

## Automatic Chinese Seal Identification

TING-JUN FAN AND WEN-HSIANG TSAI

*Institute of Computer Engineering, National Chiao Tung University  
Hsinchu, Taiwan 300, Republic of China*

Received September 13, 1982

A combination of image processing and matching techniques is proposed to solve the problem of seal identification. The techniques include thresholding, rotation, thinning, reexpansion, and finally matching of seal images. The principle for identification is based on the fact that the spatial distributions of the stroke skeletons of the characters engraved by human hands on two distinct seals will never be totally identical in appearance. An appropriate similarity measure for matching stroke skeletons of characters is proposed. A high identification rate of sample seal imprints reveals that the proposed approach is effective.

### 1. INTRODUCTION

This paper is concerned with the automatic identification of Chinese seals. Seals, also called chops, signets, or stamps, are extensively used in the oriental countries, especially in China, for entity identification. The entity may be a person, a social group, a department of the government, or even the country itself. While using a seal, a user stains the seal with red or blue paint, and then sets the seal on paper to get an imprint. For example, the imprints of the two personal seals shown in Fig. 1(a) are shown in Fig. 1(b).

For the purpose of person identification, seals are used in China instead of signatures on many types of articles such as paintings, documents, diplomas, checks, receipts, money withdrawing lists, etc. Under the reasonable assumption that the imprints of two distinct hand-engraved seals, even with the same set of characters, will never look exactly identical under human visual inspection, it is easy to see that seal identification serves the same function as signature identification which is adopted for identity proving in the western countries.

A distinguished characteristic of seal identification is that for two seal imprints to be considered exactly matched (i.e., they are produced from a single seal), not only the characters in the two imprints must be exactly the same in form, but also the scales, widths, lengths, distributions, relative spatial positions, etc., of all the strokes in either imprint must be identical to those in the other imprint, if the two imprints are not blurred or distorted. This characteristic of seal identification is somewhat different from those of other types of pattern recognition. For example, character recognition [1, 2] only requires two characters being matched to be the same in form, while signature identification [3] requires two signatures being compared to be identical both in form and in writing style. But neither character recognition nor signature identification requires that stroke scales, widths, or relative spatial positions be exactly identical. On the other hand, fingerprint identification [4-6] seems quite similar to seal identification in nature because two fingerprints, if belonging to

a single person, must also be identical in their line (or stripe) lengths, distributions, and relative positions. But fingerprints usually are identified according to the structural patterns (arches, loops, whorls, etc.) existing in the fingerprints without utilizing much of the relative geometrical or spatial information in the prints. This is feasible because human beings scarcely try to create artificial fingerprints for any criminal purpose. However, in the case of seal identification which primarily is to prevent somebody from using a false seal usually with a set of characters identical in form (i.e., in structure) to that on the real seal copy, identifying seals roughly according to character structures is obviously inadequate. What is needed is a method which can be used to compare not only the global character structure but also the local stroke geometry or relative spatial distribution.

A problem encountered in seal identification is the nearly infinite variation of seal imprints even if the imprints are all produced from a single seal. The variation results from many factors. First, the force a person uses in setting a seal on paper is almost always different and often nonuniform. Next, the paint a person uses to stain the seal may vary in color each time. Even with an identical paint, the seal may also be stained nonuniformly on the seal head. Thus, the resulting seal imprints are quite different from time to time. It is often found that a seal imprint contains spot noises, blurred parts, broken edges, missing strokes, or even double settings, etc. Finally, a seal may be set in various directions, resulting in rotated seal imprints. Figure 2 shows some seal imprints with various flaws. With so many types of variation of seal imprints, automatic seal identification is not an easy problem. In this paper, the seal imprints to be identified are assumed to satisfy the following constraints:

- (1) the seals used are all square in shape;
- (2) the seals can not be set upside down and cannot be rotated more than  $45^\circ$  from their normal orientation;
- (3) the seals are set on paper with enough care so that the resulting seal imprints are perfect enough for identification, or at least with as few of the flaws shown in Fig. 2 as possible.

