Multi-font Rotated Character Recognition using Periodicity

Multi-font Rotated Character Recognition using Periodicity

Hiroyuki Hase, Kohei Tanabe, Thi Hong Ha Tran, Shogo Tokai
Graduate School of Engineering, University of Fukui
{hase,tokai}@fuis.fuis.fukui-u.ac.jp, honghahh81@yahoo.co.jp

Abstract

This paper presents on accuracy improvement of multi-font rotated character recognition. Until now, a recognition method for rotated characters was based on distance criterion on the eigen sub-space. That is, an unknown pattern is projected onto the eigen-subspace of each category. The category which shows the closest distance between the projected point and the category locus is chosen. However, this simple method could not be cope with multi-font characters. Therefore, some unknown patterns were created by rotating the input pattern and projected onto the eigen-subspace of each category. By that method, a good performance was achieved for small size of categories like alphabetic 26 capital letters. However, the performance fell down by increasing the number of categories like 62 alpha-numeric letters. By considering the cause of the misclassification, we found that the distance criterion accidentally caused misclassification. This paper proposes a new feature based on periodic property of projected points on the eigen space. The experimental results showed a considerably high recognition rate.

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 rotated characters due to the difficulty in estimating their angle of inclination. In Fig.1, some inclined and/or rotated characters are shown. Except for the top line(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 thing? How can we find the regularity in a character alignment or character orientation without recognition? This is a paradox.

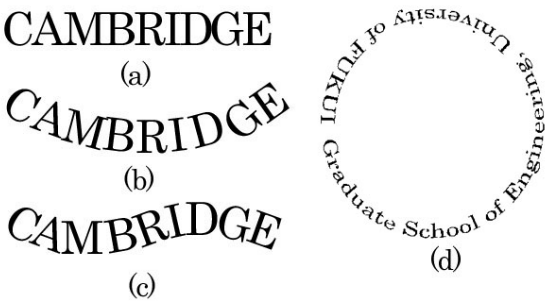


Figure 1 Distorted character lines

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 carried out by establishing plural standard patterns with different inclinations. However, only 97% of the recognition result was obtained even for ten printed digit patterns. Also 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.

While, the parametric eigen space method which was proposed by Murase in 1994 is used for 3D object recognition[5]. However, there are very few reports using this method for character recognition.

In our previous paper [6], we presented a recognition method for rotated characters. This method is based on the parametric eigen-space method.

978-0-7695-3337-7/08 $25.00 © 2008 IEEE
DOI 10.1109/DAS.2008.16

In our experiment for a single font, nearly 100% accuracy was achieved in the case using a distance between the input rotated character and the manifold(we say "locus" hereafter) in the eigen space which was created from the sample characters with the same font. However, when this method was applied to characters with other font, the recognition rate quite fell down. Therefore, in our improved method, the average character image using several fonts was used in learning process. And also, taking advantage of the characteristic of character, plural character patterns that were rotated by the fixed angle were projected onto the eigen space in the recognition process, then plural distances between the projected points of input characters and the locus were averaged and used for recognition. The reason that two or more patterns are created by rotating the input character is that the projection by a single character may cause misclassification as a closer distance coincidentally occurs in other categories. As the result, the accuracy over 99.5% and within ±2 degrees could be obtained for 26 categories. However, when this method was applied for 62 categories which are Alphabetic capital and small letters and numeric characters, the accuracy fell down about 10-20%.

This paper proposes a new approach to recognize for 62 categories by devising feature extraction. In section 2, the previous works are explained. By considering recognition error in section 3, a new method is proposed and good experimental results are shown.

2. Previous work

2.1 Learning

By rotating a character by 10 degrees at a time, we prepare 36 character images covering all the possible orientations (Fig. 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 $\theta(i) = 10 \times i \mid i = 0,1,2,\cdots,35$.

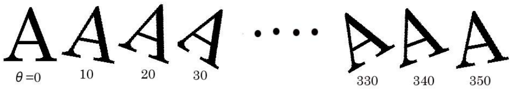


Figure 2 Rotated character images (learning samples) :Character images are rotated by 10 degrees.

