



ELSEVIER

Pattern Recognition Letters 22 (2001) 85–95

Pattern Recognition  
Letters

www.elsevier.nl/locate/patrec

# A rotation invariant printed Chinese character recognition system

Tai-Ning Yang, Sheng-De Wang \*

*Department of Electrical Engineering, National Taiwan University, EE Building, Taipei 106, Taiwan*

Received 31 December 1998; received in revised form 30 August 2000

## Abstract

In this paper, we present an approach to construct an optical character recognition (OCR) system for recognizing rotated printed Chinese characters. Our system includes preprocessing, feature extraction, candidates selection, and target character recognition. The proposed system has a three-stage structure designed mainly to reduce the time complexity in the recognition process. Our goal is to recognize the complete set of frequently used 13 053 printed Chinese characters with arbitrary orientations. The overall recognition rate of our system reaches 97.4%. © 2001 Elsevier Science B.V. All rights reserved.

**Keywords:** Printed Chinese character recognition; Rotation invariant; Clustering

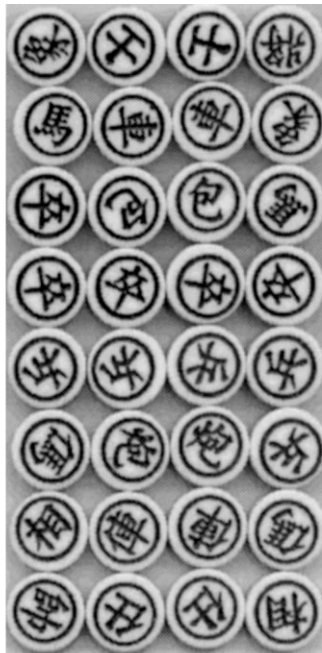
## 1. Introduction

In comparison with the one-to-one mapping of English characters in the keyboard, inputting a Chinese character needs four to five keystrokes. For the past decade, great efforts have been made to develop an efficient computer input device for Chinese characters. An intelligent recognition device seems an ideal solution to this problem. Thanks to the development in this field, for now there are many successful commercial Chinese character recognition systems including on-line and off-line versions in the market. An off-line Chinese recognition system uses an optical scanner as the input device and thus it is called optical

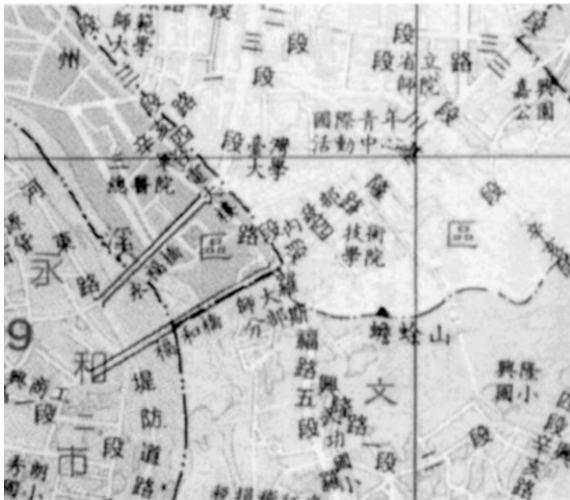
character recognition (OCR), while on-line Chinese recognition system uses a digital pen as the on-line input device. Although researchers have made great progress in this field, there are still some challenges. For instance, a good segmentation algorithm for off-line handwritten Chinese characters is still under development. The stroke-order independent on-line system is also hard to design. Besides that, various special applications including rotated character recognition are not well developed. There are many occasions that require recognizing rotated printed Chinese characters. Two examples are shown in Fig. 1. In this paper, we concentrate in designing a recognition system, which can recognize off-line rotated printed Chinese characters. In evaluating the performance of recognition systems, the recognition rate and processing speed are two important criteria. Although various features and architectures are proposed for recognizing optical Chinese

\* Corresponding author. Tel.: +886-22-3635251; fax: +886-22-3671909.

E-mail address: sdwang@cc.ee.ntu.edu.tw (S.-D. Wang).



(a)



(b)

Fig. 1. (a) Chinese chess. (b) City map of Taipei near our university.

characters, few papers are about rotated Chinese character recognition. We shall introduce the general structure of an OCR system in Section 2. In Section 3, we propose a rotation invariant feature for our system and present the corresponding extraction algorithm. In Section 4, the proposed

feature is compared to the ring projection feature and we modify the proposed feature with a histogram equalization method. Our goal is to recognize not only the 5401 daily used Chinese characters but also the more complete 13 053 Chinese characters. Thus, an appropriate architecture capable of reducing the recognition time is preferred. For achieving this purpose, we use a three-stage recognition system. The first and second stages are overlapped clustering networks that assign the input character to the appropriate cluster. The final stage is a nearest-neighbor classifier. We describe the training process in Section 5. Section 6 presents the simulation results and Section 7 contains the conclusions.

