Recognition of Rotated Characters by Eigen-space

Hiroyuki Hase[#1], Toshiyuki Shinokawa[#2], Masaaki Yoneda[#3], Ching Y. Suen[#4]

#1 Dept. of *Information Science, Fukui University, Japan*
hase@fuis.fuis.fukui-u.ac.jp

#2 *Computer Engineering, Toyama National College of Maritime Technology, Japan*
sinokawa@susanoo.toyama cmtt.ac.jp

#3 *Intellectual Information Systems Engineering, Toyama University, Japan*
yoneda@pat.iis.toyama-u.ac.jp

#4 *Centre for Pattern Recognition and Machine Intelligence, Concordia University, Canada*
suen@cenparmi.concordia.ca

Abstract

In this paper, we present a method of recognizing inclined, rotated characters. First we construct an eigen sub-space for each category using the covariance matrix which is calculated from a sufficient number of rotated characters. Next, we can obtain a locus by projecting their rotated characters onto the eigen sub-space and interpolating between their projected points. An unknown character is also projected onto the eigen sub-space of each category. Then, the verification is carried out by calculating the distance between the projected point of the unknown character and the locus. In our experiment, we obtained quite good results for the CENTURY font of 26 capital letters of the English alphabet (A, B, ... ,Z). This method has the added advantage of obtaining the recognition result (category) and angle of inclination at the same time

1. Introduction

Printed documents and advertisements often contain texts in which the characters are distorted, inclined, rotated, or stylized to catch people's attention. Recognition of certain modified characters seems to be possible using handwritten character recognition techniques. However, it is quite challenging to recognize inclined or rotated characters due to the difficulty in estimating their angle of inclination. In Figure 1, we show some inclined and/or rotated characters. Except for the top line(Figure 1(a)), recognition of the other lines is difficult, especially (c) and (d). But this poses no problem to human beings who can read even inverted characters and mirror images. Because they can easily recognize and estimate the alignment of the character by some flexible mechanism. How can we make computers do the same? How can we find the regularity in a character alignment or character orientation without recognition? This is really a paradox. Well then, what can we do?

Thus far, some rotation invariant recognition methods have been proposed. There are three approaches: One is by extracting rotation invariant features [1]. The second is the usage of neural network[2]. The last is the usage of plural templates. Xie et al [3] proposed a rotation invariant system in which the processing was done by establishing plural standard patterns of different angles. However, only 97% of the recognition result was obtained even for ten digit patterns. And some estimation methods of character alignment have been considered using mathematical models[4]. However, character alignments cannot be based on such models all the time. This paper presents a method of recognizing rotated and/or inclined characters based on a parametric eigen-space method[5] which has been used in image understanding.

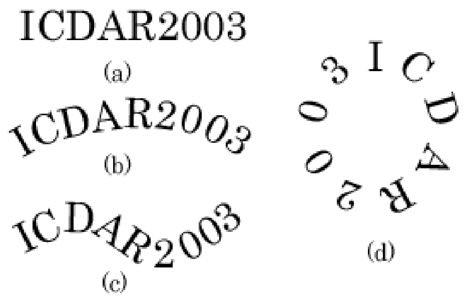


Figure 1 Inclined, rotated characters

We also show that this method has the added advantage of obtaining the recognition result (category) and angle of inclination at the same time.

2. Preparation

By rotating a character 10 degrees at a time, we prepare 36 character images covering all the possible orientations(Figure 2). We use them as learning samples. Let an image pattern be $f^k_{\theta(i)}$, where k is the category number from 1 to C and $\theta(i)$ is a character angle, that is $\theta(i) = 10 \times i \mid i = 0,1,2,\cdots,35$.

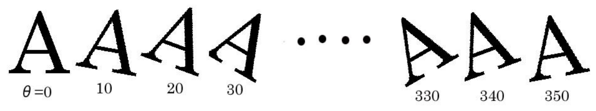


Figure 2 Rotated character images (learning samples)

Each image has a size of 32×32 pixels and all images are normalized. The value of a pixel is 0 or 1. Therefore an image data can be described by a 1024 dimensional vector.

