Online Character Recognition Using Eigen-Deformations

Hiroto Mitoma, Seiichi Uchida, and Hiroaki Sakoe
Kyushu University, Hakozaki 6–10–1, Higashi-ku, Fukuoka-shi, 812–8581 Japan
{mitoma, uchida, sakoe}@human.is.kyushu-u.ac.jp

Abstract

In online character recognition based on elastic matching, such as dynamic programming matching, many of misrecognitions are often caused by overfitting, which is the phenomenon that the distance between reference pattern of an incorrect category and an input pattern is underestimated by unnatural matching. In this paper, a new recognition technique is proposed where category-specific deformations, called eigen-deformations, are utilized to suppress those misrecognitions. Generally, matching results at overfitting are not consistent with the eigen-deformations. Thus, the overfitting can be detected and penalized by a posterior evaluation of this inconsistency. The result of a recognition experiment showed the usefulness of the proposed technique.

1 Introduction

In online character recognition, elastic matching, or nonlinear time warping, has been employed for matching input and reference patterns while compensating the deformations of their strokes. Dynamic programming (DP) matching is one of the most common elastic matching techniques and has been employed in not only classic [1, 2, 3] but also recent [4] online character recognizers. A probabilistic extension of DP, called Hidden Markov model (HMM), has also been widely employed [5, 6, 7].

Although good performance of those elastic matching-based online character recognizers has been reported, it has also been pointed out that their performance is often degraded due to *overfitting*, which is the phenomenon that the distance between the reference pattern of an incorrect category and an input pattern is underestimated due to unnatural matching. For example, an input pattern “1” might be misrecognized as “7” by matching the horizontal part of reference pattern “7” to the beginning part of “1”. Unfortunately, DP matching has a limitation on incorporating constraints to exclude overfitting, because of its Markovian property. Specifically speaking, DP can incorporate *local* constraints to control the relation between non-consecutive points, and can not incorporate *global* constraints effective to exclude overfitting. Even sophisticated versions of DP matching, such as HMM [5, 6, 7] and statistical DP [4], have the same limitation.

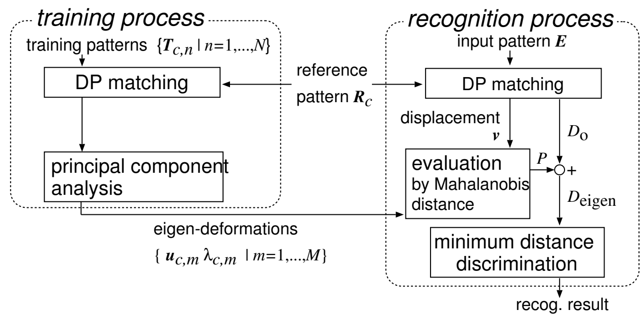

Figure 1. Overview of the proposed technique.

In this paper, a new DP-based online character recognition technique with less misrecognitions due to overfitting is proposed. The central idea of the proposed technique is to employ a posterior evaluation procedure where category-specific deformations, hereafter called *eigen-deformations*, are utilized. For example, in category “7”, the slant of its (near-)vertical part is often observed. Thus, this slant can be considered as an eigen-deformation of “7”. On the other hand, severe shrinkage of the horizontal part is rarely observed and thus not an eigen-deformation of “7”. This intuitive example suggests that the matching results at overfitting are not consistent with the eigen-deformations and therefore the overfitting can be detected and penalized by evaluating this inconsistency. As we can see later in this paper , the eigen-deformations can be estimated automatically by using elastic matching and principal component analysis (PCA). **Figure 1** is the diagram of the proposed technique.

The original idea of estimating deformation tendencies by PCA can be found in [8] and has been successfully applied to off-line handwritten character recognition [9, 10, 11]. Note that, while the proposed technique is based on DP matching, it can be readily extended to HMM and even other elastic matching schemes, such as local affine transformation (LAT) [12].

