

連想整合法に基づく高速文字認識アルゴリズム†

孫 寧^{††} 阿 曾 弘 具^{††} 木 村 正 行^{††}

文字認識についての研究が30年ぐらい前から行われ、種々の成果が得られているが、高度な実用化に耐えるにはまだ多くの問題を解決しなければならない。その代表的な問題点は認識率および認識の速度である。本論文では、一般的な全数整合法にかわる高速、高精度な大分類アルゴリズムとして、連想整合法を提案する。この方法はベクトル量子化を用いたパタン整合法に基礎を置きながら、文字の部分情報を積極的に利用して、高速化を計っているところに大きな特徴がある。連想整合法を用いた認識実験では、プリンタによる印刷漢字(字種 2965)に対し、1位認識率を99.87%に確保しながら、スピードを全数整合法の5倍程度速くできることを示す。また、手書き文字(字種 3036)に対し、100位累積認識率を全数整合法と同じ99%台に確保しながら、スピードを約4倍程度速くできることを示す。

1. はじめに

文字を認識する技術は計算機の発展と共に、これまで数多くの研究が行われ、大きな成果を得ている。現在、特定の分野においてはOCRが実用されている。一方、認識方法も古典的なものからファジー理論やニューラルネットを用いた方法まで様々なものが提案されている。これらの方法を文字のどのような特徴を捉えているかという観点から分類すると、大きくパタン整合法^{2), 8), 9)}と構造解析法^{3), 6)}に分けることができる。前者は、文字画像の大局的あるいは統計的特徴を基にアルゴリズムを構成する方法であり、ノイズや多少の変形には強いが、類似文字の識別が難しいという特徴を持つ。これに対し、後者の方法は文字画像の構造情報や文字を構成する線分間の関係に基づいて字種間の細部の相違を見分ける方法であり、類似文字の識別が可能であるが、文字の潰れやノイズに弱く、計算が非常に複雑であるという特徴を持つ。

このように両手法には一長一短があり、高精度化のためには、パタン整合法と構造解析法とも適切に結合する必要があるように思われる。すなわち、パタン整合法の高速性を生かして、候補字種を絞り込み、少ない字種に対して、構造解析法を適用して、実用的な時間内で正確な認識結果が期待できる。候補の絞り込みを大分類、その結果に対する認識アルゴリズムの適用を細分類と呼ぶ。このようにすることで、両手法の長所を生かし、短所を補うことができる。しかしながら、パタン整合法を漢字認識に適用する場合、漢字の

字種数が多いため、計算時間が長くなるという問題点がある。例えば、今回の実験に用いた電総研のデータベースの場合、一つのセットに3036種類(仮名71文字、漢字2965文字)もの文字が含まれている。従来から知られている一般的なパタン整合法である全数整合法は未知入力を全字種の標準パタンと照合し、候補を選ぶという方法であり、字種数に比例する時間がかかる。本論文では、この点を改良し、パタン整合法の長所であるアルゴリズムの簡単さ、比較的高い認識率を与えるという特徴を生かしつつ、高速、高精度に文字を認識する新しいアルゴリズム、すなわち、連想整合法を提案する。この方法はベクトル量子化を用いたパタン整合法に基礎を置いているが、文字の部分情報を積極的に利用して、高速化を計っているところに大きな特徴がある。

パタン整合法では、どのような特徴量を用いるかがその性能を支配する鍵となる。本研究では、方向線素特徴量を用いる。文字線分の方向性に着目した特徴量としては、従来、セル特徴⁴⁾、方向寄与度特徴量⁵⁾、方向密度ベクトル⁷⁾、などが提案されており、それぞれ、局所的エッジの大きさの方向別積分、各点から見た8方向への黒画素ヒストグラム計算、各点のまわりのひろい部分の方向線分の数え上げに基礎をおいている。方向線素特徴量は小さな領域の方向線素の数え上げに基礎を置くもので、それを求める方法が簡単で、高速性がある。また、これまで活字認識¹⁰⁾(JIS第一水準など合計3303文字)および字種を限定した手書き認識¹²⁾(200文字程度)において、その有効性が確認されている。

高速、高精度を目指した文字認識システム全体の構成を図1に示す。文字認識システムは前処理、特徴量抽出、連想整合法による大分類と他の方法による細分

† High Speed Character Recognition Algorithm Using Associative Matching Technique by NING SUN, HIROTOMO ASO and MASAYUKI KIMURA (Department of Information Engineering, Faculty of Engineering, Tohoku University).

†† 東北大学工学部情報工学科

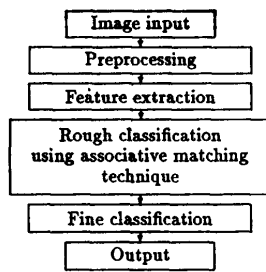


図 1 文字認識システム

Fig. 1 Character recognition system.

