

弾性マッチングと2次識別に基づくオンライン文字認識

三苦寛人[†] 内田誠一^{††} 迫江博昭^{††}

[†] 九州大学大学院システム情報科学府

^{††} 九州大学大学院システム情報科学研究所

〒 812-8581 福岡市東区箱崎 6-10-1

E-mail: {mitoma, uchida, sakoe}@human.is.kyushu-u.ac.jp

あらまし 弾性マッチング(非線形伸縮マッチング)に基づくオンライン文字認識においては, その変形への柔軟性の副作用として合わせ過ぎによる誤認識が発生する. その合わせ過ぎによる誤認識を防ぐため, 本報告では弾性マッチングを統計的識別の枠組みに組み込むことを提案する. 提案手法では, 弾性マッチングは, 入力パターンと標準パターン間の差分を表す差分ベクトルを抽出するためにのみに用いられる. 抽出後, その差分ベクトルは正規分布に従うという仮定で, 2次識別が行われる. UNIPEN データベース (Train-R01/V07,1a) を用いた認識実験により最高認識率 97.95% が得られ, 本手法の有効性が確かめられた.

キーワード オンライン文字認識, 固有変形, DP マッチング, 2次識別関数, UNIPEN データベース

Online Character Recognition Based on Elastic Matching and Quadratic Discrimination

Hiroto MITOMA[†], Seiichi UCHIDA^{††}, and Hiroaki SAKOE^{††}

[†] Graduate School of Information Science and Electrical Engineering, Kyushu University

^{††} Faculty of Information Science and Electrical Engineering, Kyushu University

Hakozaki 6-10-1, Higashi-ku, Fukuoka-shi, 812-8581 Japan

E-mail: {mitoma, uchida, sakoe}@human.is.kyushu-u.ac.jp

Abstract In this study, elastic matching is linked with a statistical discrimination framework to overcome the overfitting problem which often degrades the performance of elastic matching-based online character recognizers. In the proposed technique, elastic matching is used just as an extractor of a feature vector representing the difference between input and reference patterns. Then quadratic discrimination (i.e., Bayesian discrimination for feature vectors governed by Gaussian distributions) is performed under the assumption that the feature vector is governed by a Gaussian distribution. The result of a recognition experiment on UNIPEN database (Train-R01/V07, 1a) showed that the proposed technique can attain a high recognition rate (97.95%) and outperforms a recent elastic matching-based recognizer.

Key words online character recognition, eigen-deformations, DP matching, quadratic discriminant function, UNIPEN database

1. ま え が き

オンライン文字認識において, 弾性マッチングは入力パターンと標準パターン間の最適筆点对応を求めるためにしばしば用いられる手法であり, パターン長(すなわち筆点数)の違いを揃えたり, 特徴量(例えば, xy 座標特徴や方向特徴)の違いを最小化するなどの機能を持っている. DP マッチング [1], [2] は, 古典的な弾性マッチングの手法でありながら, 現在でも頻繁に

用いられる手法である. DP マッチングにはいくつかの長所があるが, $O(IJ)$ (I, J はそれぞれ標準パターン長, 入力パターン長) という少ない計算量で大域的に最適なマッチングを与えることができるのもその 1 つである. また, マッチングの最適化の結果得られるマッチングコストはそのまま識別関数として用いることができる.

DP マッチングやその他の弾性マッチングに基づく認識手法の有効性については数多くの報告がなされているが, 一方で,

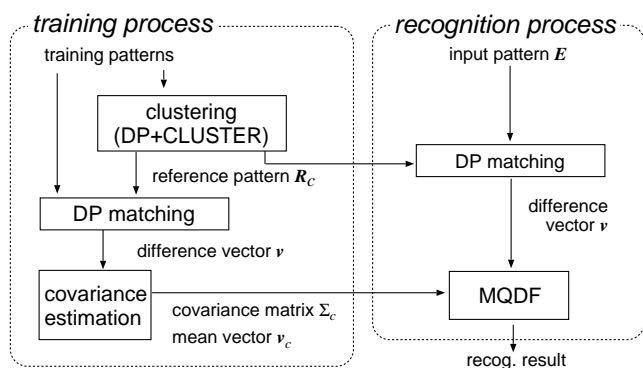


図1 本手法による学習と認識の流れ。

その変形への柔軟性の副作用として、合わせ過ぎによる誤認識がある。これは異なるカテゴリのパターン間であっても、柔軟な対応付けの結果、それらの間のマッチングコストが過小評価されてしまう現象である。例えば、“1”と“7”では、弾性マッチングによりそれらの書き始めの部分の長さの違いまで補償され、誤認識されることがある。

この合わせ過ぎの問題の1つの対処法として、確率/統計を組み込む手法がある。統計的DP [3] や隠れマルコフモデル(HMM) [4] ~ [6] は、DPの確率的拡張であり、特徴量に対し確率的な制約を加えることによって合わせ過ぎを排除することができる。これらの手法は、単純なDPマッチングより性能が良いことが報告されている。しかし、DPに基づくマルコフ性のため筆点それぞれに対し“局所的”にのみしか制約を加えることができず、すべての合わせ過ぎを排除することはできない。したがって、これらの手法では、筆点全体の“大局的”な規制をすることができない。

