

固有変形を用いたオンライン文字認識手法の改良

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 筆者らはカテゴリ内文字変動を表現する固有変形を用いたオンライン文字認識手法を検討してきた。この方式では、入力パターンとあるカテゴリの標準パターンとのマッチングがそのカテゴリの固有変形からどの程度逸脱しているかを評価する。本報告はその続報であり、主に (i) 変形逸脱量の評価形式、(ii) 変形表現形式の2点について、改善ならびに検討について述べる。また、UNIPEN データベースのオンライン数字データを用いた認識実験により最高認識率 97.95% が得られ、本手法の有効性を確かめられた。

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

Improvements of On-line Character Recognition Based on Eigen-Deformations

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

[†] School of Engineering, Kyushu University

^{††} Graduate School 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 The authors have investigated into online character recognition technique with eigen-deformations, which express frequent deformations of each category. To reduce overfitting, this technique evaluates a divergence between eigen-deformations and a fitting result obtained by DP matching between an input pattern and a reference pattern. In this report, which is a further study of the previous investigation, we describe the improvements or investigation mainly on (i) how to evaluate the deformations, (ii) how to express the deformations. The results of recognition experiment on UNIPEN database showed the usefulness of the proposed technique.

Key words online character recognition, eigen-deformations, DP matching, QDF, UNIPEN database

1. まえがき

オンライン手書き文字認識においては、DP マッチング [1]~[3] もしくはその確率的拡張である HMM [4]~[6] が広く用いられている。DP マッチングは弾性マッチングアルゴリズムの一種であり、具体的には、入力パターンおよび標準パターンという2つの時系列データ間について、最適なサンプル点対応を与える手法である。この最適化の際、対応サンプル点間の座標値（および他の特徴量）の差異をパターン全体として最小化するという基準が用いられる。その結果、パターンの変形（非線形時間伸縮および座標値の変動）が補償され、パターン間の距離（もしくは類似度）評価が安定するという効果が得られる。

DP マッチング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、この効果の副作用として、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、それらの間の距離が過小評価されてしまう現象である。

この問題の対処法として、筆者らは、DP マッチングによる従来のパターン間距離に、その DP マッチング結果の事後評価値を重みを付けて加算したものを新たな距離尺度として用いることを提案している [7], [8]。この事後評価では、各カテゴリの固有変形を用いる。固有変形とは、そのカテゴリで起りやすい対応付けを、統計的手法により求めたものである。従って、固有変形からの逸脱量を評価すれば、逸脱の大きな場合として、

そのカテゴリでは起りにくいような対応付け，すなわち合わせ過ぎを検出できることになる。

本稿では，この前報 [7], [8] で提案した手法 (以下，従来法) を以下の 3 つの項について改良・再評価した結果について報告する。

1. 従来法では，前述のように DP マッチング距離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た．これに対し本稿では，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. 従来法の固有変形は，座標特徴のカテゴリ内変動を表現するものであった．本稿では，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繁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。

3. 前報では，評価対象のデータセットとして Ethem Alpaydin Digit [9] の数字データを用いていた．本稿では，より大規模で，また頻繁に利用されているデータセットである，UNIPEN Train-R01/V07 データベース [10], [11] の数字データ (1a) を用いた．さらに実験時の標準パターンについても，従来は試行錯誤的に作成していたのに対し，本稿では，クラスタリング手法を用いて自動的に設定する方針を採った。

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はその利点として挙げてられる．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，相当の時間を要する．また，文献 [8] で示したように，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 マッチング距離と事後評価量は，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，本稿では，DP マッチング距離は用いずに，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バ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 マッチング距離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，座標特徴以上に頻繁に用いられている．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。

上記第 3 項は，本手法の性能評価をより客観的に行うための検討である．UNIPEN データベースは，それを用いた各種認識手法の性能比較も報告 [12] されており，本手法も同データベースを用いることで，他手法との性能比較が容易になる．実際，本稿では，同じ DP を用いた最新の認識手法 Statistical dynamic time warping [3] よりも，同条件で高い認識率が得られることを示す。