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 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 $\Sigma^{(k)}$ $(=1024\times1024)$ is calculated as follows;

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

where m^k is the mean vector of the k-th category, and E is the expectation operator. The eigen vectors can be obtained through eigen expansion:

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

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 because the angle changes consecutively. We denote the locus as $L^{(k)}_n$.

2.2 Recognition

Given an unknown image x, it is projected onto all $U^{(k)}_n$ $(k = 1,2,\cdots,C)$. We will denote the projected point of x as X, i.e. $X = U^{(k)^t}_n (x - m^k)$. Verification is carried out by finding the shortest distance between X and $L^{(k)}_n$. We represent the shortest distance to the category k as $d^k(X)$. Therefore we can obtain the recognition result k^* as follows:

$$k^* = \arg\min_k \{d^k(X)\} \quad (3)$$

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

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

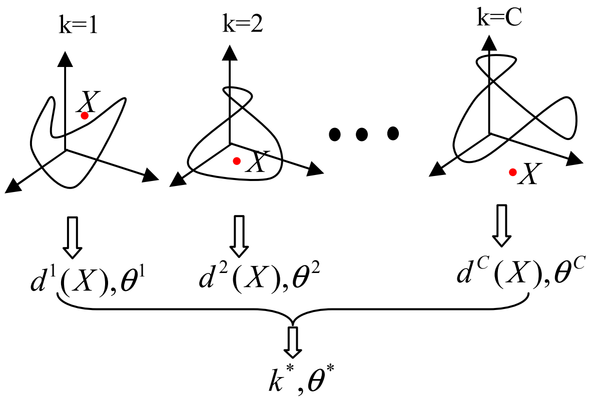


Figure 3 Recognition scheme for a rotated character: Learning samples shows an unique locus on the eigen space. X in the figure represents the projected point of input character image.

where l_1 and l_2 are the lengths from the neighboring two points. 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 Fig. 3.

2.3. Experiments and results

We used 26 capital letters of Century font (A, B, ..., Z). The feature dimensionality is 1024. The eigenvalues and eigen-vectors were calculated.

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).

2) We tabulate the co-ordinates and the angles for 1000 interpolated points. The distance was calculated using this table.

We used test patterns of the same font that were rotated by 3 degrees but do not include learning samples, that is, $3,\cdots,357$ degrees. There are 108 test samples for each category, totally 2808(=108*26) samples.

The experimental result showed a very high recognition rate of 99.89% (3 samples failed) at 13 dimensional eigen sub-space. We also obtained accurate angle estimation within one degree except some symmetrical categories like "H", "I", "N", "O", "S", "X", "Z". They have almost the same patterns when they are rotated 180 degrees. Also, the estimated angle of test patterns were within ± 1 degree from the true angle.

Furthermore, we tested for other fonts using the same loci. However, the recognition rates were degraded to 93.95% for Times New Roman and 84.33% for Courier font(see Fig.4). So, we improved this drawback of our method. In the following subsection, we present an improved recognition method for multi-font rotated characters.