Next, we create an eigen-space using 36 image data with respect to each category. The covariance matrix $\Sigma^{(k)}$ $(=1024\times 1024)$ is calculated as follows;

$$\Sigma^{(k)} = \underset{i}{E}\left[(f^k_{\theta(i)} - m^k)(f^k_{\theta(i)} - m^k)^t\right], \tag{1}$$

where m^k is the mean vector of the k-th category. The covariance matrix can be obtained through eigen expansion:

$$\Sigma^{(k)}\phi = \lambda\phi, \tag{2}$$

where, category index k was omitted for λ and ϕ.
We obtain at most 35 non-zero eigenvalues because the rank of the covariance matrix is at most 35. Let the eigenvectors corresponding to eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{35}$ be $\phi_1, \phi_2, \cdots, \phi_{35}$. Using the first n (≤ 35) eigenvectors, an eigen subspace $U^{(k)}_n = \{\phi_1, \phi_2, \cdots, \phi_n\}$ can be created.
Then, as projected $f^k_{\theta(i)} (i = 0,1,\cdots,35)$ onto the $U^{(k)}_n$, that is, the projected point $F^k_{\theta(i)}$ is $U^{(k)t}_n (f^k_{\theta(i)} - m^k)$, a set of the projected points $\{F^k_{\theta(i)}\}$ draws a locus sequentially because the angle changes consecutively (Figure 3). We denote the locus as $L^{(k)}_n$.

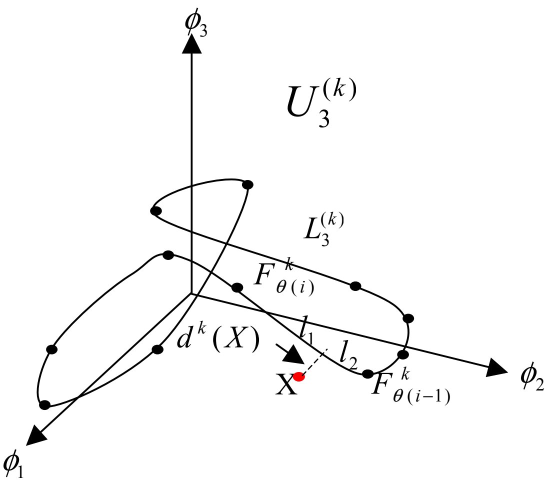


Figure 3 A locus in the eigen subspace of category k

3. Recognition

Given an unknown image data x which is first projected onto all $U^{(k)}_n$ $(k = 1,2,\cdots,C)$. We will denote the projected point of x as X that is, $X = U^{(k)t}_n (x - m^k)$. The verification is carried out by finding the least distance between X and $L^{(k)}_n$ (see Figure 3). We represent the least distance to the category k as $d^k(X)$. Therefore we can obtain the recognition result k^* as follows:

$$k^* = \arg\left[\min_k \{d^k(X)\}\right] \tag{3}$$

On the other hand, the character angle in the unknown image is calculated using two neighboring points on the locus close to X. For example, in the case of Figure 2, the angle θ^k can be interpolated by two points $F^k_{\theta(i-1)}$ and $F^k_{\theta(i)}$, that is :

$$\theta^k = \frac{l_1}{l_1 + l_2}\theta_{(i-1)} + \frac{l_2}{l_1 + l_2}\theta_{(i)}, \tag{4}$$

where l_1 and l_2 are the lengths shown in Figure 2.

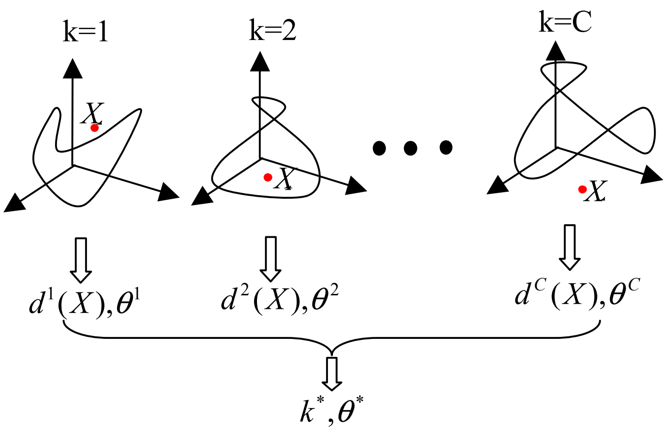


Figure 4 Recognition scheme for rotated characters

In this way, we can obtain the recognition result and the character angle of the input image at the same time. We show the recognition scheme in Figure 4.