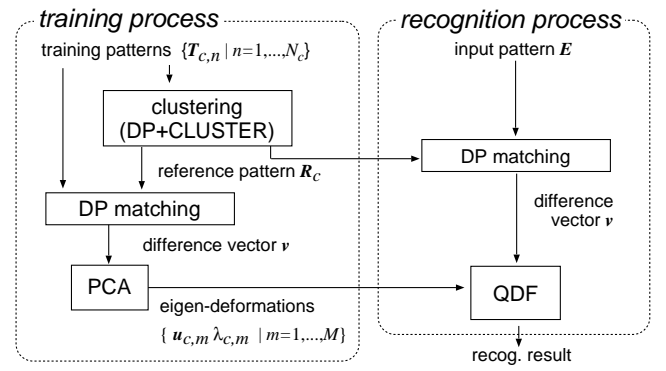


図 1 実験の流れ。

2. オンライン手書き文字の固有変形の推定

本節では，本手法 (図 1) の学習過程としてオンライン手書き文字の固有変形の推定法について述べる．この推定が (i) DP マッチングによる変形の採集と，(ii) その主成分分析，という 2 つのステップからなる点は前報 [7], [8] と同じである．ただし，位置情報だけでなく方向情報をも固有変形に含めて考える点は今回新たに検討するものである．したがって，今回用いる固有変形は，厳密には固有変形ではなく固有変動とでも称するべきものであるが，前報との統一性を図るべく本稿でも固有変形と呼ぶこととする。

2.1 DP マッチングによる変形の採集

第 c カテゴリの標準パターン R_c を時系列

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

と表し，同様に第 c カテゴリの第 n 番目の学習パターン $T_{c,n} (n = 1, \dots, N_c)$ を時系列

$$T_{c,n} = t_1, t_2, \dots, t_j, \dots, t_J \quad (2)$$

と表す．ここで r_i および t_j は筆点の特徴ベクトルであり，

$$r_i = (x_i^r, y_i^r, d_i^r)^T \quad (3)$$

$$t_j = (x_j^t, y_j^t, d_j^t)^T \quad (4)$$

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。

2 パターン R_c と $T_{c,n}$ 間のマッチングは，次の最適化問題により求まる。

[目的関数]

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

[制御変数 (自然数)]

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

[制約条件]

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

よく知られているように、この問題は DP を使うことで効率的に解ける。以下では、式 (5) の最小値を D_{DP} とする。多くのオンライン文字認識手法では、この D_{DP} がパターン間の距離としてそのまま用いられているが、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。制御変数 $j(1), \dots, j(I)$ は 2 パターンの対応関係を表す変数であるが、これを基にすれば標準パターンを基準とした場合の学習パターン $\mathbf{T}_{c,n}$ の変形 ($3I$ 次元差分ベクトル) $\mathbf{v}_{c,n}$ は次のように表現される。

$$\mathbf{v}_{c,n} = \left((x_1^r - x_{j(1)}^t, y_1^r - y_{j(1)}^t, d_1^r - d_{j(1)}^t), \dots, (x_i^r - x_{j(i)}^t, y_i^r - y_{j(i)}^t, d_i^r - d_{j(i)}^t), \dots, (x_I^r - x_{j(I)}^t, y_I^r - y_{j(I)}^t, d_I^r - d_{j(I)}^t) \right)^T \quad (6)$$

なお、前報では $\mathbf{v}_{c,n}$ 中に方向の差分は含まれていなかったが、本稿ではこのようにそれも含めて考える。

2.2 固有変形の推定

固有変形は頻出する変形であり、変形の成分分布の分散の大きな方向、いわゆる主軸に対応する。従って固有変形を求めるには、まず各カテゴリ毎の全 N_c 個の学習パターンについて前ステップを用いて N_c 個の差分ベクトル $\mathbf{v}_{c,n}$ を求め、次にそれらを主成分分析すればよい。具体的には、まず各カテゴリについて、平均

$$\bar{\mathbf{v}}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{v}_{c,n} \quad (7)$$

