書パターンとすることができる効果、及び辞書パターン数の増加を抑制する効果が期待できる。同じ文字がユーザ辞書に登録されていない場合、または、上記の判定で距離値が大きい場合は、追加登録とする。ここで、1字種の登録数には上限 L を設け [4]、その文字の登録パターン数が上限 L に達した後は、追加登録の際既に登録されている辞書パターンのうち、学習パターンと最も距離値が大きい辞書パターンを学習パターンで置き換えるようにした。これにより、学習の過程で書き間違いや操作ミスにより間違ったパターン

	パターン v	パターン u^*	平均パターン
a			
b			

図3 平均化の例
Fig.3 Examples of averaged character patterns.

表1 同形文字
Table 1 Categories of the identical pattern.

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; ツツ; ヤヤ; ユユ; ヨヨ; ワワ; カカ; ケケ; cC; . oOO; pP; sS; vV; wW; xX; zZ; エ工; カ力; トト; タタ; 又又; ハハ; て乙; え之; りリ; へへ; 7ク; 3ろ; hん; ち千; t七; テ〒; /ノ 1; XX; ++; ペペ; べべ; ニニ=; □ロロ; <<; 士士士; ○○0; ー ー; 2Z

が登録されても、いずれ削除されることが期待できる。追加登録か平均化パターンへ置換えかの判定を行う理由は、距離が大きすぎるパターンを平均化すると、意味のないパターンを生成することがあるからである。

例えば、図3のaのパターン v とパターン u_1 のように、形の似ているパターンであれば、平均化パターン $(v + u_1)/2$ を辞書に登録することにより、もとのパターン v とパターン u_1 のどちらにも近いため認識できるが、bのように、差が大きいパターンであると、平均化パターンに置き換えるとどちらも認識できなくなってしまう。判定のしきい値は、比較的丁寧に書いた文字データを複数個認識させ、正解認識した入力文字データを字種にかかわらず画数ごとに分類し、その画数ごとの平均距離値を用いた。パターンの平均化は、もとの辞書パターンと学習しようとするパターン間で、後述の文字認識の処理により対応する点を探し出し、1対1の重みで対応点座標の平均を得ることで行う。したがって、後に学習したパターンが最も学習パターンに反映される。また、1パターンで識別するのが難しい、形が同じか非常に似ている文字(以下、同形文字)間での誤認識については、ユーザ辞書を学習することによる認識率向上は期待できないため、同形文字間での誤認識では学習を行わない。表1にこの扱いをしている同形文字を示す。

3. 文字認識方式

上記の学習機能を組み込む文字認識方式は、我々が開発した楔文字モデルによる文字認識 [10] を用いた。この文字認識方式のフローを図 4 に示す。筆点座標列により取り込んだ入力パターンを、大きさや位置の正規化を行った後、折線近似する。この入力パターンと、あらかじめ同様に折線近似した辞書パターンとのマッチングを行い、入力パターンと辞書パターンとの距離値を計算する。辞書は、編集を行わない標準辞書と、学習の過程で追加登録や置換えなどの編集を行うユーザ辞書を参照する。マッチング後、距離値の小さい順に上位 10 候補を出力する。もし、第 1 候補が誤認識であった場合、ユーザは候補文字から正解文字を選び直すことになり、学習によりユーザ辞書は更新される。折線近似の処理は、まず、入力筆跡が x 軸または y 軸方向で反転する点を特徴点として抽出する。次に、特徴点間を結ぶ線分と元の筆跡までの高さがしきい値以上のときに、特徴点間の筆跡の長さを 2 分する位置に特徴点を置く。

この折線近似によれば、入力文字の続け書きした部分の特徴点で区切るように折線近似できる。マッチングの処理は、まず、折線の始終点座標に基づき、入力パターンと辞書パターン間の距離値の総和が小さくなる 1 対 1 対応の折線対応を探策する。次に、入力パターンと辞書パターンの双方の 1 対 1 対応から外れた折線について、筆記順が直前または直後の折線に対応があればこれと統合して 1 対 n 対応として解釈するか、または、対応から外れたと解釈するか距離値が小さくなる方の解釈を行う。