類の四つの部分によって構成される。大分類の結果に対する細分類については、機会を改めて報告する予定である。

本論文の構成は、まず、第 2 章で前処理と特徴量抽出について述べた後、第 3 章で連想整合法のアルゴリズムについて述べる。第 4 章と第 5 章では、活字および手書き文字を対象に行った認識実験の結果を示し、全数整合法と比較を行う。第 6 章では、本アルゴリズムの有効性を確認し、結論をまとめる。

2. 前処理および特徴量抽出

2.1 前処理

前処理は、ノイズ除去・スムージング、正規化、細線化および方向線素化の四つの過程からなる。ノイズ除去・スムージングでは、孤立したノイズの除去や文字の凹凸の修正を行う。正規化では、文字サイズを 64 ドット×64 ドットの大きさに拡大する。正規化の手法としては、活字および手書き文字の特徴を考慮して、活字の場合線形正規化を、手書きの場合非線形正規化を採用している¹¹⁾。非線形正規化¹²⁾は文字のストローク密度を利用して、正規化を行う方法であり、その長所としては、文字内部の複雑な部分を拡張する効果、個人の癖（はね、伸びなど）を矯正する効果、ノイズによる影響を軽減する効果などがある。細線化では、線の幅を数ドットから 1 ドットに細くする。これは線の太さの影響を取り除くためである。方向線素化では、線分を構成する各黒画素にその点での線分方向を対応させる。線分方向は、縦、横、 $\pm 45^\circ$ の四つに分類し、方向線素と呼ぶ。具体的には、対象画素を中心とする 3×3 の領域のドットパターンを見て、最も自然な方向と考えられる四つのうちの一つの方向線素を対応させる。例えば、上下にのみ黒画素があれば、縦であり、左右にのみ黒画素があれば、横である。詳しくは割愛するが、 3×3 のドットパターン（中心は黒画素）のすべて (2^8 個) に対して、四つの方

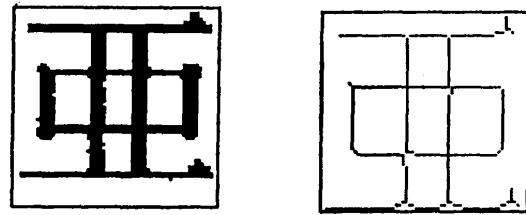
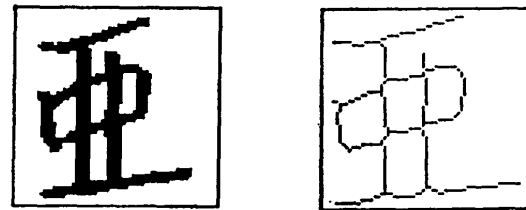
(a) 活字
(a) Printed character.(b) 手書き文字
(b) Hand-written character.

図 2 前処理の例

Fig. 2 Examples of preprocessing.

向線素を割り当てた対応表をあらかじめ作っておいて、その表に基づいて方向線素化が実行される。図 2 に前処理の一例を示す。入力文字パターンと線素化されたパターンを示している。

2.2 特徴量の抽出

方向線素特徴量の求め方を図 3 (a) に示す。まず、 64×64 ドットからなる文字画像領域を 8×8 ドットに分割する。さらに、図の左上の部分に示すように、 16×16 ドットを半分ずつ重複させてとっていく。そうすると、全部で 49 (7×7) 個の小領域が得られる。図 3 (a) の下側の図は 49 個の小領域に分割されたうちの一つの領域の拡大図である。この拡大図で示したように各小領域がさらに四つの部分領域 A, B, C, D に分割される。A は小領域の中心に位置する大きさ 4×4 の部分領域であり、B は A を除いた大きさ 8×8 の部分領域であり、C は A と B を除いた大きさ 12×12 の部分領域であり、D は 16×16 の小領域から A, B および C を除いた部分領域である。各部分領域 A, B, C, D にそれぞれ重み係数 4, 3, 2, 1 を対応づける。重みの値は実験から得られたものである。

各小領域の特徴量を次の 4 次元ベクトル (x_1, x_2, x_3, x_4) で定義する。各要素は次の式で値を定める。

$$x_i = 4x_i^{(A)} + 3x_i^{(B)} + 2x_i^{(C)} + 1x_i^{(D)} \quad (1)$$

ここに、添字 i の取る値は 1, 2, 3, 4 で、それぞれ縦、横、 $+45^\circ$ 、 -45° の 4 種の方向線素を意味し、 $x_i^{(A)}$, $x_i^{(B)}$, $x_i^{(C)}$, $x_i^{(D)}$ は各部分領域 A, B, C,

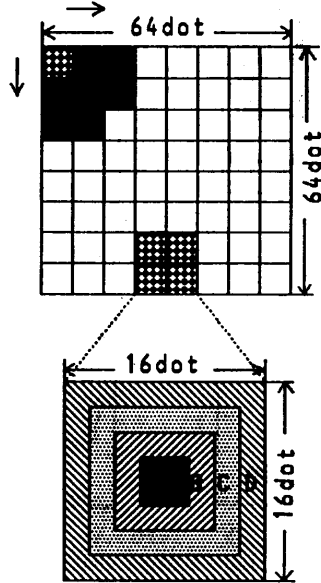


図 3(a) 方向線素特徴量

Fig. 3(a) Directional element feature.