4. Experiments

We used CENTURY fonts of 26 capital letters of the English alphabet (A, B, ..., Z) for our experiment. We first made a character pattern with zero degree for each category. Then the patterns were rotated by 10 degrees by a program and 36 images with 32×32 pixels in size were created by re-sampling the circumscribing area of the character image. The feature dimension is 1024. The eigenvalues and eigen-vectors were calculated using Mathemathica[6]. We will show an example of eigenvalues of character “A” in Figure 5. Thirty-five (35) eigen values will be realized. We also show the loci of 36 learning samples projected onto a two dimensional eigen sub-space for all categories in Figure 6. They have quite interesting shapes.

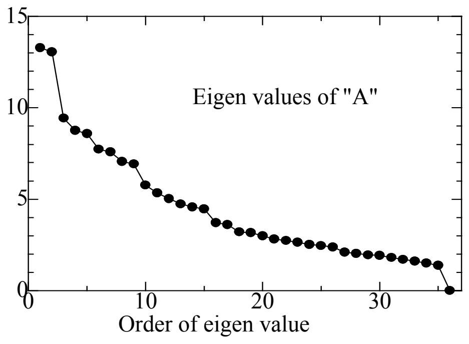
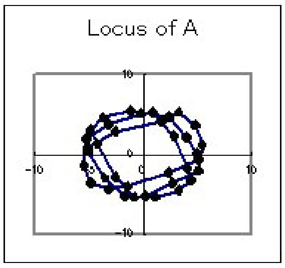
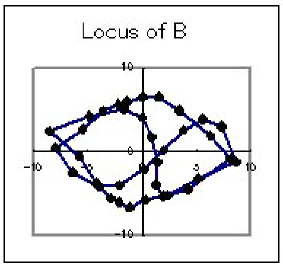
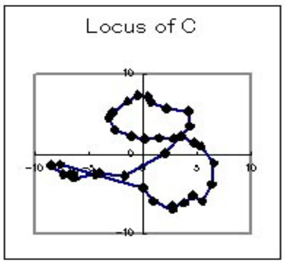
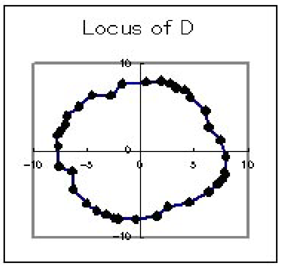


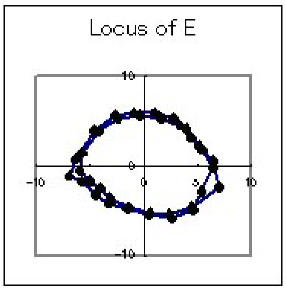
Figure 5 Eigenvalues of “A”