および共分散行列

$$\Sigma_c = \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{v}_{c,n} - \bar{\mathbf{v}}_c)(\mathbf{v}_{c,n} - \bar{\mathbf{v}}_c)^T \quad (8)$$

を求める。次にこの Σ_c を固有値展開すれば、各固有ベクトル $\{\mathbf{u}_{c,1}, \dots, \mathbf{u}_{c,m}, \dots, \mathbf{u}_{c,M}\}$ として固有変形が得られる。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。

3. 固有変形を用いた認識

前報 [7], [8] においては、固有変形を事後評価の形式で用いていた。具体的には、まず標準パターン \mathbf{R}_c とカテゴリ未知の入力パターン \mathbf{E} 間で DP マッチングを行い、距離 $D_{DP}(\mathbf{R}_c, \mathbf{E})$ ならびに \mathbf{E} の差分ベクトル \mathbf{v} を求める。次にその差分ベクトル \mathbf{v} と固有変形 $\{\mathbf{u}_{c,m}\}$ のマハラノビス距離を求め、最後に、その距離と $D_{DP}(\mathbf{R}_c, \mathbf{E})$ を加重和して評価に用いていた。

本手法では、前報とは異なり、 $D_{DP}(\mathbf{R}_c, \mathbf{E})$ との加算は行わず、固有変形に基づく 2 次識別関数単独で差分ベクトル \mathbf{v} を評価するという方針を採る。

3.1 2 次識別関数

2.1 節と同様の方法で求めた入力パターン \mathbf{E} と標準パターン \mathbf{R}_c 間の差分ベクトルを \mathbf{v} とする、合わせ過ぎを起こした際の差分ベクトルは、多くの場合、そのカテゴリの差分ベクトルの分布からは逸脱すると考えられる。そこで、変形の分布を正規分布と仮定した 2 次識別関数

$$D_{\text{tot}}(\mathbf{R}_c, \mathbf{E}) = (\mathbf{v} - \bar{\mathbf{v}}_c)^T \Sigma_c^{-1} (\mathbf{v} - \bar{\mathbf{v}}_c)$$

$$+ \log |\Sigma_c| + 3I \log 2\pi \quad (9)$$

を用いれば、合わせ過ぎが起きた際は距離値が大きくなる。この仕組みにより、合わせ過ぎによる誤認識を排除できると考えられる。なお、2 次識別関数において右辺の最終項が省略されることが多いが、本稿ではカテゴリ毎にパターンの次元数が異なるため省略できない。

以下、本手法では位置情報と方向情報に相関が無いと仮定する。これにより、位置差分と方向差分をそれぞれ独立に主成分分析でき、また両者のレンジの違いの問題が解消される。さらに、 $3I$ 次元ベクトル \mathbf{v} を $2I$ 次元と I 次元に分離して推定できるので、必要な学習パターン数の低減効果もある。この場合、式 (9) は以下のように展開できる。

$$D_{\text{tot}}(\mathbf{R}_c, \mathbf{E}) = D_{\text{pos}}(\mathbf{R}_c, \mathbf{E}) + D_{\text{dir}}(\mathbf{R}_c, \mathbf{E}) \quad (10)$$

$$\begin{aligned} D_{\text{pos}}(\mathbf{R}_c, \mathbf{E}) &= (\mathbf{v}^{x,y} - \bar{\mathbf{v}}_c^{x,y})^T (\Sigma_c^{x,y})^{-1} (\mathbf{v}^{x,y} - \bar{\mathbf{v}}_c^{x,y}) \\ &+ \log |\Sigma_c^{x,y}| + 2I \log 2\pi \\ &= \sum_{m=1}^{2I} \frac{1}{\lambda_{c,m}^{x,y}} ((\mathbf{v}^{x,y} - \bar{\mathbf{v}}_c^{x,y})^T \mathbf{u}_{c,m}^{x,y})^2 \\ &+ \sum_{m=1}^{2I} \log \lambda_{c,m}^{x,y} + 3I \log 2\pi \end{aligned} \quad (11)$$