本論文では、合わせ過ぎの問題に対処するために、弾性マッチングを統計的識別の枠組みにおいて大局的な制約を加えることを考える。具体的には、本手法では、弾性マッチングを特徴抽出法としてのみ扱い、そのマッチングコストは用いない。すなわち、弾性マッチングは対応する筆点毎の特徴量の差分からなる差分ベクトルを得るためにのみ用いられる。この差分ベクトルは、標準パターンを基準とした入力パターン全体の“大局的”な特徴を表している。差分ベクトルの抽出後、差分ベクトルの分布が事前に推定された平均と共分散を持つ正規分布に従うという仮定の下でベイズ識別を行う。合わせ過ぎが起きた際の差分ベクトルはその分布から逸脱していると考えられる。したがって、その差分ベクトルの事後確率は小さな値となり、合わせ過ぎによる誤認識は本手法により排除されると考えられる。図1に本手法による学習と認識の流れを示す。ここで、MQDF(修正2次識別関数) [7] は正規分布に従うパターンのベイズ識別関数である2次識別関数の改良版である。

文献 [8], [9] では、本論文と類似の検討がなされている。具体的な相異点は以下の通りである。

1. 同文献では、DPマッチングコストと事後評価量の加重和を識別のための距離尺度としている。これに対し本論文では、固有変形に基づく事後評価量のみを用いて厳密な2次識別関数

を構成し、それに基づいた認識を行う。

2. 同文献で、固有変形は、座標特徴のカテゴリ内変動を表現するものである。本論文では、これを拡張し、オンライン文字パターンの表現のために頻りに利用される他の特徴、具体的には方向特徴についても、そのカテゴリ内変動を併せて表現するようにした。

上記第1項については、従来必要となっていた重みの設定が不要になる点が利点として挙げられる。こうした重みの設定は、事前に学習データを用いた最適化を行う必要があり、相当の手間を要する。また、文献 [8], [9] で示されているように、学習パターンで設定された値が、テスト時にも適切であるとは限らない。さらに、DPマッチングコストと事後評価量は、類似した部分もあるものの、本来性質が異なるものであり、それらを単純に加重和することの理論的に十分妥当性であるとは言えない。これに対し、本論文では、DPマッチングコストは用いずに、事後評価量のみを用いる。具体的には、DPマッチングによる対応付けをベイズ識別理論から導出される厳密な2次識別関数を用いて評価するという、よりシンプルで自然な方式を採用する。このように本手法では、DPマッチングコストの識別能力を放棄したことになるが、本稿ではそれが大きな問題にならないことを実験により明らかにする。

上記第2項は、第1項と関係し、事後評価の識別能力を向上させるという効果を狙ったものである。方向特徴は、オンライン文字形状を安定して表現できる特徴量として、座標特徴以上に頻りに用いられている。マッチング時の対応付けを、従来の座標特徴だけでなく方向特徴も併せて事後評価すれば、座標特徴では評価しにくい対応付けも適切に評価できるものと期待される。

本論文では、2節でDPマッチングとその問題点、3節ではその解決法である2次識別関数、4節では2次識別の学習の過程で得られる固有変形、5節では2次識別関数による認識実験および考察について述べる。

2. DPマッチングと合わせ過ぎ

2.1 DPマッチング

カテゴリ $c \in \{1, 2, \dots, C\}$ の標準パターン R_c を時系列

$$R_c = r_1, r_2, \dots, r_i, \dots, r_I \quad (1)$$

と表し、未知の入力パターン E を時系列

$$E = e_1, e_2, \dots, e_j, \dots, e_J \quad (2)$$

と表す。ここで r_i および e_j は筆点の特徴ベクトルであり、

$$r_i = (X_i, Y_i, \Theta_i)^T \quad (3)$$

$$e_j = (x_j, y_j, \theta_j)^T \quad (4)$$

のように x 座標、 y 座標、および方向を要素とする3次元ベクトルとする。なお、正確には r_i, I をそれぞれ $r_{c,i}, I_c$ のように記述すべきだが、誤解のない限りこれらの簡略表現を用いる。

2パターン R_c と E 間のマッチングは、次の最適化問題により求まる。

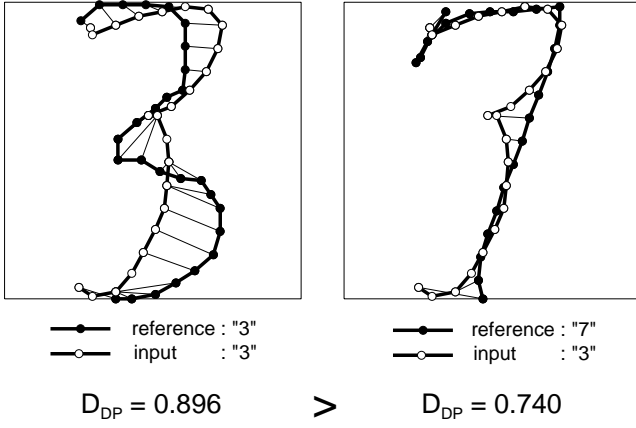


図2 合わせ過ぎによる誤認識の例.