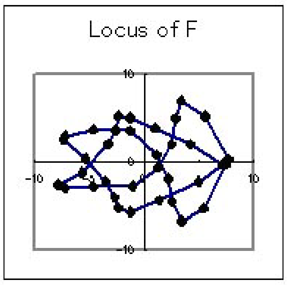


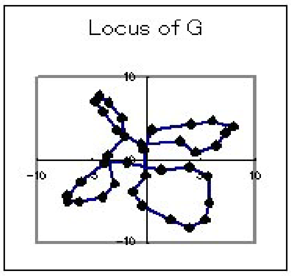


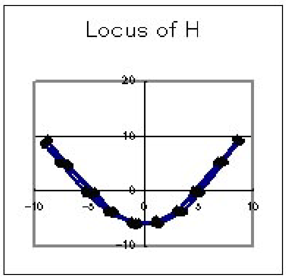


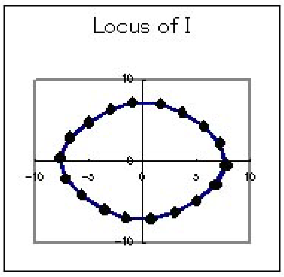


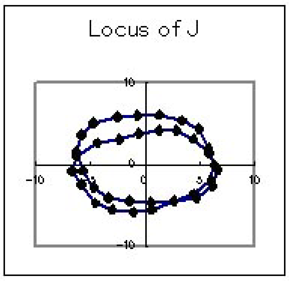


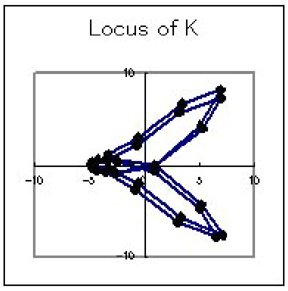


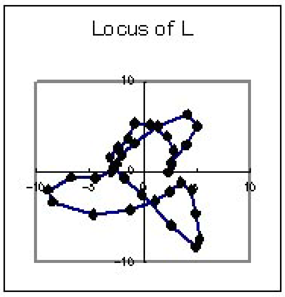


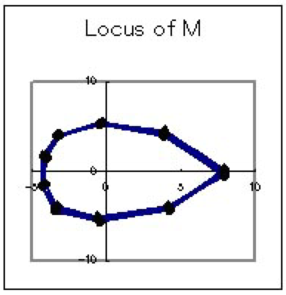


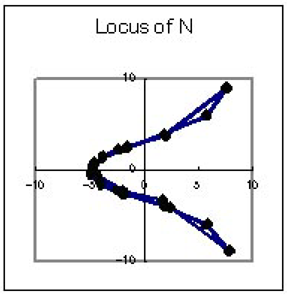


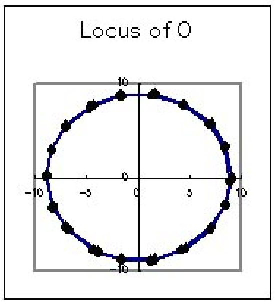


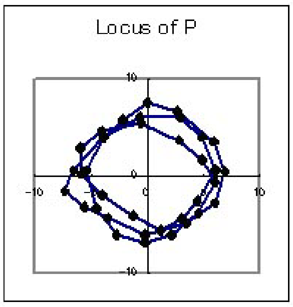












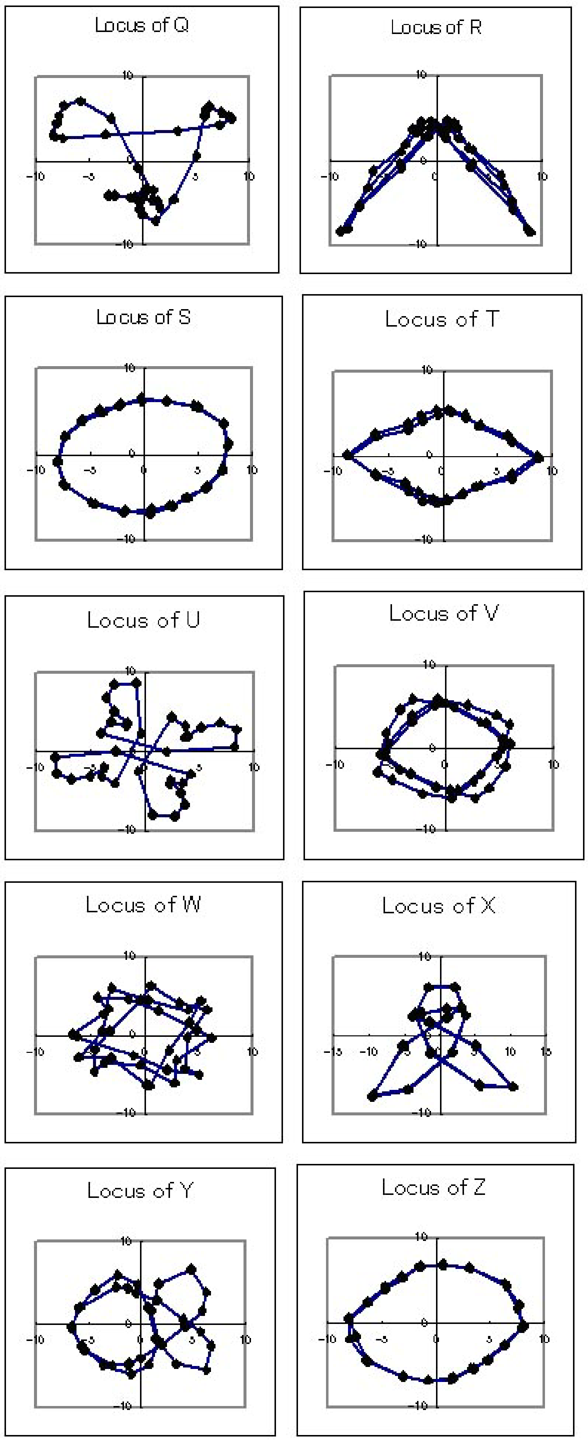


Figure 6 Loci of A to Z

The distance from the projected point X to the locus can be calculated as follows:

1) We interpolated the 36 sample points by the periodic spline so that a locus can be represented by 1000 points. The angle of each point was calculated by Eq.4. An example of interpolated locus is shown in Figure 7.
2) We tabulate the co-ordinates and the angles for 1000 interpolated points. The distance was calculated using this table.

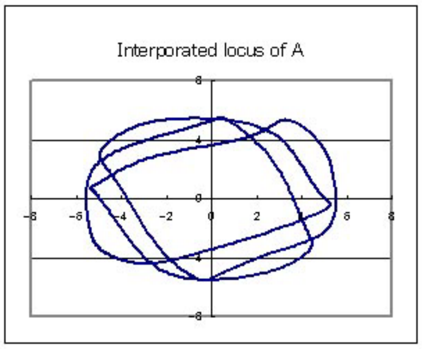


Figure 7 Interpolated locus of "A"
This locus consists of 1000 points

We used test patterns of the same font that were rotated by 3 degrees but do not include learning samples, that is, 3,⋯,357 degrees. There are 108 test samples for each category, totally 2808(=108*26) samples. Figure 