$$\begin{aligned} D_{\text{dir}}(\mathbf{R}_c, \mathbf{E}) &= (\mathbf{v}^d - \bar{\mathbf{v}}_c^d)^T (\Sigma_c^d)^{-1} (\mathbf{v}^d - \bar{\mathbf{v}}_c^d) \\ &+ \log |\Sigma_c^d| + I \log 2\pi \\ &= \sum_{m=1}^I \frac{1}{\lambda_{c,m}^d} ((\mathbf{v}^d - \bar{\mathbf{v}}_c^d)^T \mathbf{u}_{c,m}^d)^2 \\ &+ \sum_{m=1}^I \log \lambda_{c,m}^d + I \log 2\pi \end{aligned} \quad (12)$$

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 (12) 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、本報告では次の修正 2 次識別関数 [17]

$$\begin{aligned} D_{\text{pos}}(\mathbf{R}_c, \mathbf{E}) &\sim \frac{1}{\lambda_{c,M'+1}^{x,y}} \|\mathbf{v}^{x,y} - \bar{\mathbf{v}}_c^{x,y}\|^2 \\ &+ \sum_{m=1}^{M^{x,y}} \left(\frac{1}{\lambda_{c,m}^{x,y}} - \frac{1}{\lambda_{c,M^{x,y}}^{x,y}} \right) ((\mathbf{v}^{x,y} - \bar{\mathbf{v}}_c^{x,y})^T \mathbf{u}_{c,m}^{x,y})^2 \\ &+ \log (\lambda_{c,M^{x,y}+1}^{x,y})^{2I-M^{x,y}} \prod_{m=1}^{M^{x,y}} \lambda_{c,m}^{x,y} + 2I \log 2\pi \end{aligned} \quad (13)$$

を用いる。 $D_{\text{dir}}(\mathbf{R}_c, \mathbf{E})$ についても同様なのでここでは省略する。これにより、高次の固有値に含まれる数値誤差の影響を抑えることができる。パラメータ $M^{x,y}, M^d$ の決め方には様々な方法が考えられるが、本報告では累積寄与率 $\sum_{m=1}^M \lambda_{c,m} / \sum_{m=1}^M \lambda_{c,m}$ が、あるしきい値 θ をはじめて越えた時の $M^{x,y}, M^d$ を用いることとした。後述する認識実験においては、この θ は位置情報と方向情報についてそれぞれ用意し、学習データを対象とした実験により最適化した。