[目的関数]

$$\frac{1}{I} \sum_{i=1}^I \|r_i - e_{j(i)}\| \rightarrow \text{minimize} \quad (5)$$

[制御変数 (整数変数)]

$$j(1), \dots, j(i), \dots, j(I)$$

[制約条件]

$$\begin{cases} j(i) - j(i-1) \in \{0, 1, 2\} \\ j(1) = 1 \\ j(I) = J \end{cases}$$

ここで記号 $\|\cdot\|$ は、ユークリッドノルムである．よく知られているように、この問題は DP アルゴリズム (ここでは省略する) を用いることで効率的に解ける．

2.2 合わせ過ぎ

多くのオンライン文字認識手法では、マッチングコスト、すなわち目的関数 (5) の最小値

$$D_{DP}(R_c, E) = \min_{j(1), \dots, j(i), \dots, j(I)} \frac{1}{I} \sum_{i=1}^I \|r_i - e_{j(i)}\| \quad (6)$$

がパターン間の距離としてそのまま用いられている．これは、 D_{DP} が E に生じる多く非線型変形に不変な距離であるためである．

しかしながら、 D_{DP} による認識は、前述のように合わせ過ぎによる異なるカテゴリのパターン間距離の過小評価が問題となる．図2に合わせ過ぎによる誤認識の例を示す．これは、不自然な対応付けによって D_{DP} が過小評価されたため、“3”の入力パターンが“7”に誤認識された例である．このような誤認識はカテゴリ固有の変形の傾向を考慮してないために引き起こされていると考えられる．実際、同図のような“7”と“3”のマッチングは、“7”のカテゴリではほとんど起きないと考えられる．

3. 二次識別によるオンライン文字認識

3.1 特徴抽出 — 差分ベクトル

本手法では、上述の合わせ過ぎを避けるため、入力パターン

の大局的な特徴を表す新たな特徴ベクトルの分布を用いた統計的識別を考える．この特徴ベクトルは、以下に示すように DP マッチングの結果から得られるものであり、以降、これを差分ベクトルと呼ぶ．このように本手法では、DP マッチングは単に特徴抽出のために用いられ、それによるマッチングコスト D_{DP} は用いられない．

入力パターン E の $3I$ 次元差分ベクトルは、 R_c と E 間の最適筆点对応関係 $j(1), \dots, j(I)$ を用いて求めることができる．すなわち、標準パターン R_c を基準とした場合の入力パターン E の $3I$ 次元差分ベクトル v は次のように表現される．

$$v = \left((X_1 - x_{j(1)}, Y_1 - y_{j(1)}, \Theta_1 - \theta_{j(1)}), \dots, (X_i - x_{j(i)}, Y_i - y_{j(i)}, \Theta_i - \theta_{j(i)}), \dots, (X_I - x_{j(I)}, Y_I - y_{j(I)}, \Theta_I - \theta_{j(I)}) \right)^T \quad (7)$$

このとき、標準パターン R_c を基準として入力パターンの差分を表現しているため、入力パターンの長さに関係なく同じ $3I$ 次元となる．よって、以下の差分ベクトルの共分散を推定することができる．なお、以下の議論では、 $3I$ 次元差分ベクトル v の部分ベクトル $v^{x,y}$ と v^d を用いる．これらはそれぞれ v を座標成分と方向成分に分けて得られたものであり、次のように定義される．

$$v^{x,y} = \left((X_1 - x_{j(1)}, Y_1 - y_{j(1)}), \dots, (X_i - x_{j(i)}, Y_i - y_{j(i)}), \dots, (X_I - x_{j(I)}, Y_I - y_{j(I)}) \right)^T \quad (8)$$

$$v^d = (\Theta_1 - \theta_{j(1)}, \dots, \Theta_i - \theta_{j(i)}, \dots, \Theta_I - \theta_{j(I)})^T \quad (9)$$

3.2 共分散の推定

上述のように各カテゴリ毎に差分ベクトルの次元数が揃っているため、差分ベクトルの共分散行列は次の式で求めることができる．

$$\Sigma_c = \frac{1}{N_c} \sum_{n=1}^{N_c} (v_{c,n} - \bar{v}_c)(v_{c,n} - \bar{v}_c)^T \quad (10)$$

ここで、 N_c は学習パターン数、 $v_{c,n}$ は学習パターンから得られた差分ベクトル、 \bar{v}_c は $v_{c,n}$ の平均ベクトル、すなわち

$$\bar{v}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} v_{c,n} \quad (11)$$

を表す．ただし、実際には後述するように、次元削減のために差分ベクトルを位置情報と方向情報に分けて別々に推定する．

3.3 2次識別関数

差分ベクトルの分布をガウス分布と仮定すると、そのベイズ識別問題はよく知られているように2次識別問題に帰着する．その際の2次識別関数は以下のように定義される．

$$D_{tot}(R_c, E) = (v - \bar{v}_c)^T \Sigma_c^{-1} (v - \bar{v}_c) + \log |\Sigma_c| + 3I \log 2\pi \quad (12)$$

なお、式 (12) の右辺の最終項は通常よく省略されるが、オン

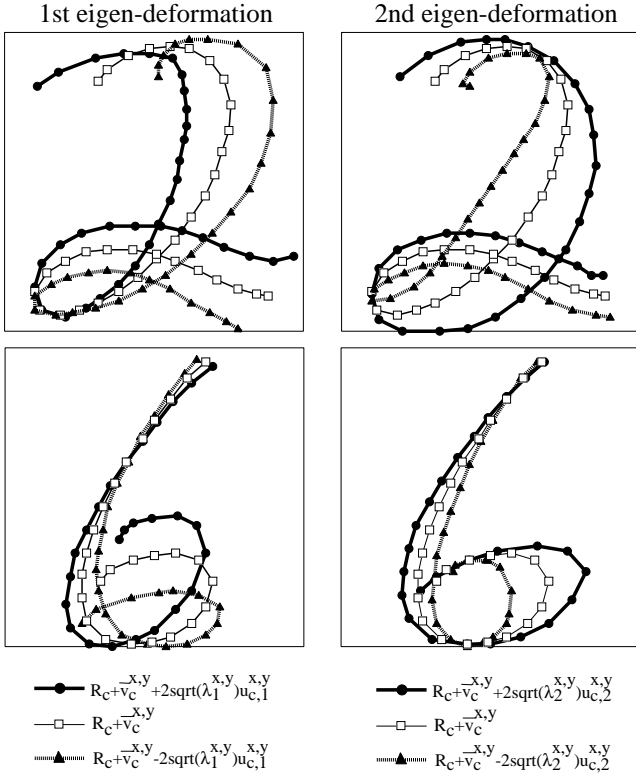


図3 上位2つの固有変形 $u_{c,1}$, $u_{c,2}$ により変形された“2”と“6”の標準パターン。

ライン文字認識においては特徴ベクトル v の次元数が各カテゴリ毎に異なるので省略できない。

以下では、 Σ_c の次元削減のため、 xy 座標特徴と方向特徴は独立であると仮定する。この仮定により、式(12)は座標特徴の識別関数 D_{pos} と、方向情報の識別関数 D_{dir} に分けられる。すなわち