入力パターンと辞書パターンの双方について 1 対 n 対応として解釈することを繰り返した結果、 n 対 m 対応となる場合もある。これにより、対応する折線間の始終点座標の差 $d(v(i), u(j))$ が最も小さくなる対応 $\min \sum d(v(i), u(j))$ を探索する。その結果、対応する折線間の始終点座標の差 d の総和 $\min \sum d(v(i), u(j))$ と、対応から外れた入力パターンの線分 $v(k)$ の長さの総和 $\sum |v(k)|$ 、及び、対応から外れた辞書パターンの線分 $u(l)$ の長さの総和 $\sum |u(l)|$ を得て、パターン間の距離値 D とする。

$$\begin{aligned} \text{入力パターン } V & \\ &= \{v(1), \dots, v(i), \dots, v(k), \dots, v(l)\} \quad (1) \\ \text{辞書パターン } U & \end{aligned}$$

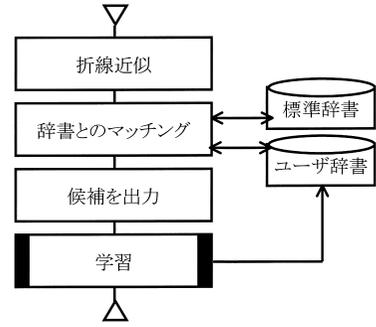


図 4 文字認識のフロー
Fig. 4 Flow of character recognition.

$$= \{u(1), \dots, u(j), \dots, u(l), \dots, u(J)\} \quad (2)$$

パターン間の距離値 $D(V, U)$

$$= \min \sum d(v(i), u(j)) + \sum |v(k)| + \sum |u(l)| \quad (3)$$

この認識方式の特徴は、上記の近似方式とマッチング方式にある。本折線近似によれば、文字の続け書き部分の特徴点で区切るように折線近似できるため、続け書き文字を楷書書き文字として表現できる。また、マッチング方式は、1 対 1 対応の折線対応探策に筆順情報を用いないため筆順に依存しない。そのため、一つの辞書パターンで続け書きや筆順違いの文字を認識でき、これにより、辞書サイズを小さくすることが可能である。

4. 評価方法

4.1 逐次学習評価

学習用データ、評価データに分けて評価する方法よりも、実際の使用状態に近い逐次学習評価方法 [4] により評価を行った。逐次学習評価方法とは、 K 文字からなる文字データを順に認識・学習していく。学習の過程で、ユーザ辞書が生成され、 k 番目 $\{k = 1, 2, \dots, K\}$ の文字は、 l 番目 $\{l = 1, 2, \dots, k - 1\}$ の文字を認識する過程で生成されたユーザ辞書を用いて認識される。認識結果が誤認識のうち、10 候補内に正解が含まれている場合に、ユーザは候補を選択し直すものとして学習を行った。すなわち、評価用データを一文字ずつ認識し、評価結果が誤認識でかつ 10 候補内に正解が含まれている場合にそのパターンを学習する。本方式は、候補文字内に含まれる文字のみを学習する点に特徴がある。したがって、効果が薄いのではないかとという懸念があるが、評価では、この点も明らかにしてい

く、この評価ツールは、図 1 に示した文字入力パレットとは別のアプリケーションであり、PC (Pentium 300 MHz, Windows 98 使用) 上で動作する。

4.2 比較評価する学習方式

3. に述べた文字認識に、2. に述べた追加・平均化併用方式及びその他の方式を組み合わせることで比較評価を行った。

(1) 追加方式

誤認識したパターンをユーザ辞書に追加登録していく。1 字種の登録数の上限 L を ∞ とした場合と、文献 [4] で提案された辞書サイズを抑えるために推奨される 3 とした場合で評価した。1 字種の登録数が上限 L に達した後は学習を行わない。