2 Estimation of Eigen-Deformations

In this section, the procedure for estimating eigen-deformations is provided. This procedure is divided into two steps: (i) the automatic collection of actual deformations by DP matching, and (ii) the analysis of the deformations by PCA. The following is the detail of the procedure.

2.1 Collection of actual deformations using DP matching

Let $\boldsymbol{R}_c = \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i, \ldots, \boldsymbol{r}_I$ denote the reference pattern of category c and $\boldsymbol{T}_{c,n} = \boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_j, \ldots, \boldsymbol{t}_J$ denote the nth training pattern of category c, where $n = 1, \ldots, N_c$ and $c = 1, \ldots, C$. Although $\boldsymbol{r}_i$, $\boldsymbol{t}_j$, I, and J should be denoted like $\boldsymbol{r}_{c,i}$, $\boldsymbol{t}_{c,n,j}$, I_c, and $J_{c,n}$, simpler notations are used whenever there is no confusion. The feature vectors $\boldsymbol{r}_i$ and $\boldsymbol{t}_j$ are composed of x-coordinate, y-coordinate, and local direction, and thus can be denoted as $\boldsymbol{r}_i = (x_i^r, y_i^r, d_i^r)^T$ and $\boldsymbol{t}_j = (x_j^t, y_j^t, d_j^t)^T$.

The problem of the elastic matching between two patterns $\boldsymbol{R}_c$ and $\boldsymbol{T}_{c,n}$ is defined as the following constrained optimization problem.

[Objective function]

$$\frac{1}{I}\sum_{i=1}^{I} \left\| \boldsymbol{r}_i - \boldsymbol{t}_{j(i)} \right\| \rightarrow \text{minimize}$$

[Control variables]

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

[Constraint]

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

where $\|\boldsymbol{x}\|$ is the Euclidean norm of vector $\boldsymbol{x}$. This problem can be solved effectively by a DP algorithm (whose detail is omitted here).

Let $D_0(\boldsymbol{R}_c, \boldsymbol{T}_{c,n})$ denote the minimum value of the objective function, i.e.,

$$D_0(\boldsymbol{R}_c, \boldsymbol{T}_{c,n}) = \min_{j(1),\ldots,j(i),\ldots,j(I)} \frac{1}{I}\sum_{i=1}^{I} \left\| \boldsymbol{r}_i - \boldsymbol{t}_{j(i)} \right\|. \tag{1}$$

This value D_0 is a deformation-invariant distance between $\boldsymbol{R}_c$ and $\boldsymbol{T}_{c,n}$, and thus has been employed as a discriminant function robust to deformations. The recognizer based on D_0, however, suffers from overfitting as shown in Section 4 and 5.

Using $j(1), \ldots, j(I)$, which show the optimal matching between two patterns $\boldsymbol{R}_c$ and $\boldsymbol{T}_{c,n}$, the $2I$-dimensional displacement vector [1] can be obtained as

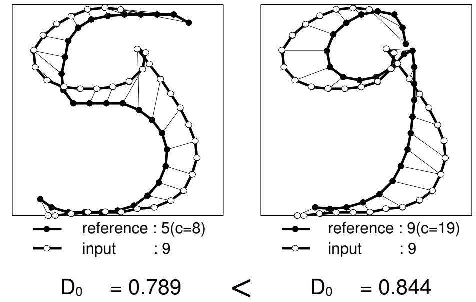

Figure 2. Example of misrecognitions due to overfitting. The parenthesized value of c is the reference pattern number c of Fig 3.

$$\begin{aligned} \boldsymbol{v}_{c,n} = \big(& (x_1^r - x_{j(1)}^t, y_1^r - y_{j(1)}^t), \ldots, \\ & (x_i^r - x_{j(i)}^t, y_i^r - y_{j(i)}^t), \ldots, \\ & (x_I^r - x_{j(I)}^t, y_I^r - y_{j(I)}^t) \big)^T. \end{aligned} \tag{2}$$

This displacement vector represents the deformation of $\boldsymbol{T}_{c,n}$ from the standard shape $Rvec_c$. Thus, by matching the reference pattern $\boldsymbol{R}_c$ to all training patterns, N_c actual deformations of category c, $\{\boldsymbol{v}_{c,n} | n = 1, \ldots, N_c\}$, are automatically collected.