$$D_{\text{tot}}(R_c, E) = D_{\text{pos}}(R_c, E) + D_{\text{dir}}(R_c, E), \quad (13)$$

となる。ここで、

$$\begin{aligned} D_{\text{pos}}(R_c, E) &= (v^{x,y} - \bar{v}_c^{x,y})^T (\Sigma_c^{x,y})^{-1} (v^{x,y} - \bar{v}_c^{x,y}) \\ &+ \log |\Sigma_c^{x,y}| + 2I \log 2\pi \\ &= \sum_{m=1}^{2I} \frac{1}{\lambda_{c,m}^{x,y}} ((v^{x,y} - \bar{v}_c^{x,y})^T u_{c,m}^{x,y})^2 \\ &+ \log \prod_{m=1}^{2I} \lambda_{c,m}^{x,y} + 2I \log 2\pi. \end{aligned} \quad (14)$$

である。行列 $\Sigma_c^{x,y}$ は $2I \times 2I$ の共分散行列であり、カテゴリ c の学習サンプルと標準パターン R_c 間の部分ベクトル $v^{x,y}$ によって推定される。 $u_{c,m}^{x,y}$ と $\lambda_{c,m}^{x,y}$ は、それぞれ $\Sigma_c^{x,y}$ を固有値展開して得られる固有ベクトルと固有値を表す。次数 m は固有変形の寄与(固有値 $\{\lambda_{c,m}\}$) の大きさ順に対応する。

一般に高次の固有値は非常に小さな値となり、数値誤差も含みやすい。よって式(14)のようにその逆数を使用した場合、数値誤差が拡大し、その影響が無視できなくなる。このため、本論文では、高次の固有値 $\lambda_{c,m}^{x,y} (m = M^{x,y} + 1, \dots, 2I)$ を $\lambda_{c, M^{x,y} + 1}^{x,y}$

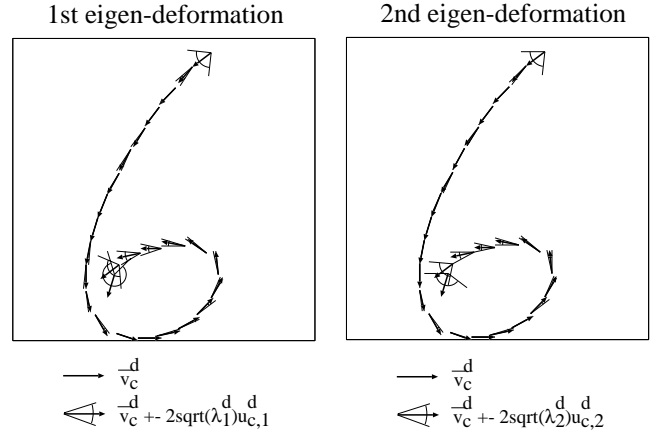


図4 上位2つの方向情報の固有変形 $u_{c,1}^d, u_{c,2}^d$ によって得られた“6”の方向の変動。

に置き換えた修正2次識別関数[7]

$$\begin{aligned} D_{\text{pos}}(R_c, E) &\sim \frac{1}{\lambda_{c, M^{x,y} + 1}^{x,y}} \|v^{x,y} - \bar{v}_c^{x,y}\|^2 \\ &+ \sum_{m=1}^{M^{x,y}} \left(\frac{1}{\lambda_{c,m}^{x,y}} - \frac{1}{\lambda_{c, M^{x,y} + 1}^{x,y}} \right) ((v^{x,y} - \bar{v}_c^{x,y})^T u_{c,m}^{x,y})^2 \\ &+ \log \left\{ (\lambda_{c, M^{x,y} + 1}^{x,y})^{2I - M^{x,y}} \prod_{m=1}^{M^{x,y}} \lambda_{c,m}^{x,y} \right\} \\ &+ 2I \log 2\pi. \end{aligned} \quad (15)$$

を用いる。パラメータ $M^{x,y}$ の決め方には様々な方法が考えられるが、本論文では $\sum_{m=1}^{M^{x,y}} \lambda_{c,m}^{x,y} / \sum_{m=1}^{2I} \lambda_{c,m}^{x,y} > \mu^{x,y}$ を満たす最小の $M^{x,y}$ を用いることとした。後述する認識実験においては、学習データを対象とした実験により最適化した $\mu^{x,y}$ を用いた。