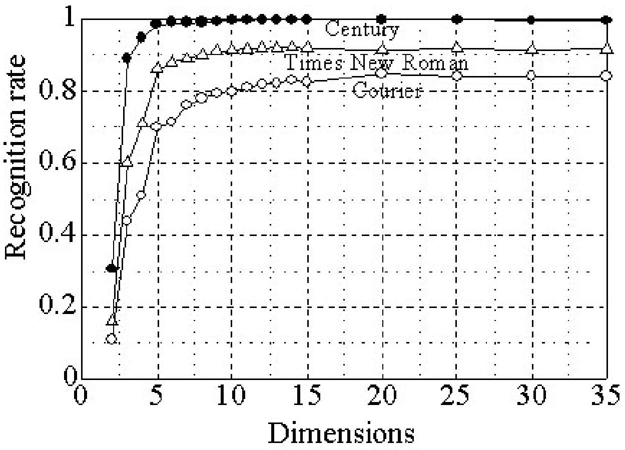


Figure 4 Experimental results using the loci created for the Century font

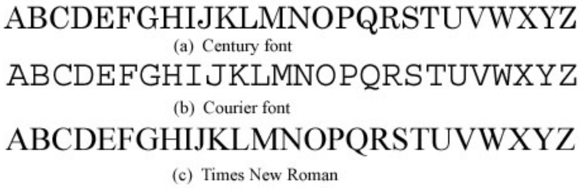
2.4. Multi-font recognition

To cope with multi-font recognition, we can think of (1)preparing standard patterns corresponding to each font, (2)utilizing statistical methods like averaging and variance, and (3)developing a new special method.

We used two techniques including (2) and (3); one is in a learning process and other is in a recognition process.

2.4.1 Learning

We used three popular fonts in the learning process, that is, Century, Courier, Times New Roman fonts (Fig. 5).


ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

(a) Century font

(b) Courier font

(c) Times New Roman

Figure 5 Character shapes of three fonts

The contrivance in the learning process is to average these character images. Three character images with the same angle and same category are re-sampled by 32*32 pixels, then they were averaged. We constructed the eigen spaces from them.

2.4.2 Recognition

We adopted an improved method in the recognition process. After inputting an unknown character, it is rotated by a fixed angle to create some rotated images. The number of the images created by rotation is denoted by *R*. These images are re-sampled by 32x32 to produce binary images. When these binary images are projected onto the eigen space, we obtain *R* distances from the locus. For example, Fig. 6 illustrates the case of *R*=3.

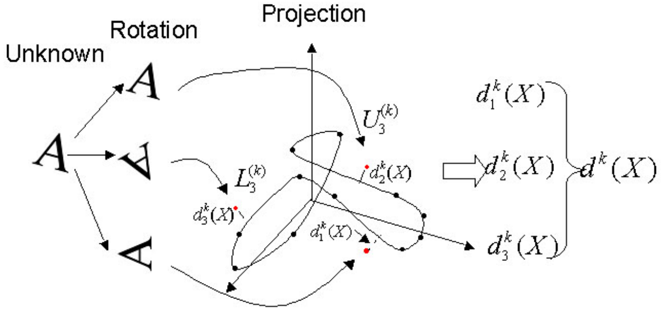


Figure 6 Illustration of the recognition process. Plural images of input image are created by rotation and projected on the eigen space.

In this recognition process, the category with the minimum averaged distance is chosen. The character angle can be estimated from the projected point of the original input image.

2.4.3 Experimental results

As well as the previous experiment, we used test samples that were rotated by 3 degrees, i.e. $3,\cdots,357$. There are 108 test samples for each category, totally 2808(=108*26) samples for a font. Learning samples were excluded.

Figs. 7, 8 and 9 show the experimental results for Century font, Courier font and Times New Roman font respectively. The parameter *R* is the number of rotations. There are five graphs in each figure to show the results when *R* varies from *R*=1 to *R*=5. The graph of *R*=1 represents the result for the original input pattern and it also means the effect of the averaging only. *R*=2 means the one for two created patterns rotated by 180 degrees. *R*=3 is for three patterns by 120 degrees. *R*=4 is for four patterns by 90 degrees. *R*=5 is for five patterns by 72 degrees. Note that the results from *R*=2 to *R*=5 means the synergistic effect of averaging and rotation.

In Fig. 7 for Century font, we can see 100% accuracy at *dim*=3,4,5 for *R*=5. In Fig.8 of Courier font, we obtained 100% accuracy for over 13 dimensions for all *R*'s.

In Fig. 9 for Times New Roman font, we obtained 99.86% at three dimensions for *R*=5. Thus, we can see that very high accuracy can be obtained at low dimension for *R*=5. Totally, we obtained the recognition rate of 99.8% at 4 dimensions when *R*=5.

In every fonts, *R*=5 produced the highest rate of recognition at the low dimension from two to five, and *R*=3, *R*=4, *R*=2 in order, and *R*=1 yielded the lowest accuracy.