(2) 置換え方式

1 字種の登録数の上限 L を設け、上限 L に達するまでは学習パターンを追加し続けるが、達した後は、既に登録されている辞書パターンのうち、学習パターンと最も距離値が大きい辞書パターンを学習パターンで置き換える。上限 $L = \{3, 2\}$ とした場合で評価した。

(3) 追加・平均化併用方式

2. で提案した方式であり、1 字種の登録数の上限 $L = \{\infty, 3, 2\}$ の場合で評価した。1 字種の登録数の上限 L に達し、追加学習する場合は、(2) の置換え方式により学習する。

4.3 辞書

学習前の初期辞書としての標準辞書は、辞書サイズの違う 2 種、認識率を優先させた高精度辞書と、辞書サイズを優先させた小型辞書を用意した。

(1) 高精度辞書

漢字、非漢字とも 1 字種複数の変形文字パターンを登録し、認識率は良いが辞書サイズが 373 kByte と、他方の辞書に対し大きい。認識率を優先させたい場合に用いる。

(2) 小型辞書

漢字について 1 字種 1 パターンとし変形パターンを登録しないため、漢字の認識率は低い辞書サイズが 189 kByte と小さい。辞書サイズを優先させたい場合に用いる。

4.4 文字データ

評価用の文字データには、kuchibue から、認識率が低/中/高のデータとしてそれぞれ mdb0014, mdb0051, mdb0066 の 3 人分を用いた。この文字データは、文章からなる約 1 万字と、文章に現れなかった文字約 2 千字、合計 11962 文字からなってい

る。うち、記号と第 2 水準の文字合わせて 177 文字は辞書パターンになく、認識対象外としている。評価は、このデータの順に文字入力され、誤認識のうち、10 候補内に正解が含まれていた場合は、ユーザは候補を選択し直すものとして行った。

5. 評価実験結果

5.1 学習なしと提案方式の比較

図 5 に、学習なしの認識方式と、追加・平均化併用方式 (上限 $L = 3$) による学習あり認識方式の、認識率と辞書サイズの 3 人分データ平均値を示す。高精度辞書の場合は、学習なしでは認識率 82.0% が、学習ありでは 5.2 ポイント向上して 87.2% に、小型辞書の場合は、学習なしでは認識率 79.9% が、学習ありでは 4.9 ポイント向上して 84.8% であり、認識率向上に大きな効果がある。また、評価終了後の辞書サイズは、学習なし方式では増加分は 0 のため、標準辞書として高精度辞書サイズ 373 kByte、小型辞書サイズ 189 kByte のみであるが、学習ありでは、高精度辞書は 26.3 kByte 増加して計 399.3 kByte、小型辞書は 35.7 kByte 増加して計 224.7 kByte である。どちらの辞書の場合でも、増加分は、初期の標準辞書のサイズに対し 10~20% と小さい。すなわち、学習なしで高精度辞書を用いた場合より、学習による方式で小型辞書を用いた場合の方が、辞書サイズが 148.3 kByte 小さい上、認識率も 2.8 ポイント高い。

5.2 認識が改善されない例

本人の文字を学習しているにもかかわらず、認識が改善されない例は、以下の理由による。

(1) 類似文字

「し」が「レ」、「う」が「ラ」、「万」が「方」に誤認識するなど、形が似た類似文字間での誤認識は依然と

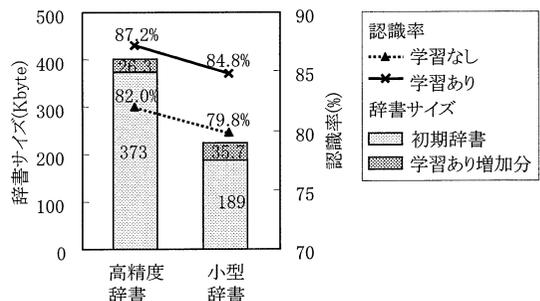


図 5 認識率と辞書サイズ

Fig. 5 Recognition rate and dictionary size.