以上では、 D_{pos} について述べたが、 D_{dir} についても $v^{x,y}$ を v^d , $\Sigma_c^{x,y}$ を Σ_c^d というように置き換えることによって同様に記述できるので、ここでは省略する。なお、この方向成分の差分ベクトル v^d に関して、その各要素の存在区間 $[-\pi : \pi]$ の両端部は同一視されるべき(すなわち方向差分 $-\pi$ と π は等しい)である。従って、厳密には座標成分と同様の処理は適用できない。しかしながら、差分ベクトルの分布の各要素の分布を観察したところ、ほとんどが0付近を中心とした正規分布状であり、 $\pm\pi$ 付近の値(すなわち標準パターンと逆方向になる文字線)は少なかったため、今回は実用上問題ないものとして、 v^d についても $v^{x,y}$ と同様の処理を適用した。

4. 固有変形

前述の固有ベクトル $u_{c,m}^{x,y}$ は特徴空間における変位ベクトル $v^{x,y}$ の分布の主軸である。従って、低次の $u_{c,n}^{x,y}$ は、カテゴリ c の頻出する変形を表す。以降、 $u_{c,m}^{x,y}$ はオンライン文字の(位置情報の)固有変形と呼ぶ。

図3に、数字“2”と“6”の標準パターンについての実際に求めた上位2つの固有変形 ($u_{c,1}^{x,y}, u_{c,2}^{x,y}$) を示す。この図より、数字の“2”の第1固有変形は全体の傾きを、第2固有変形は全

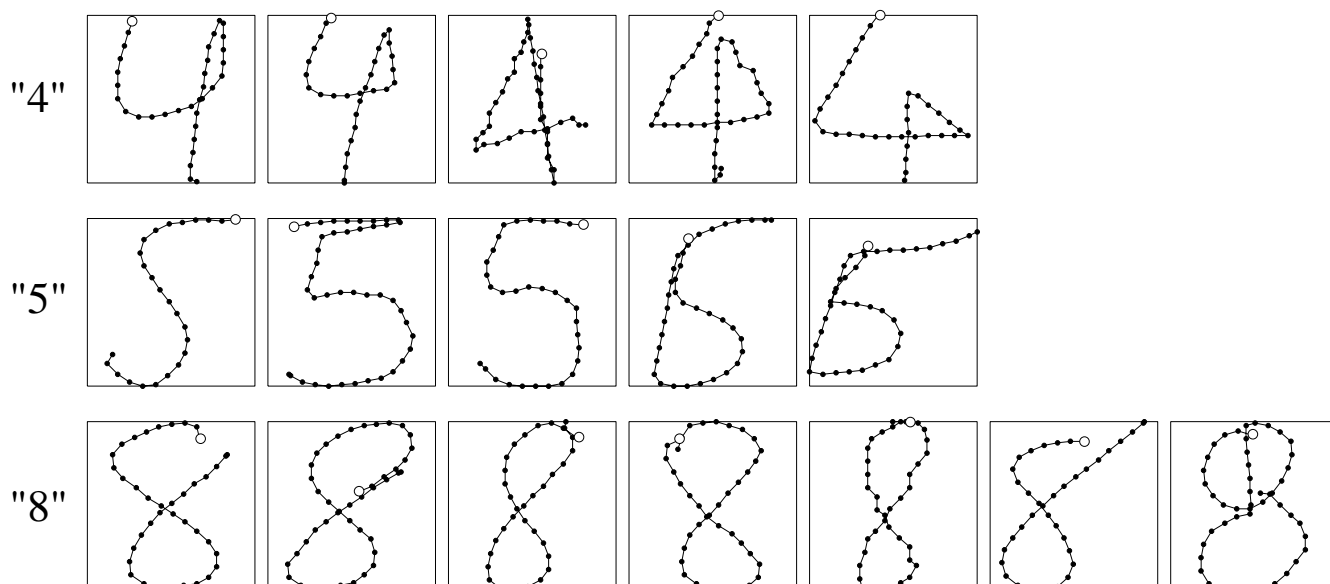


図 5 CLUSTER により求めた標準パターンの例．白点は開始点．

体的な太さの変動を表していることがわかる．同様に，数字の“6”の第1固有変形はループ部の上下の変動を，第2固有変形はループ部の左右の変動を表していることがわかる．このように求めた固有変形は，実際によく見られる変形を表している．

方向情報についても Σ^d の固有ベクトルとして同様に固有変形を考えることができる．図4に数字の“6”について上位2つの方向情報の固有変形を示す．この図より上位の固有変形には，ループ部の上部で方向の変動が見られることより，“6”のループ部の上部の上下の変動が求まっていることがわかる．

カテゴリ c の任意のオンライン文字パターン (の差分ベクトル) は，カテゴリ c の固有変形の張る部分空間に分布していると考えられる．これにより，オンライン文字認識の別の新たな枠組が導かれると考えられる．

5. 認識実験

本節では，3.章で述べた2次識別関数により，実際に得られた認識実験結果を示す．

5.1 データベース

本論文では，実験試料として UNIPEN Train-R01/V07 データベース [10], [11] の孤立数字 (1a) を用いた．本データベースは，世界的に広く用いられているものであり [12]，総数 15,953 サンプルから構成されている．本実験ではラベルミス等のクリーニングは行わず，これら 15,953 サンプルすべてを用いた．これは，他の手法 (特に [3]) との比較を容易にするためである．

以下の実験では，UNIPEN データベースの学習/テストデータ分割ソフトウェアとして推奨されている `utils2compareHWR` [12], [13] を用いて，データを学習用とテスト用に分割した．具体的には，学習用データとテスト用データで筆記者が独立でない分割方式 (multi writer 方式 [12]) を採用し，分割数を3つにして，このうちの2つ (すなわち，全データの2/3) を学習用のデータとして，残りの1つ (全データの1/3) をテスト用のデータとして用いた．以下の結果は，この3分割法に従った交差確