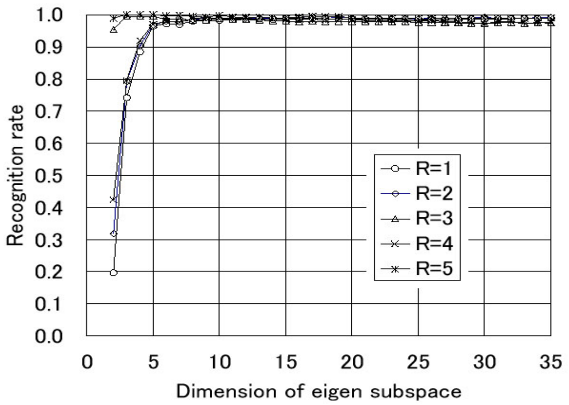


Figure 7 Recognition rate for Century font.
R **is the number of images created by rotation from the input image**

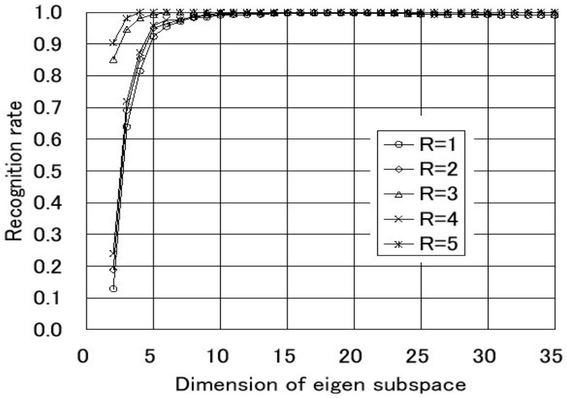


Figure 8 Recognition rate for Courier font.

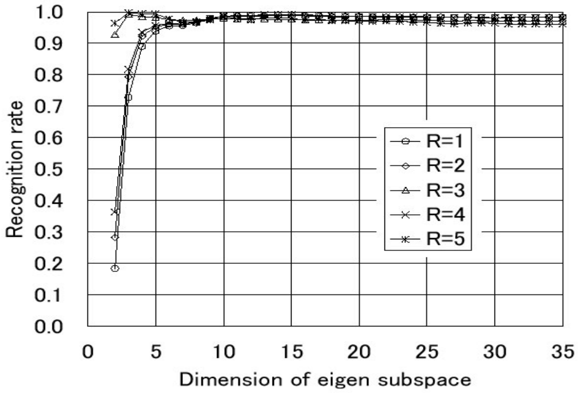


Figure 9 Recognition rate for Times New Roman font.

From the fact that the accuracy for *R*=5,3 is better than *R*=4,2, we can conjecture that the projection points around 90, 180, 270 degrees in the subspaces of some categories are close.

Next, we consider the accuracy of angle estimation. Fig. 10 shows a histogram 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. This graph shows that the estimated angles of the test samples were estimated within two degrees from the true angle. This is a little less accurate than the previous experimental result.

Fig. 11 shows the accuracy for each alphabetic letter where fine hatch ("correct" in the legend) indicates that the system outputs the correct category and correct angle. Rough hatch("upside-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 upside-down due to symmetry. In this case, the context of the character string will be helpful.

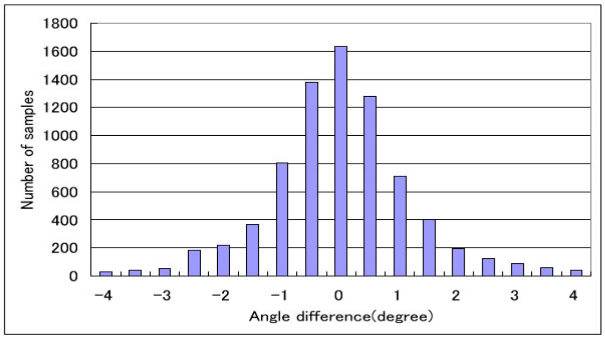


Figure 10 Accuracy of angle estimation: The samples recognized upside down are excluded.