図 6 改善しない例「万」
Fig. 6 Sample character pattern “万.”



図 7 改善しない例「間」
Fig. 7 Sample character pattern “間.”

して残る．図 6 に mdb0066 の誤認識した 1 回目（左）と 2 回目（右）の「万」のパターンを示す．この例のような形のばらつきが，類似文字との違いより大きいためと考えられる．こうした場合の対策としては，文脈後処理による改善が考えられる．

(2) 候補に入らない文字

辞書に登録されていない 177 文字以外に，略字パターンで筆記したため候補に入らないため，繰り返し誤認識する文字がある．こうした文字の例として，図 7 に門構えを略して筆記した「間」の例を示す．今回，操作上の制約から 10 候補に含む文字のみを学習しているため，候補に含まない誤認識は改善できない．

5.3 学習方式の違いによる効果

4.2 に述べた学習方式による評価結果を，表 2 から表 4 に示す．標準辞書に用いた辞書別に，高精度辞書を用いた場合を各表の (a) に，小型辞書を用いた場合を各表の (b) に示している．

表 2 は，前節に説明した学習なしの認識方式，及び文献 [4] で提案された追加方式の認識結果である．表 2(a) 高精度辞書による平均では，認識率は学習なしで 82% であるのに対し，追加方式で上限 $L = \infty$ では，87.2% に向上するが，辞書は 42 kByte 増加する．これに対し，文献 [4] で提案されたように，上限 $L = 3$ とし，上限に達した以降は辞書を追加しないようにすることで，辞書サイズは 27.6 kByte の増加に抑えることができるが，認識率も 86.6% と上限を設けない場合に比較して落ちてしまう．表 2(a) の小型辞書でも傾向は同様である．

表 3 は，置換え方式での結果である．置換え方式で上限を設けない場合は，表 2 の追加方式 ($L = \infty$) と同じ方式である．上限を設けた場合，表 3(a) 高精度辞書による置換え方式 ($L = 3$) では，辞書サイズは 27.7 kByte の増加に抑えることができ，認識率も

表 2 追加方式
Table 2 Add method.

(a) High precision dictionary

	学習なし		追加方式 ($L = \infty$)		追加方式 ($L = 3$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	76.8	0.0	83.0	55.0	81.3	40.0
mdb0051	79.4	0.0	86.6	44.0	85.9	26.0
mdb0066	89.8	0.0	92.1	27.0	92.7	17.0
平均	82.0	0.0	87.2	42.0	86.6	27.6

(b) Small dictionary

	学習なし		追加方式 ($L = \infty$)		追加方式 ($L = 3$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	74.1	0.0	81.0	60.0	78.7	51.0
mdb0051	76.9	0.0	84.2	42.0	82.6	35.0
mdb0066	88.8	0.0	91.2	28.0	90.9	26.0
平均	79.9	0.0	85.5	43.3	84.1	37.3

表 3 置換え方式

Table 3 Replace method.

(a) High precision dictionary

	置き換え方式 ($L = \infty$)		置き換え方式 ($L = 3$)		置き換え方式 ($L = 2$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	83.0	55.0	82.3	40.0	81.8	38.0
mdb0051	86.6	44.0	86.8	26.0	86.1	25.0
mdb0066	92.1	27.0	92.6	17.0	92.4	16.0
平均	87.2	42.0	87.2	27.7	86.8	26.3

(b) Small dictionary

	置き換え方式 ($L = \infty$)		置き換え方式 ($L = 3$)		置き換え方式 ($L = 2$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	81.0	60.0	80.0	51.0	79.3	48.0
mdb0051	84.2	42.0	83.3	35.0	82.7	33.0
mdb0066	91.2	28.0	91.0	26.0	90.8	25.0
平均	85.5	43.3	84.8	37.3	84.3	35.3

表 4 追加・平均化併用方式

Table 4 Add and average method.