認法による3つの結果の平均値を示す．また，学習用のサンプルは5.2の標準パターン生成および共分散の推定に，テスト用データは認識タスクとして用いた．

これら全てのサンプルについて，画数変動の影響を無くす為に，ペンアップ部分も接続して1文字1画とした．その後，文字全体の大きさを縦横比は保持したまま 128×128 に線形正規化し，さらに隣接筆点間の距離が一定になるようにリサンプリングを行った．

5.2 クラスタリングによる標準パターン設定

標準パターンの生成は，文献 [14] にならい，Jain らによる CLUSTER [15], [16] を用いて自動的に行った．これは，k-means に類したクラスタリング手法であり，クラスタの分割・統合を繰り返しながら，クラスタ数が1個, 2個, ..., K 個の場合それぞれについての準最適なセントロイドを一括して与えることができる方法である．したがって，CLUSTER を1度実行すれば各カテゴリ最大 K 個までの標準パターンが準備できることになる．なお，アルゴリズムについては，別途で付録1.に載せる．

また，固有変形の推定に関しては，求めた各セントロイドに属するメンバーをそのセントロイド (すなわち標準パターン) に関する固有変形の学習パターンとして用いた．

求めた1個, 2個, ..., K 個の場合の標準パターンのうち，何個の標準パターンを用いるかについては，全カテゴリ共通のパラメータ T を用いて決定した．この T は，各クラスタの最小メンバー数であり，実際には固有変形の学習に用いる学習パターン数に相当する．すべてのメンバー数がその閾値 T 以上となる最大の標準パターン数 C を用いる．したがって， T が小さいと K は大きくなり学習に用いるサンプル数が減少し，逆に T が大きいと K は小さくなるが学習に用いるサンプル数が増加する．

実際に CLUSTER により求めた標準パターンの例を図5に示す．図より，様々な形状の標準パターンが求まっていることがわかる．

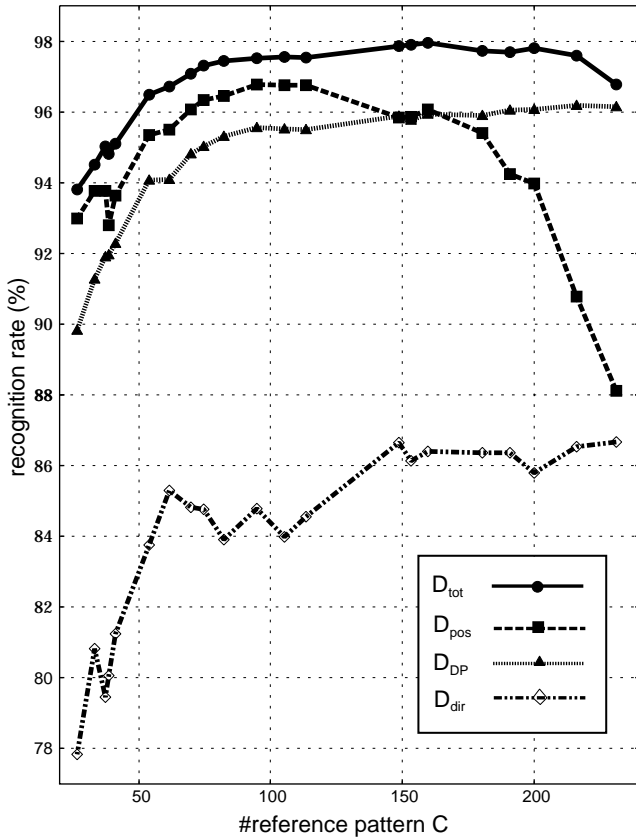


図 6 テストデータの認識率.

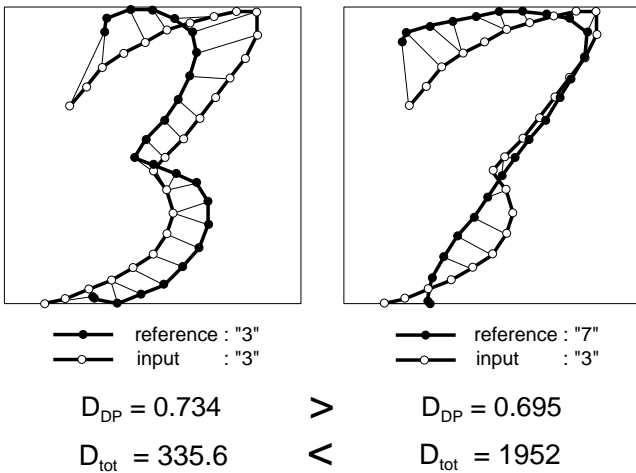


図 7 本手法により改善された例 .

5.3 認識結果と考察

図 6 に、標準パターン数 C (実際には T) を変えながら測定した、本手法すなわち D_{tot} に基づく最短距離法を用いた場合の認識率を示す。また、比較のため、位置情報のみの 2 次識別関数による認識率、および距離 D_{DP} による認識率も図に示す。

同図より D_{DP} より D_{tot} による認識率が高いことがわかる。これは、 D_{DP} を用いた際に起きていた合わせ過ぎによる誤認識が、本手法 D_{tot} により低減したためだと考えられる。図 7 に、実際に改善された例を示す。図より、 D_{DP} では、入力 の “3” は全体的に形の近い “7” に誤認識されていることがわかる。一方、大局的な変形評価をしている D_{tot} では、“7” では起こり得な