Dに存在する方向線素 i の個数である。係数は各部分領域における重み係数である。このように重みを小領域の中心部分で大きくし、外側へ行くにつれ小さくすることによって、線分の位置ずれによる影響を抑制している。

49個の小領域の並び順は左上角の小領域を1番として、左から右、そして上から下へと順番に番号をつけたものとする。方向線素特徴量はこのような49個の小領域の特徴量をならべたもので、196(49×4)次元のベクトルとして定義する。この方向線素特徴量を特徴ベクトルとも呼ぶ。

文字“千”をサンプルとして、小領域の特徴量抽出の具体例を示す。図3(b)の上の太い線で書かれた枠に前処理後の図形が入っている。この枠の上中央部に細い線で囲んだ小領域(4番目の小領域)があり、その下の図形がこの小領域の拡大図である。以下、この小領域ベクトルの抽出方法について説明する。拡大図の中に線素化後の文字図形の一部が入っている。図からわかるようにこの部分は縦と横の線素によって構成され、各部分領域には次のように線素が分布している。部分領域Aには縦線素が4個、横線素が2個、Bには縦線素が2個、横線素が5個、Cには縦線素が2個、横線素が4個、Dには縦線素が2個、横線素が4個というふうに分布している。したがって、式(1)によって、縦線素に対応する x_1 、横線素に対応する x_2 の値はそれぞれ

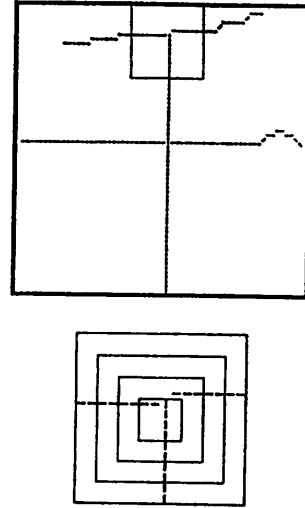


図 3(b) 特徴量の抽出例

Fig. 3(b) Sample of feature extraction.

$$x_1 = 4 * 4 + 3 * 2 + 2 * 2 + 1 * 2 = 28$$

$$x_2 = 4 * 2 + 3 * 5 + 2 * 4 + 1 * 4 = 35$$

となる。また、 $+45^\circ$ 、 -45° の線素がないため、

$$x_3 = x_4 = 0$$

となる。よって、この小領域の特徴量は(28, 35, 0, 0)になる。

3. 連想整合法のアルゴリズム

3.1 連想整合法

連想整合法はちょうど人間が文字の各部分によって、文字全体を思い出すのと同じように、文字画像の部分情報から候補を連想する方法である。この部分情報は分割された49個の小領域に含まれている情報を指す。認識の結果は各小領域の連想度合を総合評価することによって得られる。連想整合法は、次の二つの段階からなる。

(1) 連想整合用標準パターン辞書の作成

- 標準パターン構成用のデータセット(字種数を K とし、各文字あたり十分なサンプルを用意する)について、各文字の特徴ベクトルを求め、その平均を求める。平均ベクトルと呼ぶ。
- 全字種について、その平均ベクトルの小領域ごとの部分ベクトルをクラスタリングし、 L 個のクラス

$$C_1^{(m)}, C_2^{(m)}, \dots, C_L^{(m)} \quad m=1 \sim 49$$

に分ける。クラスタリングの方法として、LBG法¹⁾を用いる。

- (c) 各クラス $C_i^{(m)}$ の代表ベクトル $x_i^{(m)}$ を
- $$x_i^{(m)} = E(C_i^{(m)})$$
- と定める。ただし、 $i=1\sim L$, $m=1\sim 49$, $E(C)$ は C に属するベクトルの平均を意味する。代表ベクトルを標準パターンベクトルと呼ぶ。

(2) 候補選出

- (a) 未知入力文字に対して特徴ベクトル x を求める。
- (b) x に対して、小領域 m の部分ベクトル $x^{(m)}$ と各標準パターンベクトル $x_i^{(m)}$ ($i=1\sim L$) との距離を求める。距離尺度 d は後述する d_e を用いる。

$$D_i^{(m)} = d(x^{(m)}, x_i^{(m)})$$

- (c) $D_i^{(m)}$ を小さい順にソートする。

$$D_{i_1}^{(m)} \leq D_{i_2}^{(m)} \leq \dots \leq D_{i_L}^{(m)}$$

- (d) ソート結果に示されるクラス $C_{i_n}^{(m)}$ ($n=1\sim L$) に属する字種 j に対して、小領域 m における得点を次のように与える。

$$P_j^{(m)} = a(n) \quad (2)$$

ただし、 $a(n)$ は n に関して減少関数で、 $a(n) \geq 0$ である。 $a(n)$ を連想整合度という。

- (e) 各小領域における得点を加算し、その字種 j の得点とする。

$$P_j = \sum_{m=1}^{49} P_j^{(m)}$$

- (f) P_j の高い順に候補 j を出力する。

標準パターン辞書作成では、各小領域において似ている文字がクラス化され、各クラスに1個の代表ベクトルを対応させる操作を行っている。これは各小領域における部分パターンを分類して、小領域におけるいくつかの標準パターンを定めていることを意味する。候補選出では、各小領域ごとに未知文字に対して分類された標準パターンを連想させ、その整合度を点数で評価し、最終的に各小領域における連想の総合得点の順に候補を決定している。

3.2 距離尺度および連想整合度

距離尺度として、ユークリッド距離、重み付きユークリッド距離（厳密にはその自乗）を考える。 x, u をそれぞれ未知入力ベクトルとある文字の標準パターンベクトルとすると、ユークリッド距離と分散を考慮した重み付きユークリッド距離は、それぞれ次のように定義される。