## 2. General structure of an optical Chinese character recognition system

An OCR system is composed of several parts as shown in Fig. 2. At first, the document is scanned and digitized. The noise is removed and the text is separated from the pictures. In the segmentation stage, the character string is divided such that each image contains a single character (Lu, 1995). Although any polygon could be used for the boundary of a segmented character, the rectangle is the most popular because it is easier to store the image as an array in the computer. The individual character image is sent to the preprocessing stage. Many preprocessing techniques could be used in the preprocessing stage according to the image quality and the feature used in this system. The character image is usually normalized to a fixed size with a linear or nonlinear normalization method (Yamada et al., 1990). If the original document is printed on noisy papers such as newspaper, some specially designed noise removal algorithm may be used. Several smoothing algorithms could be used to reduce the zigzag along the stroke edge. The thinning algorithm (Suen and Zhang, 1984) is a popular procedure to reduce the stroke width by discarding the boundary pixels. On the contrary, the contour algorithm (Cheng and Yan, 1998) conserves the boundary pixels and abandons others. Although preprocessing helps the later recognition steps, some information may

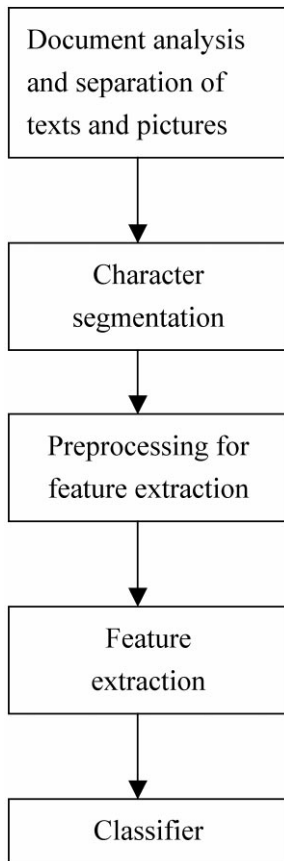


Fig. 2. Flow chart of a general optical character recognition system.

be lost and some distortions may be caused after preprocessing. For instance, the information about stroke width disappears after the thinning process. The nonunique intersection points problem and the multi-fork point splitting problem are usually brought about after thinning. A careful selection and combination of the preprocessing techniques is necessary.

The feature extraction step and the classifier are the two most important parts in the recognition system (Trier et al., 1996). In constructing a recognition system, one of the ideal ways is to simulate the structure of the human brain and build an architecture that could extract the feature and classify it.

This concept leads to the development of several rotation invariant neural networks. Some

papers discuss the way of extracting high-order rotation invariant features with a feature extraction net (Perantonis and Lisboa, 1992). Some neural nets extract partially rotation invariant features and compensate with a partially rotation invariant classification net (Fukumi et al., 1992). In traditional statistical pattern recognition, researchers prefer separating the processes of feature extraction and the classifier. Thus, various classifiers could be compared based on their recognition rates and processing speeds with the same feature set. This kind of comparison is a popular way to credit a proposed classifier. As the feature extraction procedure plays a decisive role in the recognition rate, looking for a good feature is usually challenging. Unlike the classifier that may be applied to different feature sets for constructing various recognition systems, in most cases, a suitable feature set should be designed for each target recognition system.

Fig. 3 is the flow diagram of our system. The dashed line represents the learning process and the solid line represents the testing process. The proposed system includes four parts: preprocessing, rotation invariant feature extraction, candidates selection, and target character recognition. The font database comes from the Eten Chinese operating system that is popular in Taiwan. There are six kinds of fonts in this database. Each font set contains 13053 Chinese character images. The training samples are original character images. The testing samples are images that are randomly rotated and scaled up.

### 3. Preprocessing and rotation invariant feature extraction

Since the resolution is limited in the digital image domain, the rotation operation on a digital image is not invertible. It is not a one-to-one mapping. Some undesirable empty slots called “measles” may appear after the rotation operation (Cheng and Yan, 1998). When we discuss whether a function is rotation invariant, it is better to discuss in the continuous image domain. Let us denote a segmented continuous character image with  $c_{pix}(i, j)$ , where  $i \in [0, W]$  and  $j \in [0, H]$ . In the