(a) High precision dictionary

	併用方式 ($L = \infty$)		併用方式 ($L = 3$)		併用方式 ($L = 2$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	82.3	48.0	82.3	38.0	81.7	37.0
mdb0051	86.0	37.0	86.6	25.0	85.9	24.0
mdb0066	91.9	24.0	92.6	16.0	92.4	16.0
平均	86.7	36.3	87.2	26.3	86.7	25.7

(b) Small dictionary

	併用方式 ($L = \infty$)		併用方式 ($L = 3$)		併用方式 ($L = 2$)	
	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)	認識率 (%)	辞書増加 (Kbyte)
mdb0014	80.4	52.0	80.1	49.0	79.5	47.0
mdb0051	83.3	36.0	83.3	33.0	82.8	32.0
mdb0066	91.0	26.0	91.1	25.0	90.9	25.0
平均	84.9	38.0	84.8	35.7	84.4	34.7

87.2% と，追加方式 ($L = \infty$) と同率である．

表 4 は，2. に提案した追加・平均化併用方式である．追加・平均化併用方式は上限 L を 3 または 2 と設けた方が認識率はよく，従来の追加方式 ($L = \infty$) と置換え方式 ($L = 3$) と同率であり，しかも辞書サイズ

は 26.3 kByte の増加と最も小さく抑えている．併用方式の場合、上限 L を 3 または 2 と設けた方が、認識率はよくなり、しかも辞書サイズの増加を抑えることができる．高精度辞書の場合では、追加方式、置換え方式、併用方式で、いずれも ($L = 3$) の場合と比較すると、併用方式が辞書増加を抑えつつ認識率を上げることができている．小型辞書の場合では、認識率だけに着目すれば追加方式 ($L = \infty$) が最も認識率が高いが、いずれも ($L = 3$) の場合と比較すると、併用方式が辞書増加を抑えつつ認識率を上げることができている．

5.4 学習方式の違いによる学習パターン

平均化を行うことにより、ハネがある/ないなどパターンの変動を平均化してユーザの文字の特徴的な形状を辞書パターンとすることができる効果、及び辞書パターン数の増加を抑制する効果を期待している．そこで、この効果を調べるために 4. に述べた学習評価後の辞書パターンについて置換え方式、追加・平均化併用方式と比較した．図 8 に置換え方式、図 9 に追加・平均化併用方式のユーザ辞書パターンを示す．

ここでは 3 人の中で学習による認識率向上が中程度であった mdb0014 のデータを用い、小型辞書 ($L = 3$) の場合と比較を行った．その文字の認識率がよく、かつ、正解辞書パターンと入力パターンの距離値が小さいほど良い辞書パターンと考え、一定回数以上繰り返し出てくる文字のうちこれらが併用方式の方が良い“る”“日”と置換え方式がよい“で”を示している．まず、mdb0014 のすべての文字では置換え方式 ($L = 3$) は認識率 80.0%、距離値平均 272.4、距離値標準偏差 199.6、併用方式 ($L = 3$) は認識率 80.1%、距離値平均 272.1、距離値標準偏差 199.6 である．“る”は併用方式の方が認識率と距離値平均がともによく、パターンは図 8、図 9 に示すように方式によって異なる形状の辞書パターンが登録された．“日”はどちらの方式でも認識率は同じだが、置換え方式は辞書パターンが上限数の 3 個登録されたのに対し、併用方式は 1 個でありしかも距離値平均が他方に比べて小さい．“日”のパターン筆順最後の横棒 2 本に着目すると、置換え方式のパターンはつながったものと切れているものがあるが、併用方式の方では中間の形状をしている．

このことから、平均化によりユーザの文字に特徴的に現れる平均的な形状を辞書パターンとすることができる効果、及び辞書パターン数の増加を抑制する効果が確認できる．“で”は、距離値平均は置換え方式の

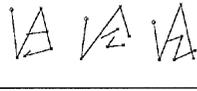
	認識率	距離値平均	標準偏差	学習辞書パターン
全ての文字	80.0	272.4	199.6	-
る	87.1	172.6	36.6	
日	85.3	220.8	44.2	
で	77.5	162.6	38.6	

図 8 置換え方式での辞書パターン例

Fig. 8 Dictionary patterns in replace method.