ユークリッド距離：

$$d_e(x, u) = \sum_{i=1}^n (x_i - u_i)^2 \quad (3)$$

重み付きユークリッド距離：

$$d_w(x, u) = \sum_{i=1}^n w_i (x_i - u_i)^2 \quad (4)$$

ここで w_i は重み付きユークリッド距離の重みで、各文字の次元ごとに次のように定義する。

$$w_i = A \frac{1}{\sum_{j=1}^n \frac{1}{v_j}} \quad (5)$$

で与える。ただし、 A は定数、 v_i は標準パターン構成用データセットにおける各次元の分散、 n は次元数である。この重みはパタンの線の位置が変化しやすいところ（分散が大きい）では小さく、線の位置が安定なところ（分散が小さい）では大きくなるように定めている。これにより、文字パターンを区別するために重要である部分に重点が置かれた距離が定まる。

連想整合度 $a(n)$ はよく似ているほど高得点にするもので、いろいろ考えられるが、今回は次のように定めた。

$$a(n) = \begin{cases} L - (n-1) \times \text{減点幅} & (n \leq H) \\ 0 & (n > H) \end{cases} \quad (6)$$

L はクラス数である。パラメータ H (得点クラス数) および減点幅は認識率がよくなるように予備実験によって定める。その結果、活字の場合は上位5個のクラス ($H=5$) に点数を与えることが適当であり、一方、手書きの場合はすべてのクラス ($H=L$) に点数を与えることが適当であることがわかった（手書き文字に関する予備実験は後述する）。減点幅に関しては、どちらの場合も1が適当であることがわかった。

3.3 処理速度

全数整合法（テンプレート法とも呼ぶ）は未知入力とあらかじめ字種ごとに用意された標準パターンとの距離を計算し、その距離の小さい順に候補を出力する方法である。一方、連想整合法は未知入力と各小領域の代表ベクトルとの距離計算をし、距離の小さい順に点数を与え、その合計である総合得点の高い順に候補を出力する方法である。全数整合法および連想整合法における1文字の処理時間をそれぞれ T_t (Template Matching), T_a (Associative Matching) とすると、次のように近似できる。

$$T_t = P + 196K(s+w) + cK \log K \quad (7)$$

$$T_a = P + 196Ls + 49cK \log L$$

$$+49a(K/L)H+cK \log K \quad (8)$$

ただし、

s : 差の2乗+1回の加算の時間

w : 重みのかけ算の時間

c : 1データの比較, ストア時間

a : 1データの点数加算時間

P : 1文字の前処理時間

L : 小領域におけるクラスタ数

H : 得点を与えるクラスタ数

K : 全字種数

両者の第2項は距離計算の時間である。 T_a の第3項は連想整合度を与えるためのソートの時間である。ソートにはクイックソートを使用し、その比較回数のオーダーは $K \log K$ である。得点の計算は正得点のクラスタ内の文字に対してのみ加算を実行するように実現したので、計算時間は平均で $49a(K/L)H$ である。 K/L はクラスタ内の平均文字数を与えている。各小領域において、 K 個の字種を L 個のクラスタに分けているから、 $L \leq K$ である。 L が小さいほど、距離計算時間が小さく、連想整合法の方が速いことが予想される。しかし、 H が一定の場合は L が小さすぎると総合得点化の計算時間が増大する。したがって、処理時間を最小にする L の値が存在する。一方、手書き文字の場合は $H=L$ としたので、 T_a の第4項は $49aK$ となり、 T_a 全体としては、 L に関して、増加関数となる。すなわち、 L が小さいほど処理時間は短い。しかし、 L が小さい場合、各クラスタが大きくなりすぎ、部分情報を正確に反映しなくなる。したがって、最適な L は実験によって認識率、処理時間を調べ、決める必要がある。

4. 活字文字についての認識実験

活字文字を対象として連想整合法による認識実験を行い、比較のために全数整合法を用いた実験も行った。本章では、実験条件を示すと共に、実験結果に基づいて両方法を比較し、性能を確認する。

4.1 活字文字実験データ

活字認識実験で対象とした文字はJIS第一水準漢字2965文字である。実験用データとして、市販されているF, N, E合計3社のプリンタによる印刷文字を10本/mmのスキヤナで読みとったものを使用する。データセットは各社5セットずつ計15セットを用意した。F社製プリンタによる印刷文字を読みとったデータセットを $F_1 \sim F_5$, N社製によるものを $N_1 \sim$

表1 活字認識実験結果(全数整合法)

Table 1 The results of printed character recognition using template matching technique.

Data	1st	2nd	3rd
N_1	99.90	99.97	100.0
N_5	99.93	100.0	—
F_1	99.93	100.0	—
F_5	99.90	99.93	100.0
E_1	99.83	99.90	100.0
E_5	99.83	99.93	100.0
average	99.88	99.95	100.0

N_5, E 社製によるものを $E_1 \sim E_5$ とする。このうちの $F_1, F_2, F_3, N_1, N_2, N_3, E_1, E_2, E_3$ を標準パターン作成用データセットとし、残りのデータセット $F_4, F_5, N_4, N_5, E_4, E_5$ を未知入力用に使用する。また、活字認識実験用計算機として、FACOM-M 360を使用する。

4.2 全数整合法による活字文字認識実験

全数整合法の距離尺度として、ユークリッド距離を用いる。表1に実験結果を示す。表1が1位平均認識率が99.88%、3位累積認識率が100%に達していることを示している。認識時間は2秒/1文字であった。この結果は方向線素特徴量の活字認識における有効性の証拠を与えていると言える。