The above constraints are imposed just to simplify the work of automatic seal identification. They might seem too restrictive, but actually are not. Square-shaped

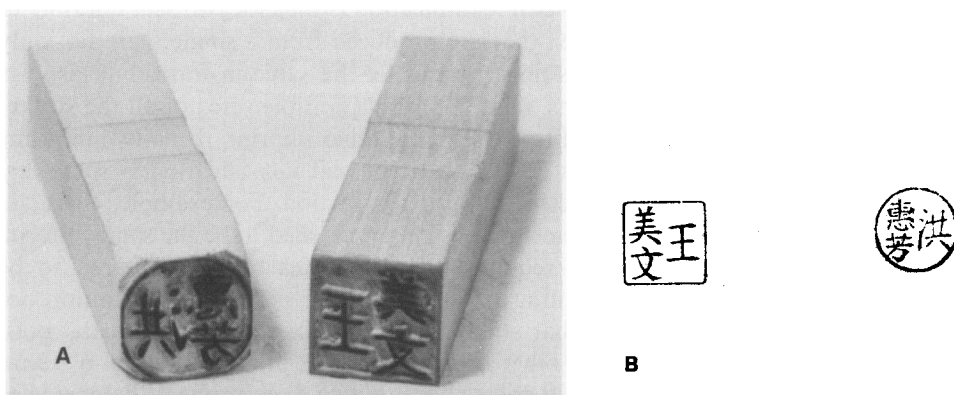


FIG. 1. Imprints of two seals.

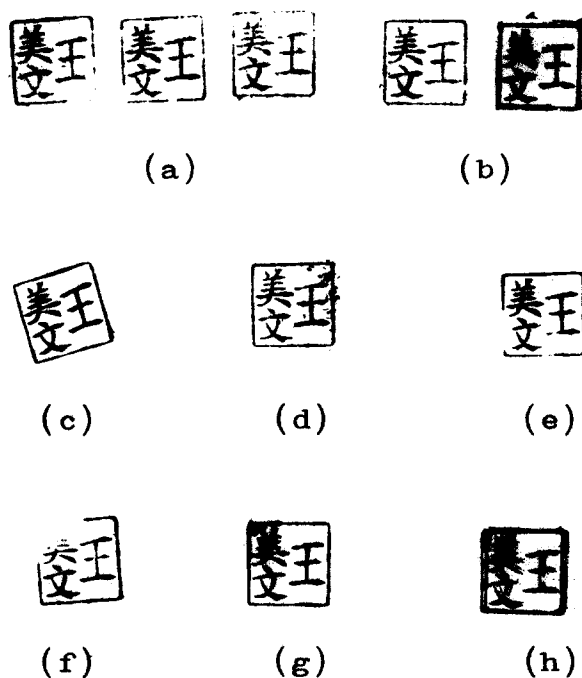


FIG. 2. Seal imprints with (a) nonuniformly stained paints, (b) different stroke widths, (c) rotation, (d) spot noises, (e) a broken edge, (f) missing strokes, (g) blurred parts, (h) double setting.

seals are the most commonly used ones in real cases. Furthermore, users are almost always required to set seals on paper carefully so that the resulting imprints are as perfect as possible with little rotation.

In the remaining part of this paper, we first introduce a new approach to measuring the similarity between characters, which is based on the aforementioned distinguished characteristic of seal identification. A system for seal identification is described next, followed by descriptions of intermediate steps of the identification procedure. Experimental results are also included. Suggestions for further work are finally pointed out.

## 2. SIMILARITY MEASURE FOR MATCHING SEAL CHARACTERS

With seal imprints satisfying the previously mentioned constraints, it seems that seal identification can be easily accomplished by correlation or template matching of thresholded seal images. But careful inspection of a lot of seal imprints from a single seal reveals that the solution is not so simple. A noticeable variation of stroke width is always found to exist in different seal imprints, even if the imprints are from a single seal. See Fig. 2(b) for an example of such stroke width variation. This means that if thresholded seal images are matched directly, the resulting degree of matching of two imprints from a single seal might be too low to be useful for identification, if the two seal imprints are quite different in stroke width. To solve this problem, we propose matching the *skeletons* of seal character strokes, instead of matching character strokes directly, when computing the similarity between two seal images. The reason is based on an observation described in the following.