2.2 Estimation of eigen-deformations

The eigen-deformations of category c are frequent deformations of the category, and thus defined as the principal axes of distribution of the displacement vectors $\{\boldsymbol{v}_{c,n}\}$ in M-dimensional space, where $M = 2I$. Namely, the eigen-deformations are estimated by applying PCA to N_c displacement vectors $\{\boldsymbol{v}_{c,n}\}$. Specifically, the eigen-deformations are derived as the eigenvectors $\{\boldsymbol{u}_{c,m} | m = 1, \ldots, M\}$ of the covariance matrix:

$$\Sigma_c = \frac{1}{N_c}\sum_{n=1}^{N_c} (\boldsymbol{v}_{c,n} - \overline{\boldsymbol{v}}_c)(\boldsymbol{v}_{c,n} - \overline{\boldsymbol{v}}_c)^T, \tag{3}$$

where $\overline{\boldsymbol{v}}_c$ is the mean vector of $\{\boldsymbol{v}_{c,n}\}$, i.e.,

$$\overline{\boldsymbol{v}}_c = \frac{1}{N_c}\sum_{n=1}^{N_c} \boldsymbol{v}_{c,n}. \tag{4}$$

The contribution, or the frequency, of the eigen-deformation $\boldsymbol{u}_{c,m}$ can be measured by its corresponding eigenvalue $\lambda_{c,m}$, where $\lambda_{c,m} \geq \lambda_{c,m+1}$. Thus, $\boldsymbol{u}_{c,1}$ represents the most frequent eigen-deformation of category c.

[1] Since all training patterns are aligned to the same reference pattern $\boldsymbol{R}_c$, the dimension of their displacement vectors is always equal to $2I_c$, regardless of their length $J_{c,n}$.

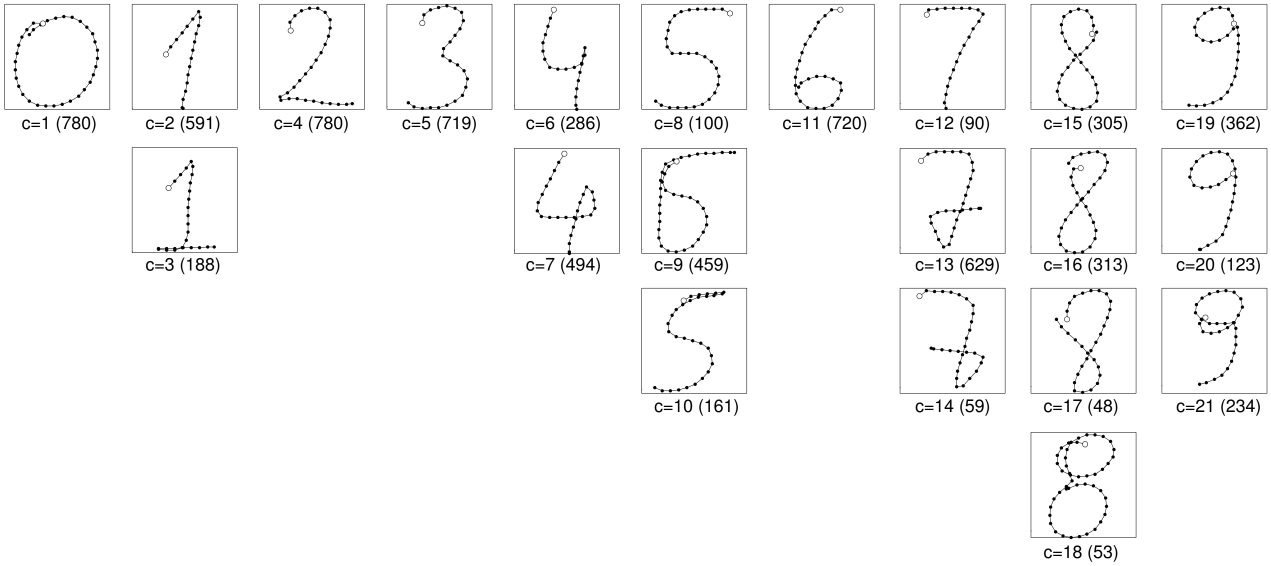

Figure 3. Reference patterns. Multiple reference patterns were prepared for several categories with stroke-order and/or heavy shape variations (e.g.,"8"). The parenthesized number represents N_c.