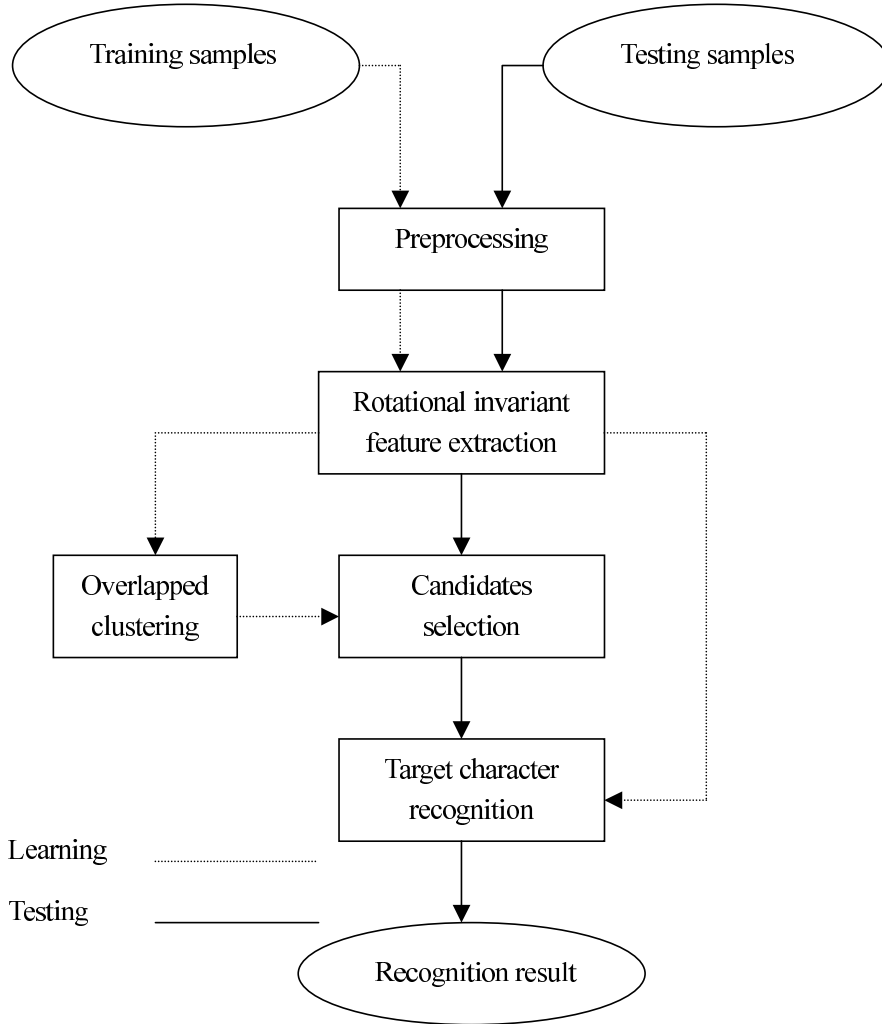


Fig. 3. Flow chart of the proposed optical character recognition system.

following discussions, we use “point” to represent the point in a continuous image and use “pixel” for the pixel in the digital image. In the segmentation process,  $cpix(i, j)$  is produced by determining the four outermost points, left-most, right-most, up-most and down-most points (Lu, 1995).  $W$  and  $H$  represent the width and height of the image, respectively. If there is a foreground point at coordinates  $(i, j)$  then  $cpix(i, j) = 1$ , else  $cpix(i, j) = 0$ .

The following is the traditional way to normalize the original image to be the image with the

height  $H'$  and the width  $W'$ . The normalized image  $ncpix(p, q)$  can be determined by  $ncpix(p, q) = cpix(p * W/W', q * H/H')$ , where  $p \in [0, W']$  and  $q \in [0, H']$ .

Fig. 4 shows this kind of traditional normalization is not rotation invariant. The reason is obvious since the width and height are different between the segmented character image and the segmented rotated character image. We aim to design a normalization approach that is rotation invariant. Our approach finds the image mean and translates the image such that the mean becomes the origin.

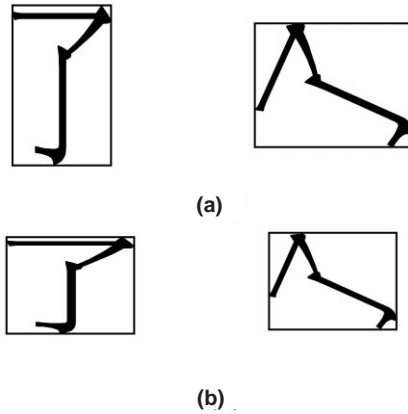


Fig. 4. (a) Two segmented character images. (b) The normalized images.

The image mean is defined to be the mean of the vector set  $\{(i, j) \mid cpix(i, j) = 1\}$ . We use Fig. 5(a) and (b) to show the estimation of mean and the translated image. The circle represents the range of the foreground points. The image is translated such that the origin is at the mean. The outer-most foreground point in this image is the foreground point with the longest distance to the mean. This form of outer-most point is rotation invariant since the outer-most point is still outer-most after

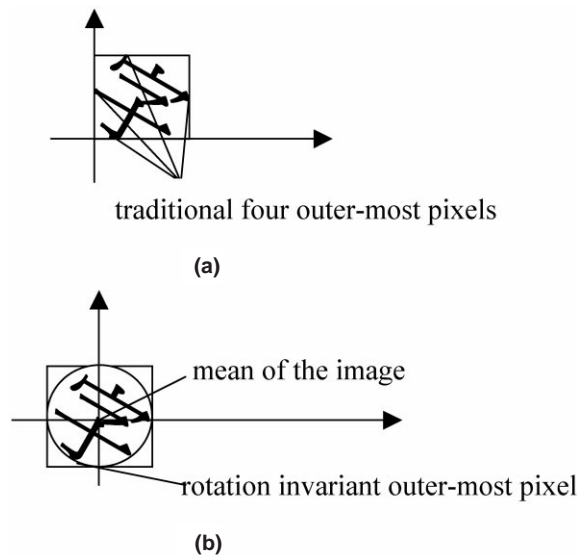


Fig. 5. (a) Original image. (b) Translated image.

rotation operations. We implement a linear normalization such that the longest Euclidean distance to the mean is a fixed value,  $\eta$ . Since the most popular font resolution used is about  $24 \times 24$  and the distance between the center and the farthest corner point is about  $\sqrt{12^2 + 12^2}$ . We set  $\eta = 17$  in all the experiments. Let  $(s, t)$  be the coordinates of the outer-most point. Following is the transformation rule:

$$ncpix(p, q) = \begin{cases} 1 & \text{if } cpix(p * \sqrt{s^2 + t^2}/\eta, q * \sqrt{s^2 + t^2}/\eta) = 1, \\ 0 & \text{else } ncpix(p, q) = 0, \end{cases}$$

where  $p \in [-\eta, \eta]$  and  $q \in [-\eta, \eta]$ . Fig. 6 shows examples of this normalization.