When a seal is set with some force on a piece of paper, it is reasonable to assume that the partial force applied locally to either side of a stroke on the seal is approximately identical to that applied to the other side of the stroke, so that even if the width of a stroke imprint varies from one seal copy to another, the median axis or the skeleton of the resulting stroke imprint will always remain in a relatively fixed position in the whole seal copy. In other words, it is reasonable to assume that the geometrical or spatial distribution of the stroke skeletons in any seal imprint produced from a single seal will never change. On the contrary, we can also say that the geometrical or spatial distribution of the stroke skeletons of one seal cannot be totally identical to that of another distinct seal.

Now, our problem is to define a match measure function for comparing the similarity between the stroke skeletons of two seal imprints. Some of the simple match measures, such as correlation coefficients, production correlations, mean absolute values of the difference, etc. [7], seem applicable here. However, since the essence of matching stroke skeletons, which are lines or curves, is to measure the similarity of the relative spatial positions of the stroke skeletons, all the above simple match measures actually are inappropriate for this purpose. The reason is that these measures basically only count overlapping pixels in computing the similarity values, and ignore nonoverlapping pixels even if these pixels are very close spatially. However, in the case of matching stroke skeletons of seal characters, after the skeleton images are registered for matching, it can be found that even when the two images are produced from a single seal, the pixels of one stroke in either image do not always overlap the pixels of the corresponding stroke in the other image. An example is shown in Fig. 3. Therefore, what is needed for seal identification is a new match measure by which we can use both overlapping and close nonoverlapping pixel pairs in computing the similarity value, or more generally, a measure by which we can utilize the geometric distances between paired pixels to compute the similarity value. One of such match measures, which we call the *distance-weighted correlation*, is proposed as follows.

#### A. Definition of Distance-Weighted Correlation

Let  $T$  and  $S$  be two curve-type *binary* images to be matched. After  $T$  and  $S$  are overlapped, the *closest pixel*  $t$  in  $T$  of a black pixel  $s(i, j)$  in  $S$  is defined as the black pixel  $t(k, l)$  in  $T$  such that the Euclidean distance  $d(s, t)$  between  $s$  and  $t$  is minimum among all black pixels  $t'(k', l')$  in  $T$ , i.e.,

$$d(s, t) = \min_{t' \in T} d(s, t')$$

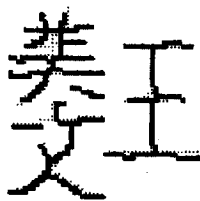


FIG. 3. Strokes in two copies from a single seal do not overlap everywhere.

where

$$d(s, t') = \left( (i - k')^2 + (j - l')^2 \right)^{1/2}.$$

The closest pixel  $s$  in  $S$  of a black pixel  $t$  in  $T$  is similarly defined. It should be noted that the relation defined by the notion of closest pixel is not necessarily one-to-one, and is irreversible. That is, the closest pixel  $t$  in  $T$  of  $s$  in  $S$  may be multiple, and even when  $t$  is single, the closest pixel  $s'$  in  $S$  of  $t$  in  $T$  need not be  $s$ .

Next, the *minimum distance*  $d_p$  of a black pixel  $p$  in either image ( $S$  or  $T$ ) is defined to be the distance between  $p$  and its closest pixel  $q$  in the other image ( $T$  or  $S$ ). And the *weight*  $w_p^K$  of a black pixel  $p$  in either image ( $S$  or  $T$ ) is defined to be