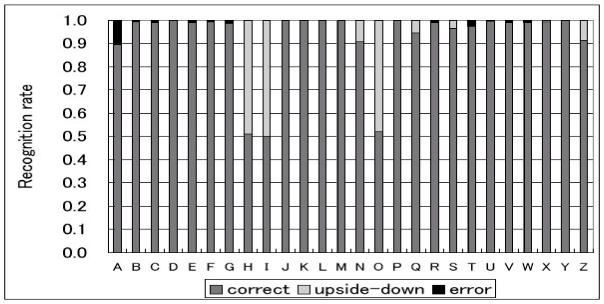


Figure 11 Category-wise recognition rate: Fine hatch ("correct" in the legend) indicates that the system outputs the correct category and correct angle. Rough hatch("upside-down" in the legend) indicates that it is the correct category but wrong angle(difference of about 180 degrees.

3. Recognition for 62 categories

3.1 Test

We experimented for 26 categories of English capital letters in the previous work. In this subsection, this method was applied for 62 categories, i.e. English capital and small letters and 10 numerals. The fonts we used are four fonts which are Century, Courier, Times New Roman and Arial font. First, character images of four fonts with the same angle of the same category are averaged. By rotating by 10 degrees, 36 averaged character images were created. After normalizing the image, 36 images with 32×32 pixel and 5 level intensity are obtained, then transformed to a 1024 dimensional vector. We used the same method as well as in the learning and recognition processes in section 2. The rotation parameter *R* of the input character is from 1 to 5. As the results, the result at *R*=3 showed the best accuracy, we showed it in Fig.12.

As shown in the figure, we can see that the recognition rate is quite low comparing with 26 categories shown in Figs. 7-9. This is because there are some special categories with similar shape like "C" and "c", "O" and "o", "S" and "s" etc. And smaller distance between the locus and the projected point incidentally occurs in other category. As an example of the latter, a "j" was misclassified as the category "Q" which has quite different shape. For example, the error rate for *R*=3 of Arial font is 21%, 55% of them is misclassification into a similar category, and 45% is into non-similar category. In addition to "C" and "c", "S" and "s", similar categories include "n" and "u", "b" and "q", "d" and "p" which have the same shape when they are rotated 180 degrees. Misclassification was sometimes occurs between non-similar categories like "Q" and "j", "g" and "S". It is thought that misclassification into a similar category is unavoidably, but we have to avoid misclassification into quite different category. So, we considered the loci of misclassification categories with different shape. Fig.13 show the locus of "j" and Arial font "Q" on three dimensional eigen space which was constructed by averaged characters of "j". In the figure, the symbol ○ indicates the projected point of average character "j", the symbol "+" is the projected point of Arial font "Q". You can see that they have different shapes but some points of the both loci are very close each other.

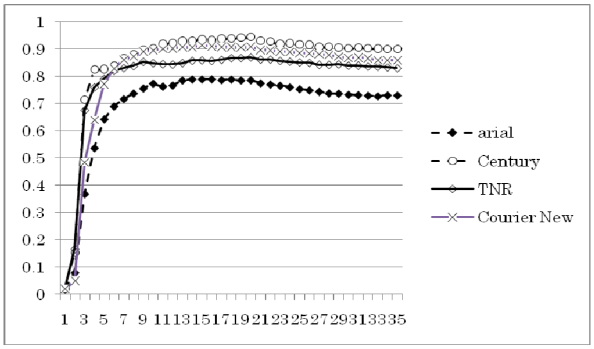


Figure 12 Experimental results for 62 categories(R=3): Totally, 26,784 samples (108 × 62 categories×4 font) were used for the experiments.

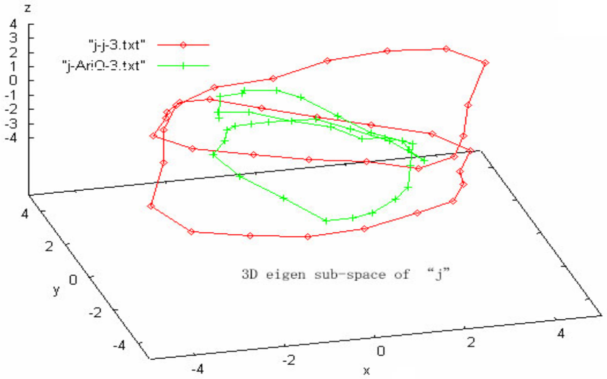


Figure 13 Projections of "j" and "Q" in the three dimensional eigen space: the outer locus indicates the projection of average character "j", the inner locus is the one of Arial font "Q".