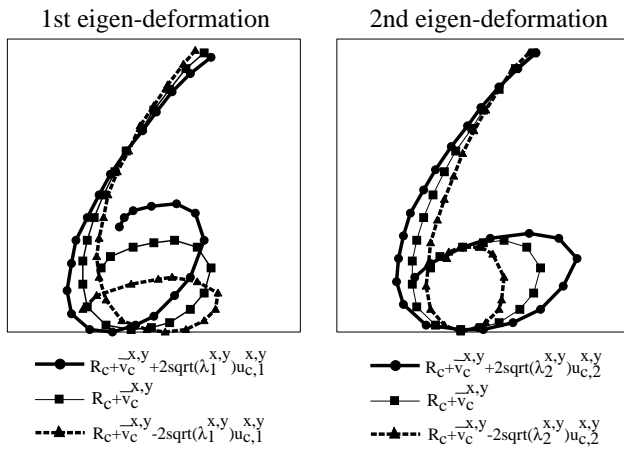


図2 上位2つの位置情報の固有変形 $u_{c,1}^{x,y}, u_{c,2}^{x,y}$ によって変形された標準パターン“6”。

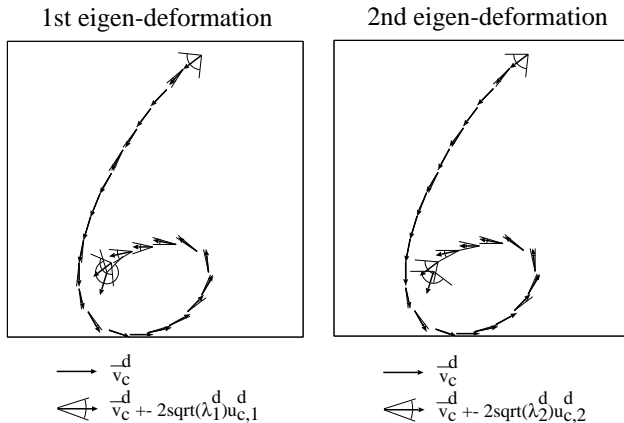


図3 上位2つの方向情報の固有変形 $u_{c,1}^d, u_{c,2}^d$ によって得られた方向の範囲“6”。

3.2 従来法 [7], [8] との比較

本手法と従来法 [7], [8] の違いについて本節で再びまとめる。

(i) 従来法では、固有変形に基づく事後評価量 (マハラノビス距離) と DP 距離を加重和したもので識別していた。この重みの最適値は、データの変動に影響されていた。これに対し本手法では、2次識別関数単独での認識手法と採っており、こうした重みを考える必要がない。また、パターン長を揃えるための正規化項についても、正規分布に対するベイズ識別の原理に基づいて、より自然な形で導入されている。

(ii) 従来法においては DP マッチング時のみ方向情報も用い、固有変形ならびにそれに基づく識別時には位置情報のみしか使用していなかった。これに対し、本手法では方向情報も用いる。これにより認識の精度が向上すると期待される。

4. 認識実験

4.1 データベース

本報告では、実験試料として UNIPEN Train-R01/07 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、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、これら全 15,953 サンプ

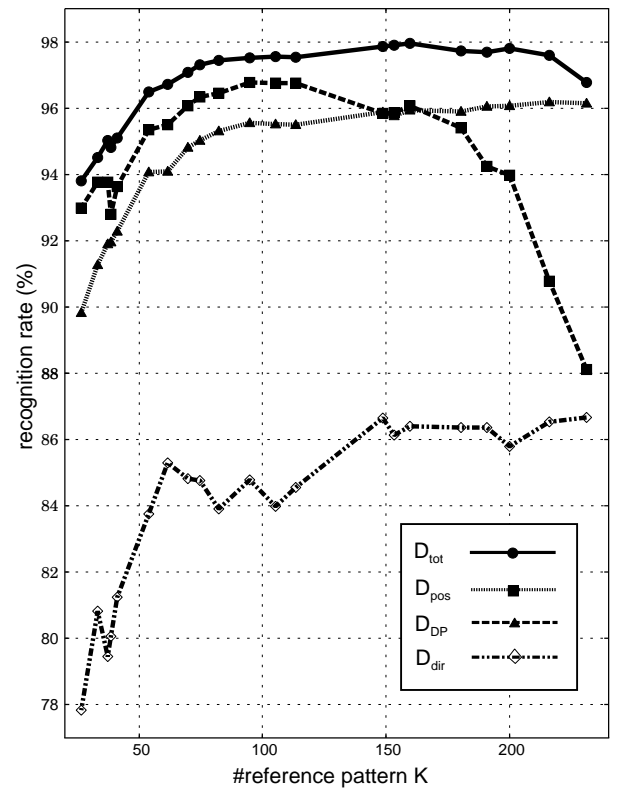


図4 テスト用データの認識率。

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 multi writer 方式を採用し、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 4.2 の標準パターン生成および固有変形推定に、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。

4.2 クラスタリングによる標準パターン設定ならびに固有変形の推定

標準パターンの生成は、文献 [14] にならひ、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。アルゴリズムについては付録 1. に載せる。したがって、CLUSTER を 1 度実行すれば各カテゴリ最大 K 個