4.3 連想整合法による活字文字認識実験

連想整合用標準パターン辞書のクラスタ数 L の最適値を求めるために、クラスタ数8, 16, 32, 64の場合について行った。クラスタ数が2のべき乗であるのはクラスタリングにLBG法を用いたためである。実験

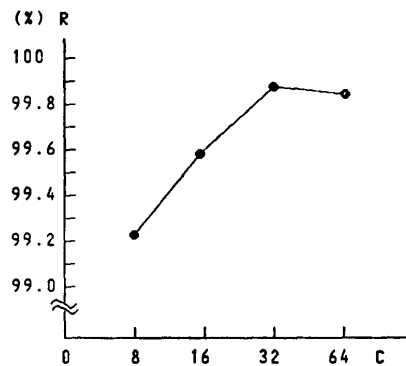


図4 クラスタ数と認識率の関係(活字)

C: クラスタ数, R: 1位認識率.

Fig. 4 The relation between cluster number and recognition rate.

C: cluster number, R: first recognition rate.

結果はクラスタ数と認識率および認識時間の関係としてまとめた。

図4にクラスタ数と平均1位認識率との関係を示す。この結果は四つのクラスタ数の中では、32のとき最も高い認識率が得られることを示している。64が32より低い原因は得点クラスタ数を上位5個に制限したからである。つまり、32のときと比べると、64の場合、クラスタ内の平均文字数が減り、その結果、点数がもらえる文字の割合も減り、わずかな変化に弱くなってしまっているからである。このことを逆にいえば、得点クラスタ数を抑えることにより、少ないクラスタ数で最高認識率が得られたことになり、高速化が可能であることを意味する。ここでは、 $H=5$ の妥当性を検証する実験結果を省略するが、結論だけを述べると、5より小さい値をとると、認識率が低下し、一方、5より大きい値をとっても、同程度の認識率しか得られず、得点操作を行うのに余計な時間がかかってしまうことになる。

各小領域におけるクラスタ数と認識時間の関係を図5に示す。図5から、クラスタ数が8, 16, 32のとき1文字認識するには大体0.4秒前後かかり、クラスタ数がそれ以上増えると時間が増加する傾向にあることがわかる(64のとき0.7秒ぐらい)。また、(8)式の最小値を与える L の値が16ぐらいにあることを示している。以上の実験結果から、クラスタ数が32のとき、認識率が最も高く、時間も小さいので、最適なクラスタ数として、32が適当であると考えられる。

クラスタ数が32のときの各未知入力データセットについての結果を表2に示す。さらに、表3には全数

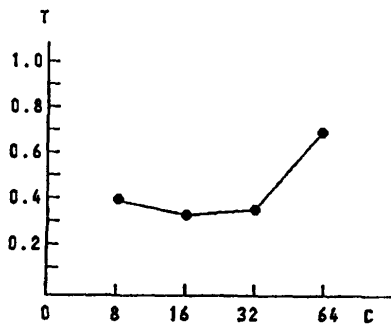


図5 クラスタ数と認識時間の関係 (活字)
C: クラスタ数, T: 認識時間 (単位: 秒/文字)

Fig. 5 The relation between cluster number and recognition time.
C: cluster number, T: recognition time (unit: seconds/character).

表2 活字認識実験結果 (連想整合法)

Table 2 The results of printed character recognition using associative matching technique.

Data	1st	2nd	3rd
N_1	99.87	99.93	100.0
N_2	99.90	99.97	100.0
F_1	99.90	99.93	100.0
F_2	99.93	100.0	—
E_1	99.80	99.87	100.0
E_2	99.87	99.97	100.0
average	99.87	99.92	100.0

表3 認識アルゴリズムの比較

T: 全数整合法, A: 連想整合法.

Table 3 Comparison of template matching technique and associative matching technique.

T: template matching technique, A: associative matching technique.

	1st	2nd	3rd	Recognition time
T	99.88	99.95	100.0	2.0 seconds/character
A	99.87	99.92	100.0	0.37 seconds/character

表4 誤認識の例

Table 4 Samples of missed characters.

Input	1st	2nd	3rd
休	休	休	—
又	又	又	—
騰	騰	騰	—
土	土	土	—
間	間	間	—
季	季	季	—
篤	篤	篤	—
遺	遺	遺	—
吉	杏	否	吉
犬	太	大	犬
玉	王	主	玉
鳥	馬	鳥	鳥

整合法との比較を示す。表3から認識率がほぼ同じで、認識速度が5倍ぐらい速くなったことがわかる。距離計算時間だけに注目すると、 $K/L=2965/32 \approx 92$ 倍になるはずであるが、全体としては5倍ということで、得点計算の時間や共通の処理時間(前処理とK個のソート)が大きいことを示している。

表4には誤認識の例を示してある。誤認識が類似文字間で起こっていることがわかる。類似文字の識別は構造上の情報を必要とし、細分類のテーマである。

5. 手書き文字についての認識実験

5.1 手書き文字の実験データ

手書き実験用データとして、電総研で作成された JIS 第一水準手書き漢字データベース ETL 9 (以降、ETL 9 と呼ぶ) を使用する。ETL 9 には合計 200 セット、総数 607,200 文字が含まれている。各セットは 3036 字種 (平仮名 71 字種、漢字 2965 字種) からなる。ここでは、この 200 セットの中から 100 セット (1~20, 41~60, 81~100, 121~140, 161~180) を選び、このうちの奇数番の 50 個のデータセットを標準パターン用に、偶数番の 50 個のデータセットを未知入力用に使用した。また、手書き文字認識実験用計算機として、SUN 4/110 を使用する。