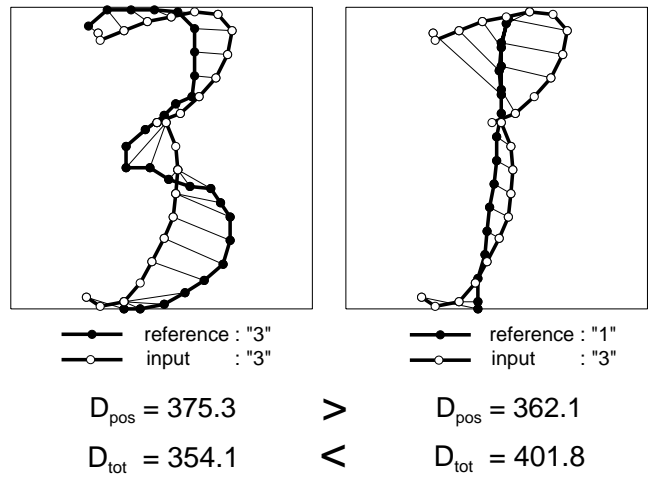


図 8 方向情報を用いることにより改善された例 .

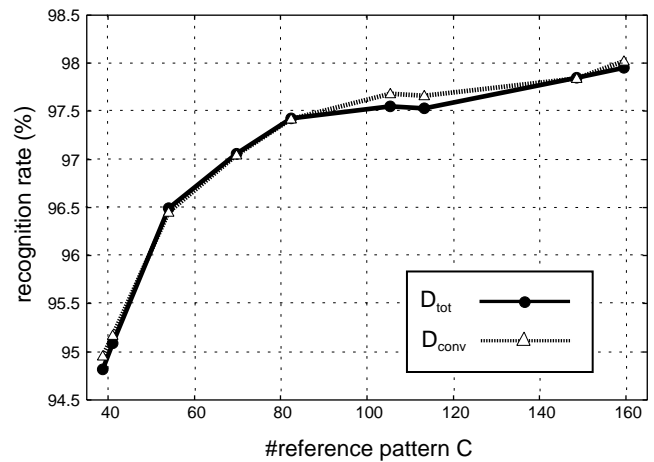


図 9 加重和型識別法 [8], [9] との認識率の比較.

い変形とみなされ、大きな距離となり、正しく認識されている。

また図 6 において D_{tot} による認識率を見ると、 $C = 160$ のときに最高認識率 97.95% を得ていることがわかる。また、DP を用いた最近の認識手法 Statistical dynamic time warping (SDTW) [3] が最高認識率 97.1% を与えた $C = 150$ の場合について見ると、本手法による認識率はそれより高い 97.85% であった。これは、SDTW が大局的な変形評価をしないのに対し、本手法が大局的な変形評価をしているため、合わせ過ぎを防げたことによるものと考えられる。

次に、 D_{tot} と D_{pos} を比べると、前者の方が常に約 1% 高い認識率を得ていることがわかる。その改善例として図 8 にそのサンプルを示す。同図よりわかるように、数字の “3” の中心の屈曲部が小さいため “1” との位置差が小さく、その結果 D_{pos} (“1”, “3”) は大きくなり、誤認識されている。しかし、屈曲部において方向情報は大きく変動しており、これは数字の “1” には見られない変動である。これにより、 D_{dir} (“1”, “3”) は大きくなり、数字の “1” との距離 D_{tot} (“1”, “3”) が大きくなったと考えられる。このように、方向情報も用いることによって、より高精度な認識が行えることがわかる。

また、図 9 に [8], [9] に倣って D_{tot} と D_{DP} を加重和した

$$D_{\text{conv}}(\mathbf{R}_c, \mathbf{E}) = (1 - \alpha)D_{\text{DP}} + \alpha D_{\text{tot}} \quad (16)$$

による認識率を示す．重み α については学習データを用いて最適化した．これら D_{tot} と D_{conv} による認識率を比べると，全体的にほぼ同程度の認識率となることがわかる．このように本手法では，加重係数 α を最適化する手間なしにほぼ同等の認識精度を得られることがわかる．また，文献 [8], [9] において報告されているように， α の値は学習データの筆記者により変動し，他の筆記者によるテストデータに対し同じ α を用いると認識率が低下することがある．この点を確かめるためには，筆記者独立という設定で，今後，検証する必要があると考えている．

計算時間については，Intel(R) Xeon(TM) CPU 3.06GHz を搭載した PC で，1 文字あたり 23.6 msec. であった．よって，計算時間の観点から見ても，単文字認識において本手法は実用的であると考えられる．

6. ま と め

弾性マッチングを統計的識別の枠組みに組み込んだ新たなオンライン文字認識を提案した．本手法では，入力パターンと標準パターン間の差分を表す特徴ベクトルを抽出するためにのみ弾性マッチングを用いた．抽出後，その特徴ベクトルが正規分布に従うという仮定の下で，2 次識別を行った．UNIPEN データベースを用いた数字実験により最高認識率 97.95% の認識率を得ることができ，本手法の高い認識性能を確認することができた．これは，最近の弾性マッチングに基づく認識手法 [3] よりも同条件で高い認識率であった．また，変形の表現に位置情報だけでなく方向情報も用いることの有効性についても実験により確認することができた．

本論文では，固有変形を明には用いなかったが，固有変形はオンラインの文字の変形の有望な表現方法だと考えられる．したがって，今後の課題として固有変形を利用した他の新たな認識手法を検討しようと考えている．他にも，DP とこの固有変形をマッチングすることで特徴抽出を介さず，直接，識別関数値 D_{DP} を求める方法についても検討する．