To avoid misclassification by contingency using the distance criterion, we consider the variation in respect to character shape.

Fig.14 is a graph of the projected points of average character "D" and Arial font "D" on the first eigen vector which was calculated from the average character "D". Similarly, Fig.15 is a graph for the average character "Q" and Arial font "Q" projected on the first eigen vector. Fig.16 shows projected points of "Q" on the second eigen vector. As shown, the same kind of character shows the similar locus but has some variations.

While, the variation also comes from character stroke width. That is, thicker stroke makes the norm of the image bigger, then the amplitude (projected coordinate) becomes bigger. Fig.17 shows the loci on 2D eigen space for different stroke width. Furthermore, different font causes different variation depending on the angle. However, similar character shape gives similar periodic function.

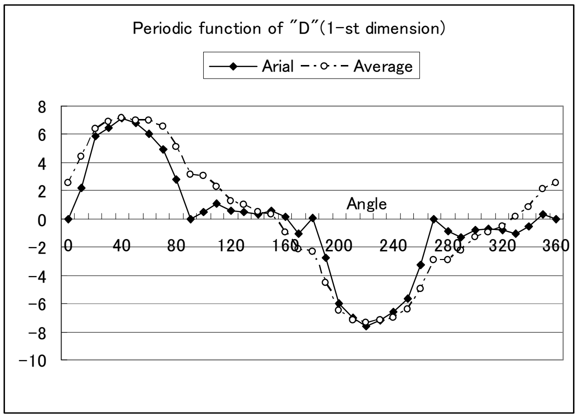


Figure 14 Behavior of "D" with Arial font on the 1-st eign vector:

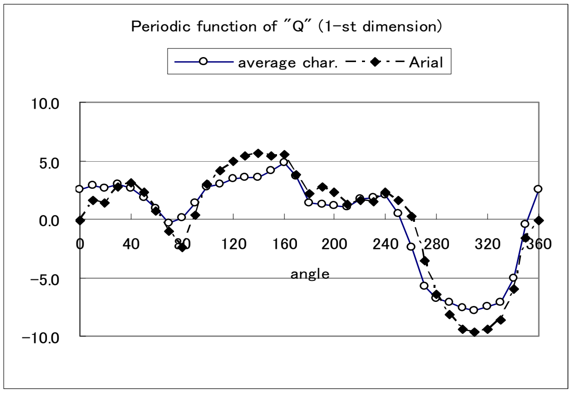


Figure 15 Behavior of "Q" on the 1-st eign vector: Projected Points of the average character "Q" (circle) and Arial font "Q"(dot) projected on the first eigen vector

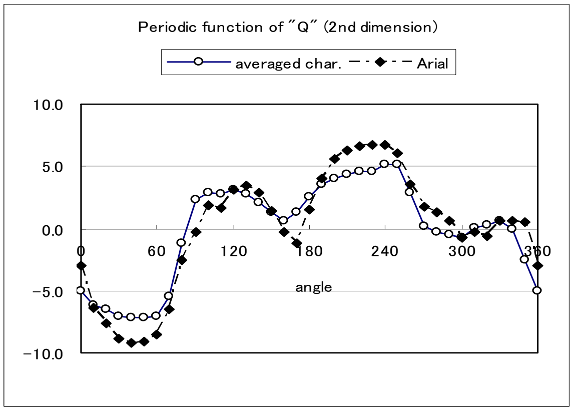
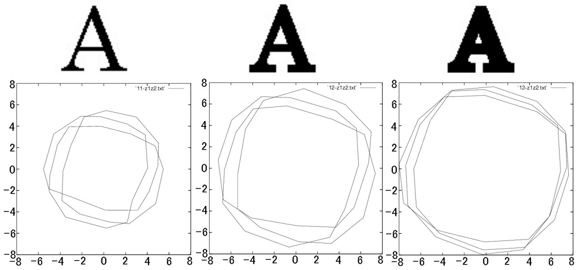


Figure 16 Behavior of "Q" on the 2-nd eign vector: Projected Points of the average character "Q" (circle) and Arial font "Q"(dot) projected on the second eigen vector



(a)original size (b) thick width (c) thicker width

Figure 17 Size of locus for character stroke width: These are shown the loci on two dimensional eigen space.