5.2 全数マッチング法による手書き文字認識実験

活字認識実験と同じように、手書きにおいてもまず全数整合法を用いて実験を行う。距離尺度として、重み付きユークリッド距離を用いる。

実験結果を図 6 に示す。図 6 は 1 位認識率 (黒丸) の高い順にデータセットをならべてある。データの性質を示すために、ETL 9 の評価を行った文献 8) の 1 位認識率も示してある。方向線素特徴量による 1 位認識率は平均 86.87% であり、方向線素特徴量が手書き認識においても有効であることを示している。また、1 文字の認識時間は 6 秒であった。

5.3 連想整合法による手書き文字認識実験

5.3.1 予備実験

高速かつ正確に候補を上位に絞り込むため、各小領域における最適クラスタ数とクラスタ内文字に与える得点の与え方を決定することが必要である。そのため予備実験を行った。

1. 各小領域における最適クラスタ数

実験用データセットとして、偶数番最初の三つのセット (以下、A, B, C と呼ぶ) を用いた。連想整合度の計算では、 $H=L$ 、減点幅=1 とした。実験結果は、各セットにおけるクラスタ数と認識率の関係を図 7 (a), (b), (c) に、クラスタ数と認識所要時間の関係を図 8 に示す。図 7 から各小領域におけるクラスタ数を増やせば、1 位認識率が増加する傾向にある

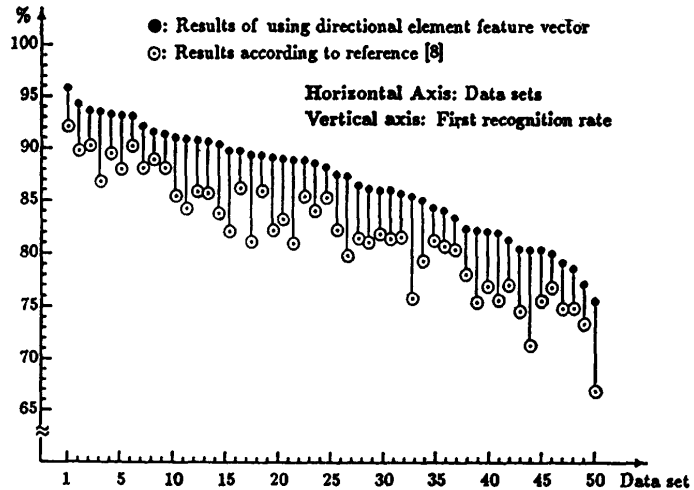


図 6 手書き文字の実験結果 (全数整合法)
Fig. 6 Results of hand-written characters using template matching technique.

が、100 位までの累積認識率はほとんど同じであることがわかる。大分類では認識率がそれほど悪くない限り、高速である方がよいので、各小領域における最適クラスタ数を 64 に決定する。

2. 連想整合度のパラメータの選定

式 (6) におけるパラメータ H および減点幅を定めるための実験を行った。

(a) 減点幅と認識率の関係

図 9 は減点幅を 0.1 から 1.5 まで変化させたときの認識率 (セット A, クラスタ数 64) の変化を示すものである。この結果は減点幅が認識率にそれほど影響を与えないことを示している。特に、100 位認識率の場合、ほとんど同じである。したがって、減点幅を計算が容易な 1 点とする。

(b) 得点クラスタ数と認識率の関係

図 10 は得点クラスタ数 (セット A, クラスタ数 64) が変化させたときの認識率の変化を示したものである。図 10 から全クラスタに点数を与えた方がいいことがわかる。つまり、この場合、最高点 64 から 1 点ずつ減らすということである。

以上の予備実験から、連想整合法を用いるときのパラメータを

クラスタ数: 64

減点幅: 1

得点クラスタ数: すべてのクラスタの数と決定した。

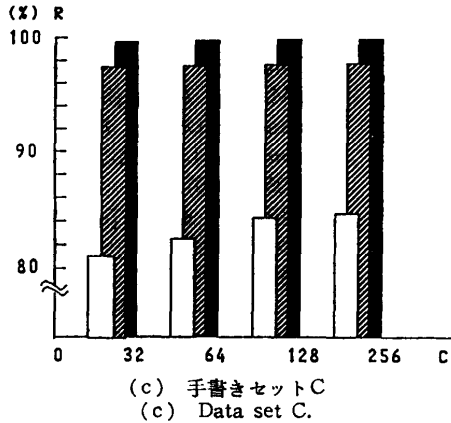
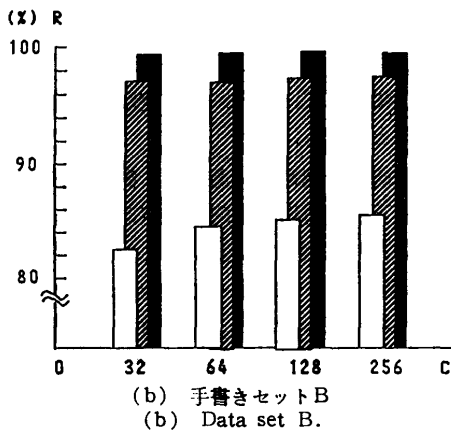
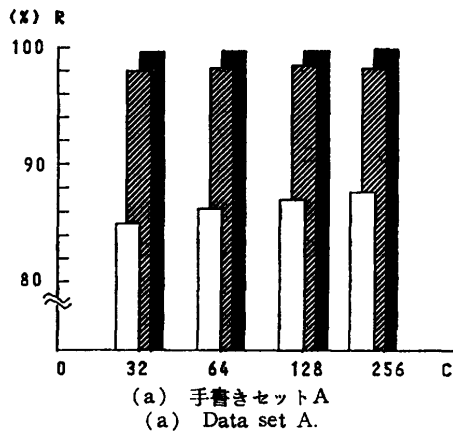