謝 辞

本研究の一部はセコム科学技術振興財団研究助成金，および総務省戦略的情報通信研究開発推進精度の支援を受けた．

文 献

- [1] 藤原塩和, 池田克夫, 富永善治, 清野 武, “接線ベクトル列を用いたオンライン手書き文字の認識,” 情報処理, vol. 17, no. 3, pp. 191-199, 1976.
- [2] K. Yoshida and H. Sakoe “Online handwritten character recognition for a personal computer system”, IEEE Trans. Consumer Electronics, vol. CE-28, no. 3, pp. 202-209, 1982.
- [3] C. Bahlmann and H. Burkhardt, “The writer independent online handwriting recognition system *frog on hand* and cluster generative statistical dynamic time warping,” IEEE Trans. PAMI, vol. 26, no. 3, pp. 299-310, 2004.
- [4] J. Hu, M.-K. Brown, and W. Turin, “HMM based online handwriting recognition,” IEEE Trans. PAMI, vol. 18, no. 10, pp. 1039-1045, 1996.
- [5] 伊藤 等, 中川正樹, “Hidden Markov Model に基づくオンライン手書き文字認識,” 信学技報, PRMU97-85, 1997.

- [6] 嵯峨山茂樹, 中井 満, 下平 博, “ストローク HMM に基づくオンライン手書き文字認識方式,” 信学技報, PRMU2000-35, 2000.
- [7] F. Kimura, K. Takashina, S. Tsuruoka, “Modified quadratic discriminant functions and the application to Chinese character recognition,” IEEE Trans. PAMI, vol. 9, no. 1, pp. 149-153, 1987.
- [8] 三苦寛人, 内田誠一, 迫江博昭, “弾性マッチングと固有変形を用いたオンライン文字認識,” 情処学論, vol. 45, no. 12, pp. 2845-2855, 2004.
- [9] 三苦寛人, 内田誠一, 迫江博昭, “弾性マッチングに基づくオンライン文字認識における固有変形の利用,” 画像の認識・理解シンポジウム, vol. 2 of 2, pp. II-501-II-506, 2004
- [10] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, “UNIPEN project of on-line data exchange and recognizer benchmarks,” Proc. ICPR, pp. 29-33, 1994.
- [11] <http://hwr.nici.kun.nl/unipen/>
- [12] E. H. Ratzlaff, “Methods, report and survey for the comparison of diverse isolated character recognition results on the UNIPEN database,” Proc. ICDAR, pp. 623-628, 2003.
- [13] <http://www.alphaworks.ibm.com/tech/comparehwr>
- [14] S. D. Connell and A. K. Jain, “Template-based online character recognition,” Pattern Recognit., vol. 34, no. 1, pp. 1-13, 2001.
- [15] A. K. Jain and R. C. Dubes, Algorithms for clustering data, Prentice Hall, 1988.
- [16] R. Dubes and A. K. Jain, “Clustering techniques: the user dilemma” Pattern Recognit., vol. 8, pp.247-268, 1976.

付 録

1. CLUSTER のアルゴリズム

以下にクラスタリング手法 CLUSTER のアルゴリズムについて述べる．なお，以下でいう歪みは 2 パターン間の DP 距離 D_{DP} を用いて測定されるものである．

●初期化

Step0 $k = 1$

Step1 全学習パターンのセントロイドの選択

総歪みが最小となるサンプルをセントロイド x_1 として選択．

●クラスタの分割

Step2 セントロイドの追加

k 個のセントロイド $\{x_1, x_2, \dots, x_k\}$ から最も遠いサンプルを $k + 1$ 番目のセントロイド x_{k+1} として選択．

Step3 クラスタの形成

各サンプルについて，最も近いセントロイド x_p を選び，クラスタ C_p のメンバとする．

Step4 セントロイドの更新

各クラスタ C_p について，総歪みが最も小さくなるサンプルを新しいセントロイド x_p として選択．

Step5 k-means ループ

Step3, Step4 をセントロイド x_p が動かなくなるか、最大ループ回数に達するまで繰り返す。

Step6 微小クラスタの除去

各クラスタについてメンバ数が閾値以下となるクラスタは除去。

Step7 ループ

$k < K_{\max}$ ならば、 $k = k + 1$ として Step2 へ。

●クラスタの統合

Step8 ループ

k 個のクラスタから 2 つのクラスタ C_s, C_t を選ぶ全組合せについて Step9 ~ Step12 を行う。

Step9 2 つのクラスタの仮統合

2 つのクラスタ C_s, C_t を 1 つのクラスタとし、そのクラスタのセントロイド x_m を選択。

Step10 クラスタの再構成

各サンプルについて、Step8 で選ばれたセントロイド x_m と選ばれなかった $k - 2$ 個のセントロイドの中から最も近いセントロイド x_p を選ばせ、クラスタ C_p のメンバとする。

Step11 セントロイドの更新

各クラスタ C_p について、総歪みが最も小さくなるサンプルを新しいセントロイド x_p として選択。

Step12 k-means ループ

Step9, Step10 をセントロイド x_p が動かなくなるか、最大ループ回数に達するまで繰り返す。

Step13 2 つのクラスタの統合

k 個のクラスタから 2 つのクラスタ C_s, C_t を選ぶ全組合せの中から総歪みが最小となる 2 つのクラスタを統合し、 $k - 1$ 個のセントロイドを選択。

Step14 ループ

$k > 2$ ならば、 $k = k - 1$ として Step8 へ。

●終了判定

$k = 2, 3, \dots, K_{\max}$ それぞれにおけるの総歪みの最小値がいずれも更新されなければ終了。更新されていれば、 $k = 2$ として $k = 2$ のときの最小総歪みを与えるセントロイドを用いて Step2 からやり直す。