Therefore, it seems that misclassification reduces if such variations can be absorbed.

Thus, some kinds of variations are seen in the loci of all categories. To absorb these variations, some techniques have been proposed so far. One method is to define an acceptable region along the locus or to define stochastically variation region centering at the locus [7] . However, the variation region is so different from category and angle that we focused on periodicity of the locus.

3.2 Recognition using zero-cross points

According to the consideration on misclassification, we found that characters of the same category and different font draw a similar locus on the eigen space. Therefore, we checked how much variation the projected point of the input character has around the locus.

Until now, an acceptable region as mentioned before has been proposed. This kind of idea contains a problem, that is, the setting of the acceptable region is quite complicate because its width depends on the angle.

Fig.18 shows the projected points of Arial font “A” on the 1-st, 5-th, 10-th eigen vectors. The horizontal axis is the angle of the input character, the vertical axis is the projected coordinate.

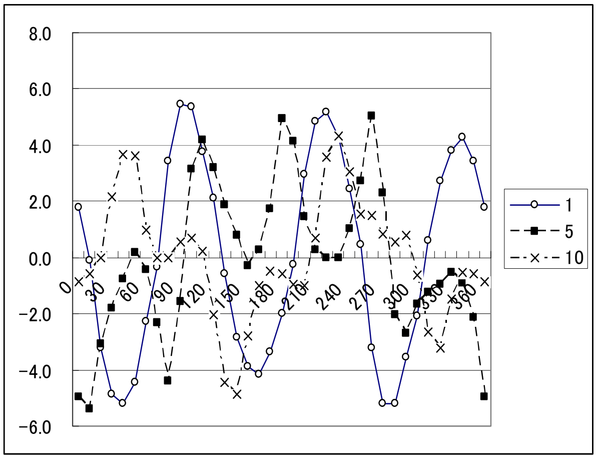


Figure 18 Functions of projected points for Arial font “A”(1, 5, 10 dimensions): these are shown the projected points on each eigen vector. The horizontal axis is the angle.

As you can see, a simple sinusoidal wave can be seen at low dimension. In higher dimension, higher frequency and some distorted wave tend to appear. This is commonly seen in any categories but the shape of the distortion and frequency depend on category.

According to this consideration, we adopt the number of zero-cross points as a feature to absorb the variation because the number of zero-cross points is generally stable for character variation. The number of zero-cross points can be obtained by rotating the image 360 degrees.

Let the number of zero-cross points on the *j-th* eigen vector be $\bar{z}_j^{k}$, where k is the category number. And let the number of zero-cross points of the input character on the *j-th* eigen vector be z_j. Then, the disparity η using d dimensions is defined as Eq.(6)

$$\eta^k(d)=\sum_{j=1}^{d}|z_j-\bar{z}_j^{k}| \qquad (6)$$

The output category k^* is determined by Eq.(7).

$$k^*=\arg\min_k[\eta^k(d)] \qquad (7)$$

The experimental result is shown in Fig.19. In this experiment, the capital and small letters with similar shape are assumed to be different category. In this figure, we can see a different tendency of recognition rate from Fig.12. This means that there is little difference in the number of zero-cross points at low dimension among all categories, the difference can be seen at higher dimension. The character information that each eigen vector has can be seen by reconstraction from the projected points. That is, a character image g_θ^k used n dimensions can be reconstructed using the projected point F_θ^k of an angle θ.

$$g_\theta^k(n)=U_n^{(k)}F_\theta^k+m^k \qquad (8)$$

Examples of reconstructed character are shown in Fig.20. In this figure, character images are binarized.