図 7 クラスタ数と認識率の関係
C: クラスタ数, R: 認識率.
□, ▨, ■ はそれぞれ 1 位, 10 位, 100 位認識率を表す。

Fig. 7 The relation between cluster number and recognition rate.
C: cluster number, R: recognition rate.
□: first recognition rate.
▨: recognition rate within the 10th candidates.
■: recognition rate within the 100th candidates.

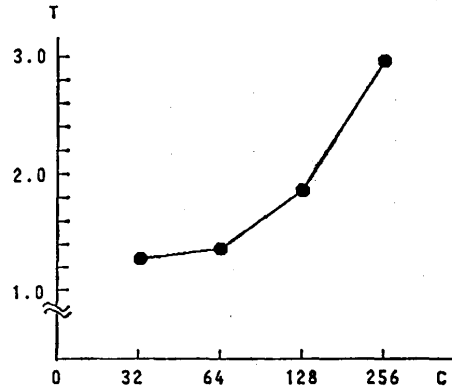


図 8 クラスタ数と認識時間の関係 (手書き)
C: クラスタ数, T: 認識時間 (秒/文字).
Fig. 8 The relation between cluster number and recognition time.
C: cluster number, T: recognition time (unit: seconds/character).

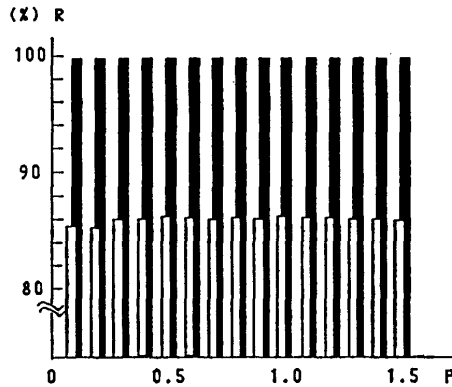


図 9 減点幅と認識率の関係
P: 減点幅, R: 認識率.
□, ■ はそれぞれ 1 位および 100 位認識率を表す。
Fig. 9 The relation between difference of points assigned to adjacent clusters and recognition rate.
P: difference of points assigned to adjacent clusters, R: recognition rate.
□: first recognition rate.
■: recognition rate within the 100th candidates.

5.3.2 連想整合法による大分類実験

予備実験から得られた最適パラメータを用いて、偶数番の 50 個のセットに対して、大分類を行った。図 11 はその結果である。データセットは 1 位認識率の高い順に並べた。平均認識率を計算した結果、1 位は 78.0%、3 位は 89.3%、100 位は 99.0% である。ま

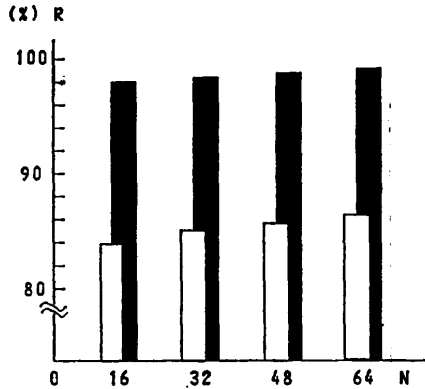


図 10 得点クラスタ数と認識率の関係
N: 得点クラスタ数, R: 認識率.

□, ■ はそれぞれ 1 位および 100 位認識率を表す.

Fig. 10 The relation between number of clusters assigned non-zero points and recognition rate.

N: number of clusters assigned non-zero points, R: recognition rate.

□: first recognition rate.

■: recognition rate within the 100th candidates.

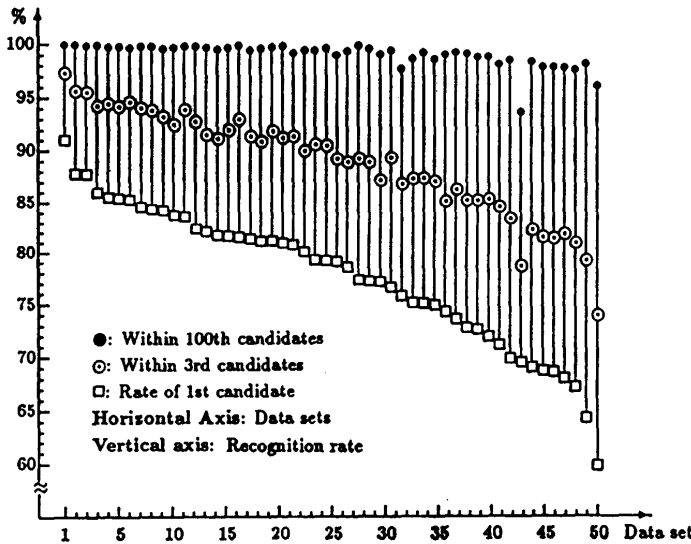


図 11 手書き文字の大分類結果 (連想整合法)

Fig. 11 Results of rough classification for hand-written characters using associative matching technique.