A rotation invariant feature is defined to be a special kind of feature that is invariant when the input pattern is rotated. For character recognition, there are some previously proposed rotation invariant features such as ring projection, the number of strokes, the number of stroke intersections, the number of multi-fork points, Fourier descriptor moments, classical rotation invariant moments, and Zernike invariant moments, etc.

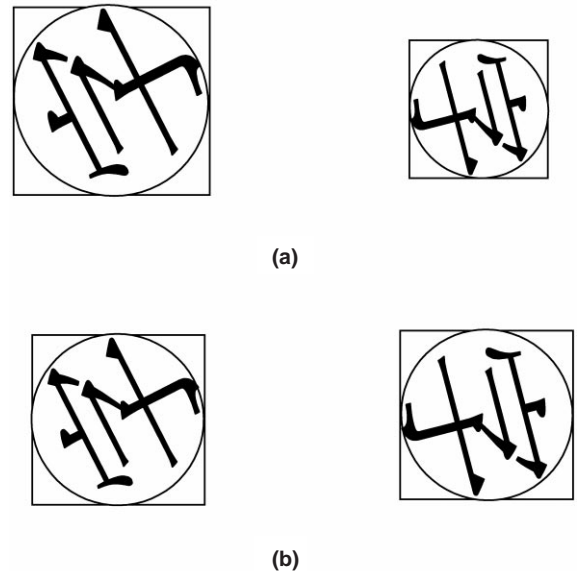


Fig. 6. (a) Two segmented character images after translation. (b) The normalized images.

(Perantonis and Lisboa, 1992; Fukumi et al., 1992; Wong et al., 1995; Fukumi et al., 1997; Khotanzad and Hong, 1990; Wang and Lin, 1996; Wang et al., 1994; Deseilligny et al., 1995; Chiu and Tseng, 1997). We propose another kind of rotation invariant feature that takes the Chinese character into consideration.

Each pattern (or each character) is represented by  $F \times N \times A$  features, which count the number of occurrences of geometry measures  $(p, q, r)$  of all pairs of foreground pixels. Let  $x$  denote the feature vector and  $x(p, q, r)$  be a feature corresponding to measures  $(p, q, r)$  in the vector, where  $i \in [0, W]$ , and  $j \in [0, H]$ .  $F$ ,  $N$ , and  $A$  are positive integers.

The feature vector is initialized as zero and becomes the counter of the occurrences of a geometry relationship between each pair of pixels in the image. We design a rotation invariant feature vector such that the same pair of pixels corresponds to the same geometry measures  $(p, q, r)$  after the rotation operation. In Fig. 7,  $p_1$  and  $p_2$  are two foreground pixels with the coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$  in a character image after the preprocessing. For convenience, we assume  $\sqrt{x_2^2 + y_2^2} \leq \sqrt{x_1^2 + y_1^2}$ . So  $(x_1, y_1)$  is a far pixel and  $(x_2, y_2)$  is a near pixel. The following measure is formed,

$$\left( \left\lfloor \frac{\sqrt{x_1^2 + y_1^2} * F}{\eta + \delta} \right\rfloor, \left\lfloor \frac{\sqrt{x_2^2 + y_2^2} * N}{\eta + \delta} \right\rfloor, \left\lfloor \frac{angle * A}{\pi + \delta} \right\rfloor \right).$$

$F$ ,  $N$ , and  $A$  determine the range of the quantized values and  $\delta$  is a very small value that is set to be  $10^{-5}$  for the convenience of quantization. The variable *angle* represents the value of the angle from the near pixel to the far pixel in counter-clockwise. In our implementation, we compute the quantized value of the angle by rotating the near pixel and far pixel together until the near pixel falls on the  $X$ -axis, then check which segment the far pixel is in. The rotation is a simple computation and could be easily determined by the coordinates of the near pixel. An example is shown in Fig. 7, where  $A = 8$  and  $\lfloor angle * A / (2\pi + \delta) \rfloor = 1$ .

It is clear that the proposed feature vector is rotation invariant since all of the variables in the