3 Recognition Using Eigen-Deformations

Online character recognition within the framework of the conventional DP matching suffers from misrecognitions due to overfitting. An example of such misrecognitions is shown in **Fig 2**, where the input pattern "9" is misrecognized as "5" by D_0 underestimated by the unnatural matching, i.e., the overfitting, between "9" and "5". Such overfitting is caused by the regardlessness of category-dependent deformation tendencies during DP matching. In fact, the unnatural matching of this example is not consistent with the deformation tendencies of category "5".

In this section, a new DP matching-based online character recognition technique is described, where the eigen-deformations are newly employed to suppress the misrecognitions due to overfitting. The proposed recognition technique is composed of the following four steps.

1. Perform the conventional DP matching between $\boldsymbol{R}_c$ and an unknown input pattern $\boldsymbol{E}$ to obtain $D_0(\boldsymbol{R}_c, \boldsymbol{E})$ and optimal matching $j(1), \ldots, j(I)$.
2. Derive the displacement vector $\boldsymbol{v}$ from $j(1), \ldots, J(I)$ (cf.(2)).
3. Calculate $P_c(\boldsymbol{v})$, which evaluates the inconsistency between $\boldsymbol{v}$ and the eigen-deformations of category c.
4. Perform the discrimination according to a new distance

$$D_{\text{eigen}}(\boldsymbol{R}_c, \boldsymbol{E}) = (1-\alpha)D_0(\boldsymbol{R}_c, \boldsymbol{E}) + \alpha P_c(\boldsymbol{v}) \quad (5)$$

where α is a constant$(0 \leq \alpha \leq 1)$.

The posterior evaluation term $P_c(\boldsymbol{v})$ is defined as

$$P(\boldsymbol{R}_c, \boldsymbol{E}) = \frac{1}{I}\sqrt{p(\boldsymbol{R}_c, \boldsymbol{E})}, \quad (6)$$

where $p(\boldsymbol{R}_c, \boldsymbol{E})$ is the quadratic Mahalanobis distance between $\overline{\boldsymbol{v}}_c$ and $\boldsymbol{v}$, i.e.,

$$\begin{aligned} p(\boldsymbol{R}_c, \boldsymbol{E}) &= (\boldsymbol{v} - \overline{\boldsymbol{v}}_c)^T \Sigma_c^{-1} (\boldsymbol{v} - \overline{\boldsymbol{v}}_c) \\ &= \sum_{m=1}^{M} \frac{1}{\lambda_{c,m}} ((\boldsymbol{v} - \overline{\boldsymbol{v}}_c)^T \boldsymbol{u}_{c,m})^2. \end{aligned} \quad (7)$$

In the case of overfitting, the displacement vector $\boldsymbol{v}$ will not be consisted with the eigen-deformations of category c and thus $P_c(\boldsymbol{v})$ becomes large. This means that $D_{\text{eigen}}(\boldsymbol{R}_c, \boldsymbol{E})$ also becomes large and the input $\boldsymbol{E}$ will not be discriminated into category c. Consequently the misrecognition due to overfitting can be suppressed.