た, 1 文字認識する所要時間は 1.4 秒であった. この速度は全数整合法に比べて, 4.3 倍速くなっている. 具体的に計測すると, T_1, T_2 に共通な処理時間が $P=0.2, cK \log K=0.47$ であり, その合計は約 0.67 秒であった. よって, 全数整合における距離計

算時間に比べて, 連想整合法における距離計算と総合得点計算の合計時間が 7.3 倍に速くなっていることを示す.

6. おわりに

文字を高速, 高精度に認識する手法として, 連想整合法を提案した. 処理速度の向上について評価し, 実験によって確かめた. さらに, 認識率についても, 実験により評価した. 認識実験は活字 (2965 字種) および手書き文字 (3036 字種) に対して行い, 活字では 1 位認識率を 99.87% (3 位までが 100%) に, 手書き文字では 100 位累積認識率を 99% に確保できることが示された. 認識速度がいずれも全数整合法を用いた実験より 4 倍以上速くなっている. したがって, 本手法によって, 高速, 高精度な大分類の目的が達成されたといえる.

謝辞 本研究を進めるにあたり, 御討論頂いた堀口進助教授, 下平 博助手, 阿部 亨助手をはじめ, 木村研究室の方々に感謝します. また, データベース ETL 9 を提供して下さった電総研の皆様に感謝いたします. 本研究の一部は, 文部省科学研究費特別推進研究 (No. 63060001) の援助を受けている.

参考文献

- 1) Linde, Y., Buzo, A. and Gray, R.M.: An Algorithm for Vector Quantizer Design, *IEEE Trans. Comm.*, Vol. COM-28, No. 1, pp. 84-95 (1980).
- 2) 萩田紀博, 梅田三千雄, 増田 功: 三つの概形特徴を用いた手書き漢字の分類, *信学論 (D)*, Vol. J63-D, No. 12, pp. 1096-1102 (1980).
- 3) 山本和彦: 弛緩整合法による手書き教育漢字認識, *信学論 (D)*, Vol. J85-D, No. 9, pp. 1169-1174 (1982).
- 4) 岡 隆一: 図形からのセル特徴の抽出, *信学論 (D)*, Vol. J65-D, No. 10, pp. 1219-1226 (1982).
- 5) 萩田紀博, 内藤誠一郎, 増田 功: 大局的・局所方向寄与度特徴による手書き漢字認識方式, *信学論 (D)*, Vol. J66-D, No. 6, pp. 722-729 (1983).
- 6) 馬場口登, 相原恒博, 真田英彦, 手塚慶一: 構造的セグメント整合による手書き漢字部分パターンの抽出と同定について, *信学論 (D)*, Vol. J68-D, No. 3, pp. 337-344 (1985).
- 7) 江島俊朗, 中村洋介, 木村正行: 構造情報を含

- む手書き漢字認識のための特徴量, 信学論(D), Vol. J68-D, No. 4, pp. 789-796 (1985).
- 8) 斉藤泰一, 山田博三, 山本和彦: JIS 第1水準手書き漢字データベース ETL9 とその解析, 信学論(D), Vol. J68-D, No. 4, pp. 757-764 (1985).
- 9) 江島俊朗, 勝山 裕, 木村正行: 特徴ベクトルの分割と統合による手書き文字の大分類, 信学論(D), Vol. J70-D, No. 2, pp. 398-404 (1987).
- 10) 孫 寧, 市村 洋, 木村正行: 活字漢字の高速認識に関する研究, 昭63信学春季全大, D-456 (1988).
- 11) 大町真一郎, 阿曾弘具, 木村正行: 高速文字認識前処理用ストリックアレー, 信学論(D), Vol. J73-D-II, No. 2, pp. 167-174 (1989).
- 12) 内山喜照, 阿曾弘具, 木村正行: 非線形正規化と重み付き距離尺度を用いた文字認識, 平成元年東北連大1C2 (1989).

(平成2年9月11日受付)
(平成2年12月18日採録)



孫 寧

昭和37年生。昭和61年職業訓練大学校電子科卒業。昭和63年東北大学大学院工学研究科情報工学専攻修士課程修了。現在、同大学院博士課程在学中。文字認識に関する研究に従事。電子情報通信学会、人工知能学会各会員。



阿曾 弘具 (正会員)

昭和21年生。昭和43年東北大学工学部電気卒業。昭和48年同大学院博士課程修了。同年同大工学部助手。昭和54年名古屋大学工学部講師。昭和57年同助教授を経て、現在、東北大学工学部情報工学科助教授。工学博士。その間、学習オートマトン、セル構造オートマトン、並列処理理論などの研究に従事。昭和53年電子情報通信学会学術奨励賞受賞。著書「UNIX と Pascal」(近代科学社, 共著)。電子情報通信学会, IEEE, ACM 各会員。



木村 正行 (正会員)

昭和2年生。昭和29年東北大学工学部電気卒業。昭和34年同大学院博士課程修了。同年同大電気通信研究所助手。昭和37年同研究所助教授。昭和45年同大工学部教授。現在情報工学科に所属。工学博士。システム理論とその応用, しきい値理論, 学習オートマトン, 視覚神経系のモデル(神経回路網)などの研究に従事してきた。最近, 文字, 図形および顔画像の認識, 音声認識など, 知的情報処理の分野に興味を持っている。