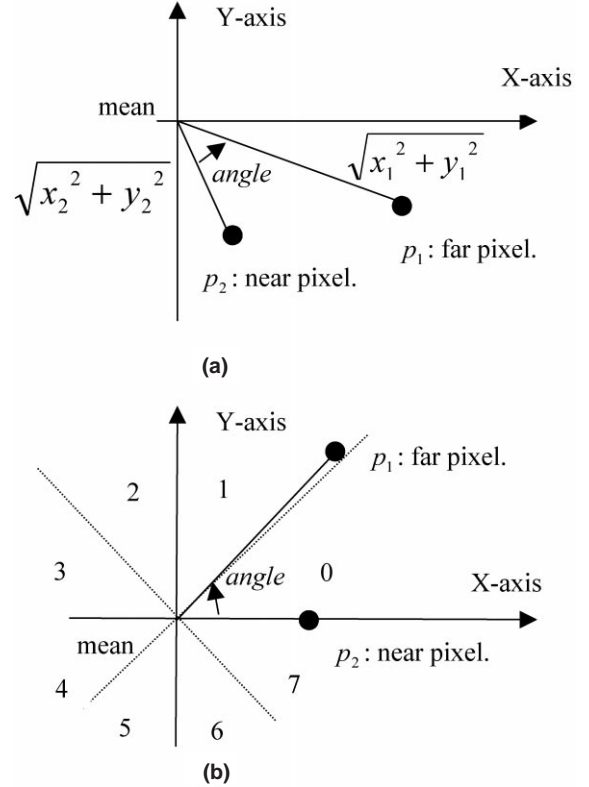


Fig. 7. (a) Compute the feature vector between two pixels  $p_1$  and  $p_2$ . (b) Rotate  $p_1$  and  $p_2$  together until  $p_2$  overlaps with the  $X$ -axis, the number of the segment that  $p_1$  falls in is the quantized value of the angle.

geometry measures are invariant to rotation operations. The following is the proposed feature extraction method.

*Feature extraction method.* Initially, set  $x(p, q, r) = 0$ , for all  $p$ ,  $q$ , and  $r$ . For each pair of foreground pixels  $(x_1, y_1)$ , and  $(x_2, y_2)$ ,

$$x \left( \left\lfloor \frac{\sqrt{x_1^2 + y_1^2} * F}{\eta + \delta} \right\rfloor, \left\lfloor \frac{\sqrt{x_2^2 + y_2^2} * N}{\eta + \delta} \right\rfloor, \left\lfloor \frac{angle * A}{2\pi + \delta} \right\rfloor \right) = x \left( \left\lfloor \frac{\sqrt{x_1^2 + y_1^2} * F}{\eta + \delta} \right\rfloor, \left\lfloor \frac{\sqrt{x_2^2 + y_2^2} * N}{\eta + \delta} \right\rfloor, \left\lfloor \frac{angle * A}{2\pi + \delta} \right\rfloor \right) + 1. \quad (1)$$

If the system memory is sufficient, the various measures could be computed in advance and

stored as a reference table. Using memory reference instead of computing the measures directly could speed up the processing steps.

In some ways, this method could be compared with the Hough transform. The Hough transform uses a two-dimensional (2D) feature vector and the measures are determined by each pixel. Our approach uses a 3D feature vector and the measures are determined by each pair of pixels. In both approaches, the feature vectors are the pre-defined set of bins in the quantized space and represent the histogram. We can call the result of Hough transform as the general feature vector because it could be used for character recognition as well as line or curve extraction (Kushnir et al., 1983).

#### 4. Comparative experiments and histogram equalization

Strokes projection may be the most popular feature used in the Chinese characters recognition society (Yamada et al., 1990; Trier et al., 1996). The rotation invariant version of that kind of feature is called ring projection. Instead of projecting the strokes in the horizontal and vertical directions, ring projection is the histogram of the strokes number in a circular direction. Fig. 8 shows the crossing counts between the circle and the strokes are rotation invariant. We use  $RP(s)$  to denote ring projection feature, where  $s \in [0, S - 1]$ .  $S$  is the segmentation number of the distances. The following is a ring projection feature extraction method: Initially set  $RP(s) = 0$  for all  $s$ . For each foreground pixel with the coordinates  $(x, y)$ , if this pixel is a boundary point along the circular direction then

$$RP\left(\left\lfloor \frac{\sqrt{x^2 + y^2} * S}{\eta + \delta} \right\rfloor\right) = RP\left(\left\lfloor \frac{\sqrt{x^2 + y^2} * S}{\eta + \delta} \right\rfloor\right) + 1. \quad (2)$$

The boundary point is detected when a transition between the background point and the foreground point occurs. In the next, we compare the perfor-



Fig. 8. The crossing counts of the strokes and the circle is rotation invariant.

mance of ring projection feature and the proposed feature. As shown in Fig. 9, we extract 100 Chinese characters that are the 100th, 200th, ..., 10 000th character images in a font base of the popular Eten Chinese operating system. The training samples are original character images and the testing samples are randomly rotated and scaled up images. A nearest-neighbor classification rule is used for recognition. Table 1 shows the results. The increase of the resolution in the segmentation does not necessarily increase the recognition rate in the case of ring projection feature. While the proposed feature reaches 100% recognition rate when the resolution is sufficient as shown in Table 2. Table 3 shows one example of the estimated feature vectors. Carefully comparing the values in the feature vector, we know the discrimination power provided by each dimension of the feature vector is not equivalent. The reason is that the distribution of the pixels number in the distance domain is not uniform. Let us show the distribution. There are 19 838 foreground pixels in the 100 characters. The distances between the origin and these pixels are sorted in an array and plotted in Fig. 10 which shows the distribution is not uniform. Instead of using linear segmentation, the following feature extraction uses a nonlinear segmentation. Assume that the total foreground pixels number in the training set is  $TPN$ . Put the distances between the origin and the pixels into an array,  $DIST(1 : TPN)$  and sort the array. Fig. 10 is an example. Compute  $SF(f) = DIST(f * (\lfloor TPN/F \rfloor))$ , where  $f \in [1, F - 1]$  and  $SN(n) = DIST(n * (\lfloor TPN/N \rfloor))$ , where  $n \in [1, N - 1]$ . Set  $SF(0) = 0$  and  $SN(0) = 0$ . Set  $SF(F) = \eta + \delta$  and  $SN(N) = \eta + \delta$ .  $f$  and  $n$  are index integers.