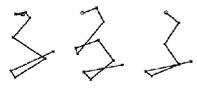
	認識率	距離値平均	標準偏差	学習辞書パターン
全ての文字	80.1	272.1	199.6	-
る	90.5	168.1	35.7	
日	85.3	218.0	48.6	
で	78.1	169.4	37.4	

図 9 追加・平均化併用方式での辞書パターン例

Fig. 9 Dictionary patterns in add and average method.

方がよい．置換え方式では誤認識のたびにパターンが次々に置き換わっていくため、その過程で併用方式に見られるようなパターンが登録されても置換えにより登録から外されたものと考えられる．

5.5 学習過程での認識率向上の効果

学習過程での認識率向上の効果調べるため、実際に筆記して感じる体感認識率として、逐次学習評価の過程での直前 1,000 文字の平均認識率の推移を図 10 に示す．例えば、横軸 1,000 の位置は、評価データ文字の 1~1,000 番目の文字、また、横軸 2,000 の位置は、評価データ文字の 1,001~2,000 番目の文字の平均認識率である．1,000 文字筆記する前の文字は、書き始めからその文字までの平均認識率を示している．書き始めは学習ありも学習なしも認識率は同じだが、mdb0014 では 1,000 文字程度、ほか 2 人分では 500

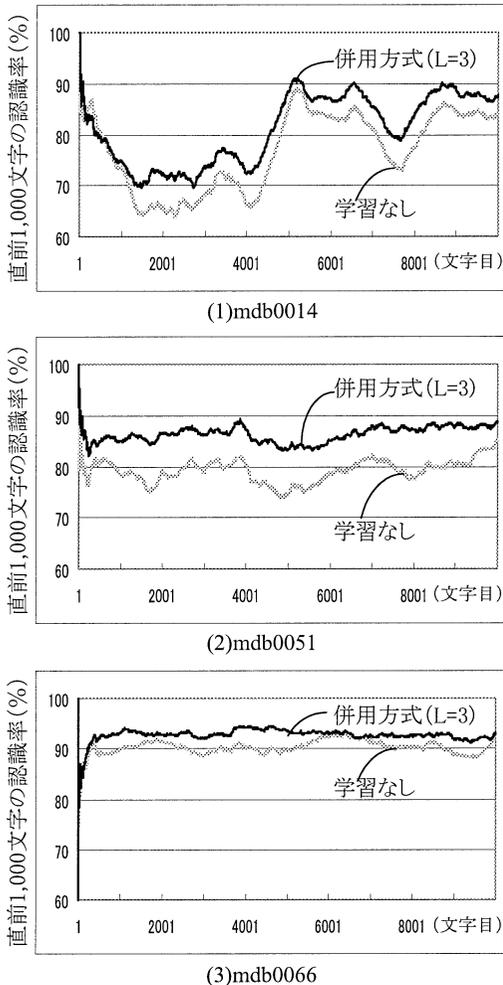


図 10 学習過程での認識率向上

Fig. 10 Increase of recognition rate in the learning process.

文字程度の時点で両者の認識率に差が出ている．実際は、どの程度同じ文字を繰り返し筆記するかにより異なるものの、今回使用した評価データのような一般的な文章を筆記するとすれば、1,000 文字に至るまでに効果が感じられると考えられる．実際に何文字程度で効果が感じられるかは体感テストが必要であるため別実験に譲る．

直前 1,000 文字に含む文字種は横軸位置により異なる．このため認識率は学習あり、なしの両者とも変動しているが、学習された文字の多い部分では、両者の差が開き、少ない部分では差が狭まっているものと考えられる．比較的認識率の良い mdb0066 では、学習

効果があまり見られない部分もある．

6. 考 察

特定ユーザによる使用が前提の場合、ユーザの文字を学習することは、辞書サイズを抑えつつ認識率を向上させるのに有効な手段といえる．また、学習操作を容易にするために、10 候補内に正解を含む誤認識のみを学習対象とした場合でも、略字などを除いて効果を認めることができた．