8 represents the recognition rate vs. the number of dimensions of the eigen sub-space. One can see that it is already over 90% at only four dimensions. The maximum rate was 99.89% (3 samples failed) at 13 dimensional eigen sub-space. The reason of error is because the number of learning samples is not good enough. Or, rotation by 10 degrees will not be appropriate, that is, rotation by uneven angle or less than 10 degrees may be effective. But the distance between the first candidate and the second one was very small for all three erroneous samples. For example, for the sample 5(N(177)) in Table 1, the distance of the first candidate(error) was 3.422 and the second candidate(correct) was 3.449.

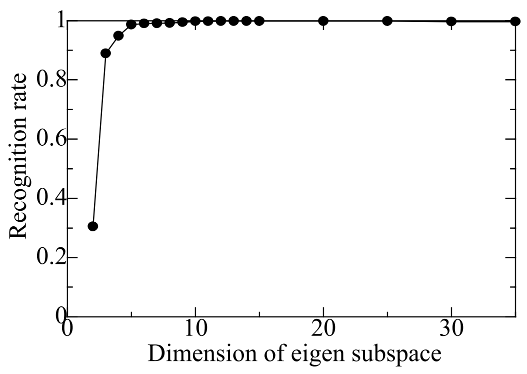


Figure 8 Character recognition result

In our method, we can obtain not only the category but also the angle of inclination of the input character image.

We show the accuracy of angle estimation in Figure 9, which represents the angular difference along the abscissa and the number of samples in the vertical axis. In this figure, the samples recognized upside down are excluded.

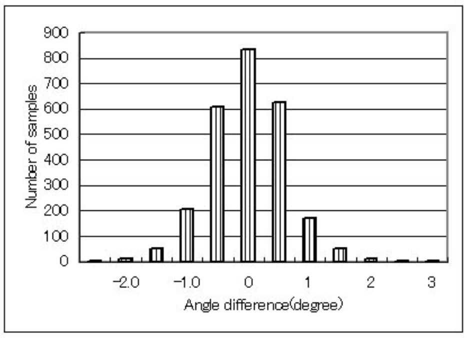


Figure 9 Accuracy of output angles

This graph shows that almost all angles of the test samples were estimated correctly.