Feature extraction method with histogram equalization: Initially, set  $x(p, q, r) = 0$ , for all  $p, q,$

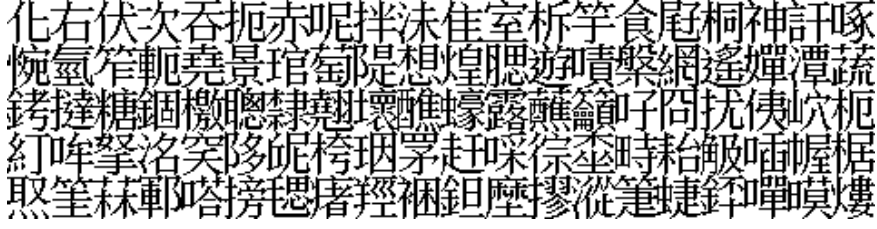


Fig. 9. 100 characters for testing.

Table 1  
Recognition rates of ring projection feature

The segmentation number: $S$	Recognition rates (%)
12	82
13	85
14	84
15	84

Table 2  
Recognition rates of the proposed feature

$F$	$N$	$A$	Recognition rate (%)
4	4	8	88
6	6	8	93
8	8	8	97
4	4	16	93
6	6	16	98
8	8	16	100

and  $r$ . For each pair of foreground pixels  $(x_1, y_1)$ , and  $(x_2, y_2)$ , find  $f$  and  $n$  such that

$$SF(f-1) \leq \sqrt{x_1^2 + y_1^2} < SF(f)$$

and

$$SN(n-1) \leq \sqrt{x_2^2 + y_2^2} < SN(n).$$

$$x\left(f-1, n-1, \left\lfloor \frac{\text{angle} * A}{2\pi + \delta} \right\rfloor\right) = x\left(f-1, n-1, \left\lfloor \frac{\text{angle} * A}{2\pi + \delta} \right\rfloor\right) + 1. \quad (3)$$

Comparing the results shown in Tables 2 and 4, we know this feature with histogram equalization could reach a similar recognition rate with a smaller number of dimensions.

Table 3  
The feature vector of the upper-left character in Fig. 9 when  $F = 4$ ,  $N = 4$ , and  $A = 8$ 

$$\begin{aligned}
 x(:, :, 0) &= \begin{pmatrix} 79 & 0 & 0 & 0 \\ 113 & 72 & 0 & 0 \\ 151 & 173 & 69 & 0 \\ 238 & 151 & 192 & 94 \end{pmatrix}, & x(:, :, 1) &= \begin{pmatrix} 46 & 0 & 0 & 0 \\ 143 & 93 & 0 & 0 \\ 160 & 147 & 72 & 0 \\ 130 & 102 & 113 & 37 \end{pmatrix}, \\
 x(:, :, 2) &= \begin{pmatrix} 105 & 0 & 0 & 0 \\ 174 & 58 & 0 & 0 \\ 132 & 121 & 50 & 0 \\ 73 & 159 & 97 & 52 \end{pmatrix}, & x(:, :, 3) &= \begin{pmatrix} 65 & 0 & 0 & 0 \\ 156 & 44 & 0 & 0 \\ 136 & 160 & 92 & 0 \\ 173 & 190 & 178 & 59 \end{pmatrix}, \\
 x(:, :, 4) &= \begin{pmatrix} 54 & 0 & 0 & 0 \\ 175 & 55 & 0 & 0 \\ 183 & 144 & 82 & 0 \\ 78 & 131 & 170 & 83 \end{pmatrix}, & x(:, :, 5) &= \begin{pmatrix} 82 & 0 & 0 & 0 \\ 142 & 66 & 0 & 0 \\ 135 & 123 & 58 & 0 \\ 197 & 173 & 113 & 76 \end{pmatrix}, \\
 x(:, :, 6) &= \begin{pmatrix} 39 & 0 & 0 & 0 \\ 127 & 97 & 0 & 0 \\ 142 & 170 & 76 & 0 \\ 161 & 129 & 140 & 29 \end{pmatrix}, & x(:, :, 7) &= \begin{pmatrix} 91 & 0 & 0 & 0 \\ 126 & 76 & 0 & 0 \\ 117 & 118 & 62 & 0 \\ 72 & 87 & 119 & 98 \end{pmatrix}.
 \end{aligned}$$



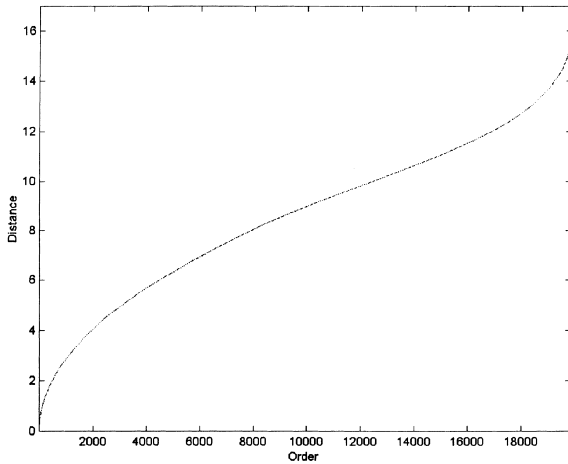


Fig. 10. This figure shows that the distribution of the pixels in the distance domain is not uniform.