今回使用した 1 人分約 1 万字というデータでは、継続して大量に文字を筆記していく場合をシミュレーションできていると思われる．このように続けて文字を筆記するうちに筆記文字が変化してきた場合でも、より最近の文字を学習できる点で、追加方式と比較して、置換え方式あるいは追加・平均化併用方式は優れていると考える．学習方式の違いに着目すると、認識率を重視する場合、高精度辞書では、追加方式 ($L = \infty$)、置換え方式 ($L = \infty, 3$)、追加・平均化併用方式 ($L = 3$) が同率で認識率がよく、この中で辞書サイズを抑えながら認識率を向上させる点では、今回提案した追加・平均化併用方式が最も良い結果を得た．リソースに制約がある小型システムへの組込みを前提とすると、組込みを容易としかつマッチング処理の負担を軽くするためには辞書パターン数を抑えるすなわち辞書サイズを抑える必要があり、この場合には追加・平均化併用方式は有効な方式と考えられる．評価データ全体の数値で見ると辞書サイズを抑える効果は置換え方式もまた有効であるといえるが、個別に辞書パターンに着目するならば、追加・平均化併用方式では、平均化によるユーザの文字の特徴の平均的な形状を辞書パターンとすることができる効果と、辞書パターン数の増加を抑制する効果が認められる．

追加・平均化併用方式では、上限を設けない ($L = \infty$) よりも、上限を ($L = 3$) と設けた方が良い認識率を得ることができている．これは、平均化パターンを登録することで何らかの副作用が生じているものと考えられ、上限を設けて、使用しないパターンを削除することでこの副作用を回避できたのではないと思われる．追加・平均化併用方式による“で”の辞書パターンは認識率向上に寄与しないパターンを含むと考えられるが、こうしたパターンを削除して認識率向上に寄与するパターンを登録する方式を検討することで、更に辞書増加を抑え効率良く認識率を向上させていくことが可能と考える．また、今回は平均化方式として、

ユーザ辞書パターンとこれから学習するパターンとを1対1の重みで平均化する単純な方式をとったが、この方式については最適化する必要があると思われる。

7. む す び

オンライン手書き文字認識の認識率を向上させるため、ユーザごとの文字パターンの学習機能を開発・評価した。リソースの少ない小型システムに組み込むことを想定して辞書増加量を抑える学習方式を提案した。また、誤認識の際にユーザが候補を選択し直したタイミングでユーザの文字パターンを学習する操作を前提として、10候補内に正解が含まれているという制約を設けた評価を行い、従来方式としての追加方式、置換え方式との比較評価を示した。

得られた結果を以下に要約する。1) 誤認識した際に正解が候補内に含まれている制約を設け、3人分のデータで評価を行った。初期辞書として高精度辞書を用いた場合、学習機能なしでは認識率82%が、提案した追加・平均化併用方式による学習機能ありでは5.2ポイント向上して87.2%に、小型辞書の場合は、学習なしでは認識率79.9%が学習ありでは5.0ポイント向上して84.8%になり、認識率向上に大きな効果を示した。2) 辞書増加量を抑えながら認識率を向上させる方式を検討し、(1)追加方式、(2)置換え方式、(3)追加・平均化併用方式に、1パターン当りの登録上限 L を変化させて、認識率向上と辞書増加量を3人分のデータの平均値で比較した。最も認識率が向上したのは、認識率を優先する高精度辞書を用いた場合は同率で、(1)追加方式(上限 $L = \infty$)、(2)置換え方式(上限 $L = \infty, 3$)、(3)追加・平均化併用方式(上限 $L = 3$)であった。これを辞書増加量の面から見ると、今回提案した(3)追加・平均化併用方式($L = 3$)が最も増加量を低く抑えることができた。3)追加・平均化併用方式では、平均化によるユーザの文字の特徴の平均的な形状を辞書パターンとすることができる効果と、辞書パターン数の増加を抑制する効果を認められた。