It is well known that the estimation errors of higher-order eigenvalues are amplified in (7). Thus, in practice, the quasi-Mahalanobis distance [13] is employed, where the higher-order eigenvalues $\lambda_{c,m}$ $(m = M'+1, \ldots, M)$ are replaced by $\lambda_{c,M'+1}$, i.e.,

$$\begin{aligned} & p(\boldsymbol{R}_c, \boldsymbol{E}) \\ \sim \quad & \sum_{m=1}^{M'} \frac{1}{\lambda_{c,m}} ((\boldsymbol{v} - \overline{\boldsymbol{v}}_c)^T \boldsymbol{u}_{c,m})^2 \\ & + \sum_{m=M'+1}^{M} \frac{1}{\lambda_{c,M'+1}} ((\boldsymbol{v} - \overline{\boldsymbol{v}}_c)^T \boldsymbol{u}_{c,m})^2 \\ = \quad & \frac{1}{\lambda_{c,M'+1}} \|\boldsymbol{v} - \overline{\boldsymbol{v}}_c\|^2 \\ & + \sum_{m=1}^{M'} \left(\frac{1}{\lambda_{c,m}} - \frac{1}{\lambda_{c,M'}} \right) ((\boldsymbol{v} - \overline{\boldsymbol{v}}_c)^T \boldsymbol{u}_{c,m})^2. \end{aligned} \quad (8)$$