Table 4  
Recognition rates of the proposed feature with histogram equalization

$F$	$N$	$A$	Recognition rate (%)
4	4	8	94
6	6	8	96
8	8	8	99
4	4	16	99
6	6	16	100
8	8	16	100

## 5. Clustering and candidates selection

Because of the large volume of Chinese characters, a multi-level clustering algorithm is often used to reduce the recognition time (Tung et al., 1994; Hwang et al., 1997). A clustering procedure is applied to the training character set for constructing a net called a clustering net. Each cluster in the net is composed of a subset of the whole

Chinese characters. Only the characters in the dedicated cluster are qualified for the next run. In this paper, we call this training process clustering and the recognition process candidates selection. The selected candidates form a set for the input to the nearest-neighbor recognition step. This final classifier performs the task of target character recognition. Fig. 11 is an illustration of this recognition structure. The number of candidates is reduced after each assignment of the dedicated cluster. Note that the clustering algorithm used in our system is an overlapped clustering such that the total number of candidates in the sub-clusters is greater than the original cluster.

Our three-stage recognition system contains a two-stage clustering and another stage for target character recognition. The clustering algorithm used in this system is frequency-sensitive competitive learning (FSCL) (Desieno, 1988). In the first stage, the input feature vectors contain 13053 feature vectors and  $c_1$  clusters are generated after FSCL and the overlapped membership assignment. In the second stage, FSCL and overlapped membership assignment rule are performed again and  $c_2$  clusters are generated for each cluster. FSCL is designed to overcome the initialization problem which occurs in the ordinary clustering algorithm. In FSCL, each prototype incorporates a count of the number of times it has been the winner. The distance measure is modified to give prototypes with a lower count value a chance to win the competition. The prototypes  $V = \{v_1, v_2, \dots, v_c\}$  are cluster centers.  $u_i$  is the number of times  $v_i$  has been the winner.

FSCL algorithm (Desieno, 1988):

*Input:* all of the training feature vector set  $X = \{x_1, x_2, \dots, x_n\}$  and the number of clusters  $c$ .

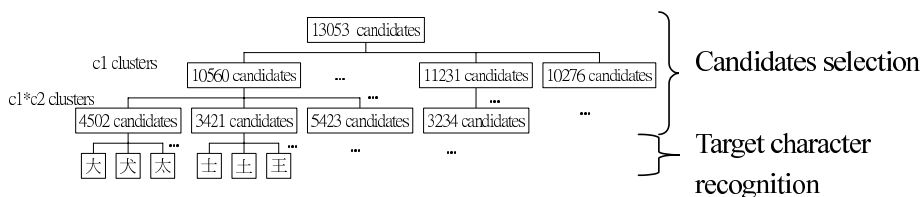


Fig. 11. An illustration of our recognition structure.

*Output:* the final prototypes of clusters  $V = \{v_1, v_2, \dots, v_c\}$ .

*Procedure:*

*Step 1.* Initially set the iteration count  $t = 1$ , iteration bound  $T$ , winning count  $u_i = 0$ , and learning coefficient  $\alpha_0 \in (0, 1]$ .

*Step 2.* Randomly set the initial prototype set  $V = \{v_1, v_2, \dots, v_c\}$ .

*Step 3.* Compute  $\alpha_t = \alpha_0(1 - t/T)$ .

*Step 4.* For  $k = 1, 2, \dots, n$ , find the winning neuron  $v_i$ , such that  $u_i * \|x_k - v_i\| = \min_{1 \leq j \leq c} \{u_j * \|x_k - v_j\|\}$ . Update the winner and winning count with  $v_i^{\text{new}} = v_i^{\text{old}} + \alpha_t * (x_k - v_i^{\text{old}})$  and  $u_i = u_i + 1$ .

*Step 5.* Add 1 to  $t$  and repeat steps 3–5, until  $t$  is equal to  $T$ .

After the clustering with FSCL, the following rule is used. Let  $\text{mem}(i, j)$  denote the membership of character  $j$  in cluster  $i$ .

*Overlapped membership assignment rule:* For each training feature vector  $x_j$ , find the first  $d$  winning prototypes. If  $v_i$  is among the first  $d$  winning prototypes, then  $\text{mem}(i, j) = 1$ , else  $\text{mem}(i, j) = 0$ .

The overlapping degree,  $d$ , in the first and second stages is represented by  $d_1$  and  $d_2$ , respectively. After training, a simple nearest-neighbor classifier is used for the determination of the winning prototype in candidates selection. The final target character recognition is also composed of a nearest-neighbor classifier formed by the feature vectors belonging to the winning cluster.