今後の課題としては、学習の過程で認識率向上に寄与しない辞書パターンを削除する機能を追加すること、及び、類似文字への誤認識を改善するための文脈後処理と提案した学習機能との連携により認識率向上を図っていく。

文 献

- [1] 大倉 充, 塩野 充, “カテゴリ内クラスタリングによる多重辞書類似度法の辞書パターン作成の一検討,” 信学論

- (D-II), vol.J72-D-II, no.4, pp.499-506, April 1989.
- [2] 荒井正之, 奥田健三, 宮道尋一, “手書き漢字認識用マルチテンプレート辞書の作成方法,” 信学論(D-II), vol.J77-D-II, no.4, pp.890-891, April 1994.
- [3] 塩野 充, “多重辞書類似法による手書き漢字認識の基礎実験,” 情処学論, vol.27, no.9, pp.853-859, Sept. 1986.
- [4] 木村義政, 小高和巳, 鈴木 章, 佐野睦夫, “携帯型ペン入力インタフェース用辞書の学習,” 信学論(D-II), vol.J84-D-II, no.3, pp.509-518, March 2001.
- [5] 秋山勝彦, 岩山尚美, 田中 宏, 石垣一司, “オンライン手書き文字認識のためのテンプレートキャッシングによる筆者適応手法,” 信学技報, PRMU2000-210, March 2000.
- [6] 木村義政, 若原 徹, 杉村利明, “有効範囲付テンプレート追加法による辞書学習,” 信学技報, PRMU98-37, June 1998.
- [7] 内藤誠一郎, 増田 功, “個人性に着目した手書き漢字認識,” 信学論(D), vol.J67-D, no.4, pp.480-487, April 1984.
- [8] Microsoft Corporation, Pen Services プログラマーズガイド for Microsoft Windows 95, アスキー, 東京, 1995.
- [9] T. Yokota, S. Kuzunuki, K. Gunji, and N. Hamada, “User adaptation in handwriting recognition by an automatic learning algorithm,” HCI International 2001, Usability Evaluation and Interface Design, vol.1, pp.455-459, Aug. 2001.
- [10] 横田登志美, 葛貫社四郎, 郡司圭子, 桂 晃洋, 浜田長晴, 福永 泰, “筆順回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,” 情処学論, vol.44, no.3, pp.980-990, March 2003.
- [11] 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, Sept. 1995.

(平成16年1月23日受付, 9月9日再受付)



横田登志美 (正員)

昭59茨城大・工・情報卒。同年(株)日立製作所入社。日立研究所にて、オンライン手書き文字認識・図形認識の研究に従事。東京農工大大学院博士課程在学中。情報処理学会会員。



葛貫 壮四郎

昭 53 茨城大短大・電気卒。平 13 放送大学卒。昭和 37 (株)日立製作所入社。日立研究所にて、電力系統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。平 14 日立製作所を定年退職し、アイティードクターコーポレーションを設立。現在、中小企業向けの IT 支援に従事。



浜田 長晴 (正員)

昭 40 鹿児島大・工・電気卒。同年 (株)日立製作所入社。日立研究所にて、グラフィックス&イメージ処理プロセッサとマルチメディアシステムの研究開発に従事。平 13 より倉敷芸術科学大学教授。工博。情報処理学会、画像電子学会、IEEE、SID 各会員。



桂 晃洋

昭 52 京大・工・電気卒。昭 54 同大学院修士課程了。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、コンピュータアーキテクチャ、制御システム等の研究に従事。工博。情報処理学会、電気学会、IEEE、ACM 各会員。



中川 正樹 (正員)

昭 52 東大・理・物理卒。昭 54 同大学院修士課程了。同大在学中英国 Essex 大学留学 (M. Sc. in Computer Studies)。昭 54 東京農工大・工・助手。現在、教授。オンライン手書き文字認識、手書きインタフェースなどの研究に従事。理博。