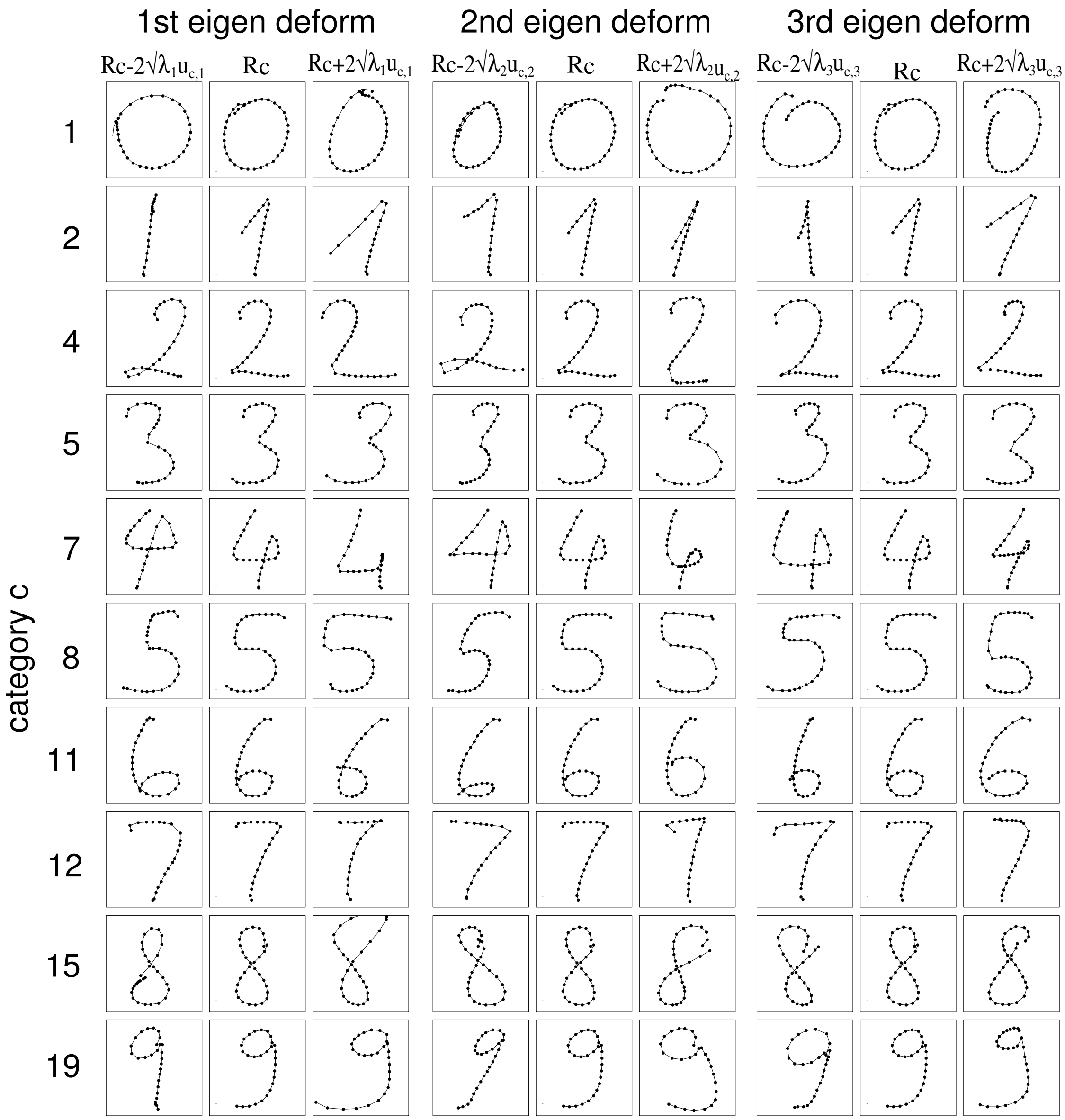

Figure 4. Reference pattern deformed by the first three eigen-deformations.

While various methods for determining the parameter M' can be considered, the smallest M' which satisfy $\sum_{m=1}^{M'} \lambda_{c,m} / \sum_{m=1}^{M} \lambda_{c,m} \geq 0.9$ was used in the recognition experiment of the next section.

4 Experimental Results

4.1 Dataset

About 10,000 isolated online digit samples from the UNIPEN format database called Ethem Alpaydin Digit[14] were used in our experiment. This database consists of 7,494 training samples (i.e., about 750 samples per category) and 3,498 test samples (i.e. 350 samples per category). The training samples were used as $\boldsymbol{T}_{c,n}$ for estimating eigen-deformations and the test samples were used as unknown input patterns $\boldsymbol{E}$.Note that 30 writers of the training samples are independent of 14 writers of the test samples.

All samples were preprocessed as follows. First, each pen-up part was connected by a line segment in order to eliminate stroke-number variations. Thus, all character patterns subjected to the experiment were one-stroke characters. Then, their size was normalized to 128×128, keeping aspect ratio. Finally, resampling was performed so that the distance between consecutive points became constant.

Table 1. Major misrecognitions decreased by D_{eigen} (a), and increased by D_{eigen} (b).

	input→ result	#misrecog. D_0	#misrecog. D_{eigen}	#diff
(a)	“9”→“5”	10	1	9
	“0”→“8”	16	9	7
	“1”→“7”	6	0	6
	“2”→“7”	6	1	5
	“3”→“7”	9	6	3
(b)	“7”→“9”	8	12	4
	“5”→“9”	5	8	3

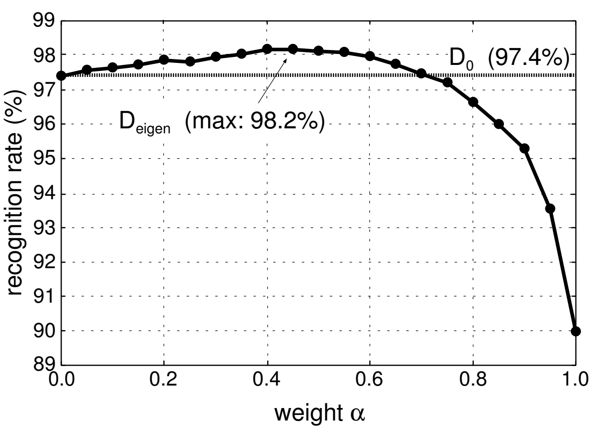

Figure 5. Recognition rate.

4.2 Reference patterns

After observing all training patterns, reference patterns $\boldsymbol{R}_c$ were designed manually. For categories with stroke-order and/or heavy shape variations (i.e., “1”, “4”, “5”, “7”, “8”, and “9”), multiple reference patterns were prepared. The reference patterns were also normalized and resampled one-stroke characters like input patterns. **Figure 3** shows all 21 reference patterns. As indicated by the fourth reference pattern of digit “8”, the training patterns include many unusual patterns.