Next, we show some examples of the experimental results. Table 1 shows three candidates for 6 input patterns. The first three samples are correctly recognized and next three samples are falsely recognized. The number within parentheses represents the angle. We can see that erroneous samples(4,5,6) contain the correct categories within the top three nominees.

Table 1 Examples of experimental results(13 dimensions)

Input data		1st nominee	2nd nominee	3rd nominee
1	A(45)	A(44.6)	V(225.3)	Y(220.6)
2	N(93)	N(273.7)	I(55.6)	O(99.8)
3	Z(135)	Z(315.2)	I(163.7)	N(29.1)
4	A(339)	V(159.9)	A(339.9)	P(144.1)
5	N(177)	A(159.8)	V(340.5)	N(175.4)
6	V(336)	A(154.9)	V(335.0)	F(318.5)

Lastly, we show the recognition rate for each category in Figure 10. There are some symmetrical patterns in alphabetic letters. For example, "H", "I", "N", "O", "S", "X", "Z" have almost the same patterns even when they are rotated 180 degrees (see samples 2 and 3 in Table 1). Figure 10 shows their recognition results where fine hatch ("correct" in the legend) indicates that the system outputs the correct category and correct angle. Rough hatch("up-side down" in the legend) indicates that it is the correct category but wrong angle(difference of about 180 degrees).
We can see that "H", "I", "O", "X" can be recognized up-side down due to symmetry. In this case, the context of the character string will be helpful.

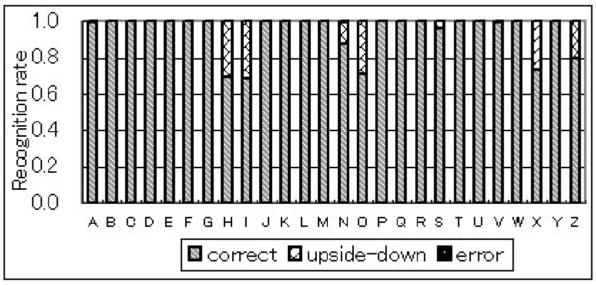


Figure 10 Recognition rate for category

5. Summary

We have presented a recognition method for inclined and/or rotated characters using an eigen sub-space. We used CENTURY fonts of 26 capital letters of the English alphabet (A, B, ..., Z) in this experiment. The results show that this method has a good recognition performance and a precise estimation of character orientation.

The advantage of this method is to not need to estimate the regularity of the character string and to allow us for character-wise recognition. Furthermore, this method has the added advantage of obtaining the recognition result (category) and angle of inclination at the same time.

6. References

[1] S.X.Liao and M.Pawlak, "On Image Analysis by Moments," IEEE Trans. on PAMI, Vol.18, No.3, pp.254-266, (1996).

[2] S.Sato, S.Miyake and H.Aso, "Evaluation of Two Neocognitron-type Models for Recognition of Rotated Patterns," ICONIP 2000, WBP-04,pp295-299 (2000).

[3] Q.Xie, A.Kobayashi, "A Construction of Pattern Recognition System Invariant of Translation, Scale-Change and Rotation Transformation of Patterns(in Japanese)," Trans. of The Society of Instrument and Control Engineers, Vol.27,No.10, pp.1167-1174 (1991).

[4] H.Hase, M. Yoneda, T. Shinokawa, C.Y.Suen "Alignment of Free Layout Color Texts for Character Recognition," Proceedings of the 6th International Conference on Document Analysis and Recognition, pp.932-936 (Seattle, USA) (2001).

[5] H.Murase, S. K. Nayar, "3D Object Recognition from Appearance—Parametric Eigenspace Method—(in Japanese)," Trans. of IEICE, Vol.J77-D-II,No.11, pp.2179-2187 (1994).

[6] Stephen Wolfram, "Mathematica Book," Wolfram Research, Inc.Vol.4 (2000).