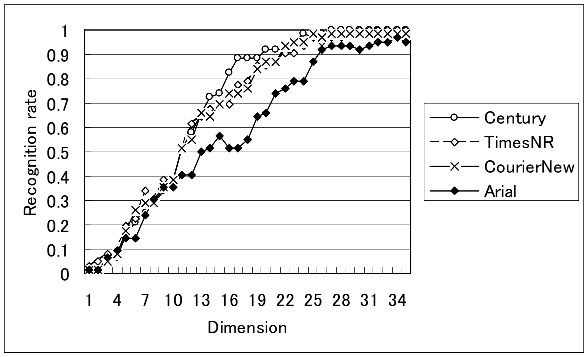
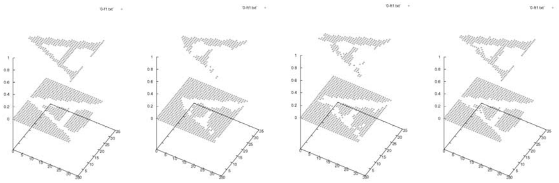


Figure 19 Recognition results for four fonts using periodicity:



(a)Original (b)10 dim. (c)15 dim. (d)25 dim.

Figure 20 Restoration image: The images are binarized by a threshold. Character “A” is not reconstructed perfectly even when 25 dimensions are used.

In Fig.20, (a) is the original binary image, (b) is the reconstructed image using 10 dimensions, (c) and (d) are for 15 dimensitons and 25 dimensions respectively. You can see that the character image “A” is not reconstructed perfectly even when 25 dimensions are used.

We obtained a better recognition rate over 95% for 62 categories as shown in Fig.19. In this research, we used only the number of zero-cross points but did not use the shape or phase of loci so that we could not obtain high recognition rate at low dimension.

Fig.21 shows the anglar difference between the estimated angle and the true one. The condition of the input characters is the same as the case of Fig.10. Comparing with Fig.10, the preciseness is a little broad.

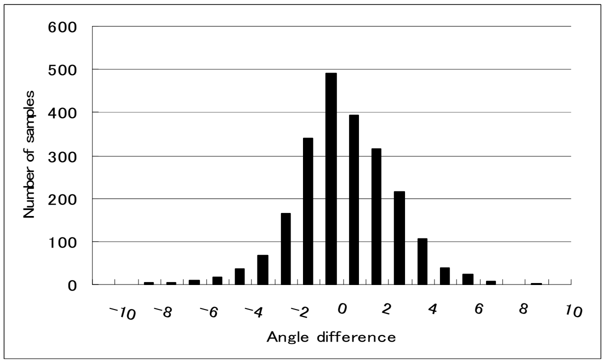


Figure 21 Precision of the estmated angle: The samples recognized upside down and misclassified are excluded.

4. Summary

In this paper, we improved the accuracy of multi-font rotated character recognition using the parametric eigen space method. A conventional distance criterion shows high accuracy over 99% for 26 categories. However, when this method was applied for 62 categories of English capital and small letters and numeric characters, the accuracy fell down about 10-20%. The reason is that some categories with similar shape exist and a smaller distance between the projected point and the locus happen accidentally in other categories with quite different shape.

So, this paper proposed a new method by devising feature extraction. That is, we used the number of zero-cross points. As the result, a better accuracy could be obtained. This is because we focused on the periodicity of locus. However, we can not obtain such a good accuracy without all dimensions so that we are not satisfied on this point because we used the eigen space method aiming at dimension reduction. We want to consider this problem as future work.

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] H. Hase, T. Shinokawa, M. Yoneda, C.Y. Suen, “Recognition of Rotated Characters by Eigen Space,” ICDAR2003, PII-2,p.731-735 (2003).

[7] Lina , T. Takahashi, I. Ide, H. Murase, “Appearance Manifold with Embedded Covariance Matrix for Robust 3D Object Recognition”, MVA2007 IAPR Conf. on Machine Vision Application, pp.504-507, 2007.