Note that there are some “gaps” between the test samples and the training samples because of their writer-independence. For example, although the training samples include 48 samples similar to the third reference pattern of “8” (which is written in inverse direction), the test samples include no such sample. In addition, the test samples include only one sample similar to the fourth reference pattern of “8”. As shown in 4.4, the performance of the proposed technique was degraded due to these gaps.

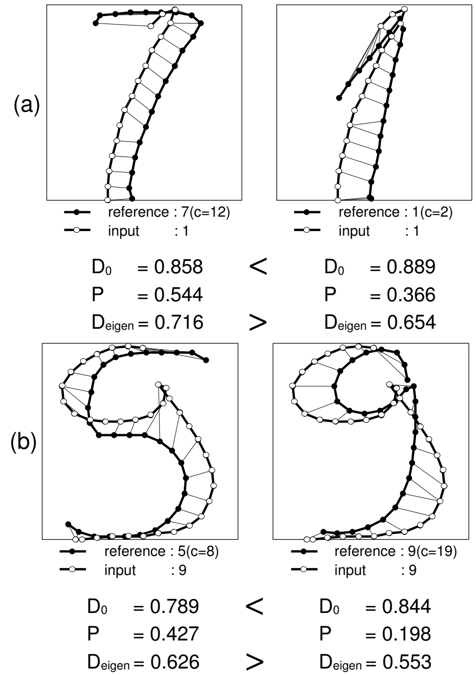

Figure 6. Samples correctly recognized by the proposed technique.

4.3 Estimated eigen-deformations

For the digit with one reference pattern (i.e., “0”, “2”, “3”, and “6”) the estimation of the eigen-deformations were performed according to the procedure of Section 2. Namely, DP matching was firstly performed between the reference pattern $\boldsymbol{R}_c$ and every training pattern $\boldsymbol{T}_{c,n}$ for collecting N_c displacement vectors $\{\boldsymbol{v}_{c,n}|n = 1, \ldots, N_c\}$ of category c. The collected vectors were then subjected to PCA and the eigen-deformations $\{\boldsymbol{u}_{c,m}\}$ and corresponding eigenvalues $\{\lambda_{c,m}\}$ were obtained. On the other hand, the digit with $K(> 1)$ reference patterns (i.e., “1”, “4”, “5”, “7”, “8”, and “9”), its training patterns were firstly decomposed into K subsets by the minimum-distance discrimination according to the DP matching distance between each training pattern and those K reference patterns. Then, for each of K reference patterns, eigen-deformations were estimated using the training patterns belonging to the corresponding subset.

Figure 4 shows reference patterns deformed by the top three eigen-deformations $\boldsymbol{u}_{c,1}$, $\boldsymbol{u}_{c,2}$, and $\boldsymbol{u}_{c,3}$. From this figure, it is suggested that the eigen-deformations estimated in the proposed framework are “reasonable”. That is, deformations frequently observed in actual characters were detected as primary eigen-deformations. For example, the first eigen-deformation of “0” ($c = 1$) represents the variations of beginning point. In “6” ($c = 11$), the first, the second, and the third eigen-deformations represent the variations of global slant angle, the height of the loop part, and the width of the loop part, respectively.

4.4 Recognition result

Figure 5 shows the recognition rates of the proposed technique as a function of α. When $\alpha = 0$, the proposed technique is reduced to the conventional technique, i.e., the minimum distance discrimination with D_0. The highest recognition rate was 98.2% and higher than 97.4% of the conventional technique. Their difference 0.8% consisted of 35 improved samples and 8 degraded samples.

Table 1 (a) shows major improved samples, which are correctly recognized by the proposed technique (D_{eigen}) and misrecognized by the conventional technique (D_0). **Fig. 6** shows two examples of those improved samples. These examples clarify the ability of the proposed technique on the suppression of misrecognitions due to overfitting. The input pattern "1" of **Fig. 6**(a) was originally misrecognized as "7", because the distance D_0 between "1" and the reference pattern "7" was underestimated by overfitting. The posterior evaluation term P, however, could detect this overfitting (i.e., became larger) and finally the test sample "1" was correctly recognized by D_{eigen}. In **Fig. 6**(b), since the deformation that the curving part of "5" becomes closer to its top horizontal part is rarely observed in training patterns, the term P became larger at the matching between "5" and "9".