## 6. Simulation results

It is clear that if the image domain is continuous then a recognition system using the proposed rotation invariant feature could reach the 100% recognition rate. Since digital images are scanned and stored with a limited resolution in the computer, the rotation of images will cause distortion (Cheng et al., 1990). In the simulation, we use digital images of the Chinese characters that come from the popular Eten Chinese operating system. There are six fonts in the operating system. Each character image is composed of  $24 * 24$  binary

Table 5

Feature extraction parameters

Segmentation number for the far pixel, $F$	5
Segmentation number for the near pixel, $N$	5
Segmentation number of angles, $A$	16

Table 6

Clustering parameters

Iteration number, $T$	100
Clusters number in the first stage, $c_1$	40
Clusters number in the second stage, $c_2$	25
Overlapping degree in the first stage, $d_1$	25
Overlapping degree in the second stage, $d_2$	8
Initial learning rate, $\alpha$	1

Table 7

The results of our recognition system

Classifying stage	Recognition rate (%)
First cluster assignment	99.5
Second cluster assignment	98.3
Final target character recognition	97.4

pixels. The training samples are original character images and the testing samples are randomly rotated and scaled up images. By comparison of various parameter settings, we used the parameters shown in Tables 5 and 6. The average number of candidates after the first cluster assignment is 8158 and reduces to 2610 after the second assignment. Using the feature extraction rule with histogram equalization, the recognition rate is shown in Table 7. Note that if the histogram equalization is not used, the segmentation number must be increased from 5 to 7 to produce a similar recognition rate. The recognition rate in Table 7 is a decreasing number because it is impossible for the later stage to produce correct recognition if the right character is not in the cluster assigned by the previous stage. We implemented this system on a personal computer with an Intel's Pentium III 500 MHz processor. The recognition speed is about 6 characters per second. We also tried to use some techniques such as thinning or contour algorithm in the preprocessing to reduce the number of the foreground pixels and speed up the process of feature extraction. But unfortunately, these ap-

proaches are not rotation invariant and the final recognition rate drops more than 5%.

## 7. Conclusions

This paper describes an approach for a rotation invariant printed Chinese character recognition system. We propose a rotation invariant feature and build a three-stage recognition system. The procedure to compute the feature from each pair of pixels is given. The proposed system reaches a 97.4% recognition rate.

In related fields, there are some associated problems such as the segmentation of a rotation character image from maps or 3D objects and the detection of noise in the segmented image. In this paper, we construct a part of the overall system from the segmented character image to the final recognition. The recognition rate of our system may be further improved if different rotation invariant features are used in each stage. These issues are topics for our future research.

## References

- Cheng, H.D., Tang, Y.Y., Suen, C.Y., 1990. Parallel image transformation and its VLSI implementation. *Pattern Recognition* 23 (10), 1113–1129.
- Chiu, H.P., Tseng, D.C., 1997. Invariant handwritten Chinese character recognition using fuzzy min–max neural networks. *Pattern Recognition Letters* 18, 481–491.
- Cheng, D., Yan, H., 1998. Recognition of handwritten digits based on contour information. *Pattern Recognition* 31 (3), 235–255.
- Deseilligny, M.P., Men, H.L., Stamon, G., 1995. Character string recognition on maps a rotation-invariant recognition method. *Pattern Recognition Letters* 16, 1297–1310.
- Desieno, D., 1988. Adding a conscience to competitive learning. *Proceedings of the IEEE International Conference Neural Networks* 1, 117–124.
- Fukumui, M., Omatu, S., Takeda, F., Kosaka, T., 1992. Rotation-invariant neural pattern recognition system with application to coin recognition. *IEEE Transaction on Neural Networks* 3 (2), 272–278.
- Fukumui, M., Omatu, S., Nishikawa, Y., 1997. Rotation-invariant neural pattern recognition system estimating a rotation angle. *IEEE Transaction on Neural Networks* 8 (3), 568–581.
- Hwang, W.J., Ye, B.Y., Lai, L.Y., 1997. Nonparametric classifier design using greedy tree-structure vector quantization techniques. *Pattern Recognition Letters* 18, 409–414.
- Kushnir, M., Abe, K., Matsumoto, K., 1983. An application of the Hough transform to the recognition of Hebrew characters. *Pattern Recognition* 16 (2), 183–191.
- Khotanzad, A., Hong, Y.H., 1990. Rotation invariant image recognition using feature selected via a systematic methods. *Pattern Recognition* 23 (10), 1089–1101.
- Lu, Y., 1995. Machine printed character segmentation – an overview. *Pattern Recognition* 28 (1), 267–273.
- Perantonis, S.J., Lisboa, P.J.G., 1992. Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Transaction on Neural Networks* 3 (2), 241–245.
- Suen, T.Y., Zhang, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. *Communications on ACM* 27, 236–239.
- Tung, C.H., Lee, H.J., Tsai, J.Y., 1994. Multi-stage pre-candidate selection in handwritten Chinese character recognition system. *Pattern Recognition* 27 (8), 1093–1102.
- Trier, O.D., Jain, A.K., Taxt, T., 1996. Feature extraction methods for character recognition – a survey. *Pattern Recognition* 29 (4), 641–662.
- Wong, W., Siu, W., Lam, K., 1995. Generations of invariant moments and their uses for character recognition. *Pattern Recognition Letters* 16, 115–123.
- Wang, S.S., Lin, W.G., 1996. A new self-organization model for invariant pattern recognition. *Pattern Recognition* 29 (3), 677–687.
- Wang, S.S., Chen, P.C., Lin, W.G., 1994. Invariant pattern recognition by moment Fourier descriptor. *Pattern Recognition* 27 (12), 1735–1742.
- Yamada, H., Yamamoto, K., Saito, T., 1990. A nonlinear normalization method for handprinted KANJI character recognition-line density equalization. *Pattern Recognition* 23 (9), 1023–1029.