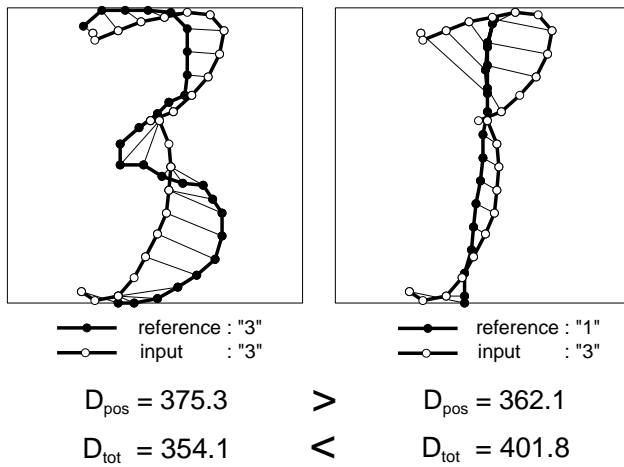


図 5 固有変形の利用による改善例

までの標準パターンが準備できることになる。

固有変形の推定には、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。

求まった 1 個, 2 個, ..., K 個の場合の標準パターンの内、何個の標準パターンを用いるかは、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、各クラスター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 K を用いる。よって、 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。

4.3 固有変形の推定結果

図 2 に実際に求まった数字の “6” の 1 標準パターンについての上位 2 つの位置情報の固有変形 ($\mathbf{u}_{c,1}^{x,y}, \mathbf{u}_{c,2}^{x,y}$) を示す。この図より、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。

同様に図 3 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、DP の制約条件である始点と終点を一致させた等の方向のばらつきが求まっていることがわかる。また、ループ部の上部で方向の変動が見られるが、これはループ部の上部の上下を表すことがわかる。

4.4 認識結果と考察

図 4 に、標準パターン数 K (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} 単独による認識率も図に示す。

図 4 より D_{tot} による認識率を見ると、 $K = 160$ 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 $K = 150$ の場合について見ると、本手はそれより高い 97.85% であった。これは、STDW 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であると考えられる。

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 5 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” との位置差分が小さく、その結果 D_{pos} (“1”, “3”) は反応せず、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”) が大きくなり、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 4 のグラフでは表示していないが、従来法 [7], [8] に倣って D_{tot} と D_{DP} を加重和した

$$D_{conv}(\mathbf{R}_c, \mathbf{E}) = (1 - \alpha)D_{DP} + \alpha D_{tot} \quad (14)$$

による認識率についても実験を行った。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、全体的にはほぼ同程度の認識率となることがわかった。具体的には、 $k = 41, 82, 160$ のとき、 D_{tot} による認識率がそれぞれ 95.09%, 97.43%, 97.95% であったのに対し、 D_{conv} の認識率は 95.17%, 97.42%, 98.02% であった。これにより DP マッチングによる距離 D_{DP} を加重和という形で積極的に用いなくても 2 次識別関数 D_{tot} 単独でほぼ同等の性能があることがわかった。なお、この $k = 82$ の場合、式 (14) の加重和を取ることでわずかであるが劣化していることになる。このように学習データで定めた α がテストデータにおいて必ずしも適当でないことがわかる。さらに、文献 [7], [8] 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。この結果は、筆記者独立の場合に D_{conv} の結果が D_{tot} をさらに下回る可能性を示唆している。この点を含め筆記者独立という設定については、今後、検証する必要があると考えている。

5. まとめ

前報で提案した手法を変形評価形式・変形表現形式について改良し、その性能を検討した。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。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。

変形評価形式に関しては、本報告では学習・テスト用データの筆記者が独立でない場合でしか評価していない。そのため、今後、筆記者独立の場合でも本手法の性能を評価する必要がある。関連して学習・テストデータの筆記者を 1 名に限定した場合についても評価を行う予定である。また、固有変形の別の利用形態として、固有変形をマッチング過程へ直接組み込む方針の検討が挙げられる。この方針を採用することで、認識精度だけでなく計算量的にも有効ではないかと考えている。

謝 辞

本研究の一部は文部科学省科学研究費補助金(若手 B No.14780293), および総務省戦略的情報通信研究開発推進制度の支援を受けた。

文 献

- [1] 藤原 暁和, 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] H. Mitoma, S. Uchida and H. Sakoe, “Online character recognition using eigen-deformations,” Proc. IWFHR, pp. 3-8, 2004.
- [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,” 情報学論, vol. 45, no. 12, pp. 2845-2855, 2004.
- [9] <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pendigits/>
- [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.
- [17] 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.

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