Table 1 (b) shows the degraded samples, which are misrecognized by the proposed technique and correctly recognized the conventional technique. By observing the misrecognition pairs of "7" and "5", it was found that the shapes of those "7"s are rarely found in training samples. (As noted in 4.2, there are gaps between the test and the training patterns.) Thus, the eigen-deformations, which are estimated from the training samples, can not cover the deformations of those misrecognized "7"s . A possible remedy to reduce such misrecognitions will be the use of rich training patterns covering all deformations.

5 Conclusion

A new elastic matching-based online character recognition technique was proposed. In the technique, category-specific deformations, called eigen-deformations, are employed to suppress misrecognitions due to overfittng. Specifically, a posterior evaluation procedure is incorporated into an elastic matching-based recognizer for detecting and penalizing overfitting by the divergence from the eigen-deformations. The results of the experiment with about 10,000 online numeral patterns showed that (i) reasonable eigen-deformations can be automatically estimated by elastic matching and principal component analysis (PCA) and (ii) higher recognition rates can be attained by suppressing the misrecognitions due to overfitting.

Acknowledgement

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology in Japan under a Grant-in-Aid for Scientific Research No.14780293 and Public Management, Home Affairs, Posts and Telecommunications in Japan under Strategic Information and Communications R&D Promotion Programme (SCOPE).

References

[1] K. Ikeda, T. Yamamura, Y. Mitamura, S. Fujiwara, Y. Tominaga, and T. Kiyono, "On-line recognitoin of handwritten characters utilizing positional and stroke vector sequence," Pattern Recognition, vol. 13, no. 3, pp. 191-206, 1981.

[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] D. J. Burr, "Designing a handwriting reader", IEEE Trans. PAMI, vol. PAMI-5, no. 5, pp. 554-559, 1983.

[4] C. Bahlmann and H. Burkhardt, "The writer independent online handwriting recognition system flog on hand and cluster generative statistical dynamic time warping," IEEE Trans. PAMI, vol. 26, no. 3, pp. 299-310, 2004.

[5] R. Nag, K. H. Wong, and F. Fallside, "Script recognition using hidden Markov models," Proc. ICASSP, pp. 2071–2074, vol. 3 of 4, 1986.

[6] J. Hu, M.-K. Brown, and W. Turin, "HMM based online handwriting recognition," IEEE Trans. PAMI, vol. 18, no. 10, pp. 1039-1045, 1996.

[7] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line Kanji handwriting recognition," Proc. ICDAR, pp. 491–495, 2001.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application," Comput. Vis. Image Und., vol. 61, no. 1, pp. 38–59, Jan. 1995.

[9] D. Shi, S. R. Gunn, and R. I. Damper, "Handwritten Chinese radical recognition using nonlinear active shape models," IEEE Trans. PAMI, vol. 25, no. 2, pp. 277–280, Feb. 2003.

[10] S. Uchida and H. Sakoe, "Eigen-deformations for elastic matching based handwritten character recognition," Pattern Recognition, vol. 36, no. 9, pp.2031-2040, 2003.

[11] S. Uchida and H. Sakoe, "Handwritten character recognition using elastic matching based on a class-dependent deformation model," Proc. ICDAR, vol. 1 of 2, pp. 163–167, Aug. 2003.

[12] T. Wakahara, "On-line cursive script recognition using local affine transformation," Proc. ICPR, vol. 2 of 2, pp. 1133-1137, 1988.

[13] M. Kurita et al., "Handprinted KANJI and HIRAGANA character recognition using weight direction index histograms and quasi-Maharanobis distance," IEICE Tech. Rep.,PRL82-79, 1982.(in Japanese)

[14] `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pendigits/`