$$w_p^K = \frac{1}{d_p^2 + 1}, \quad \text{if } 0 \leq d_p \leq K,$$

$$= 0 \quad \text{otherwise,}$$

where  $K$  is a constant which sets up a limit of distances within which the closest pixel of  $p$  should be searched for. If the closest pixel of  $p$  is at a distance greater than  $K$  (i.e.,  $d_p > K$ ), then  $w_p^K$  is set 0. In the proposed system,  $K$  is set  $2\sqrt{2}$ , i.e., the closest pixel of  $p$  is searched within a  $5 \times 5$  neighborhood of  $p$ . The reason to set up such a distance limit on computing  $w_p^K$  values is to reduce the time required for computing the proposed *distance-weighted* correlation which finally is defined in the following:

$$C^K(S, T) = \frac{1}{2} \left( \frac{1}{N_S} \sum_{s \in S} w_s^K + \frac{1}{N_T} \sum_{t \in T} w_t^K \right),$$

where  $N_S$  and  $N_T$  are the total numbers of the black pixels (on the lines or curves) in  $S$  and  $T$ , respectively. Two separate terms are included in  $C^K(S, T)$  above to make  $C^K(S, T)$  symmetric (i.e., to make  $C^K(S, T)$  equal to  $C^K(T, S)$ ), because the distance measure  $d_p$  as defined above is unsymmetric. It is not difficult to verify that  $0 \leq C^K \leq 1$  and that  $C^K = 1$  if and only if  $S = T$ . It can also be seen that not only pixels which match exactly but also pixels with small distances apart are all considered in computing the distance-weighted correlation  $C^K(S, T)$ . The following is an illustrative example.

### B. Illustrative Example

Either of Figs. 4(a) and (b) shows the result of overlapping a set of reference data (circles) with a set of testing data (crosses). It can be seen that the two sets of data shown in Fig. 4(b) are more similar to each other with respect to relative spatial positions than those in Fig. 4(a). As a comparison, we first compute the product correlation values, for example, for both cases according to the following definition

$$R(S, T) = \frac{(E(ST))^2}{E(S^2)E(T^2)},$$

where  $S$  and  $T$  are two binary images, and

$$E(S^2) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N p_s^2(i, j),$$

$$E(T^2) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N p_t^2(i, j),$$

$$E(ST) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N p_s(i, j)p_t(i, j),$$

where  $p_s(i, j), p_t(i, j) = 0$  or  $1$  with  $i, j = 1, \dots, N$  are the binary gray values of pixels  $s(i, j), t(i, j)$  in images  $S$  and  $T$ , respectively, and  $N$  is the common size of both images in pixel number. Assume that the images are both  $10 \times 10$  arrays (i.e.,  $N = 10$ ); then we have

(1) for Fig. 4(a),

$$E(ST) = \text{number of overlapping points}/100 = 0.04,$$

$$E(S^2) = \text{number of circles}/100 = 0.20,$$

$$E(T^2) = \text{number of crosses}/100 = 0.20,$$

$$R(S, T) = \frac{(0.04)^2}{(0.20)(0.20)} = 0.04;$$

(2) for Fig. 4(b),

$$E(ST) = 0,$$

$$R(S, T) = 0,$$

because no overlapping point exists.

Therefore, we conclude that the two sets of data shown in Fig. 4(a) are more similar according to the product correlation values, a result contrary to what is expected.

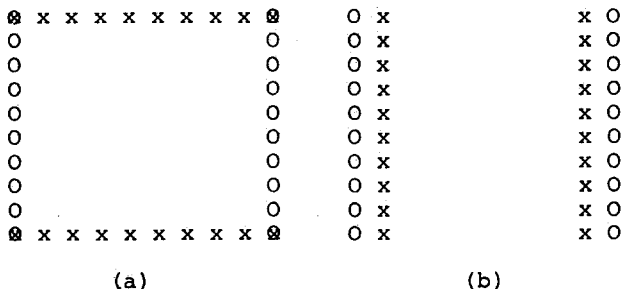


FIG. 4. An example of matching.

Next, we compute the distance-weighted correlation values, for  $K = 2\sqrt{2}$ , as follows:

(1) for Fig. 4(a), by symmetry,

$$C^K(S, T) = \frac{1}{2} \times \left( 2 \times \left( \frac{1}{20} \times \left( 4 \times \left( \frac{1}{1+0^2} + \frac{1}{1+1^2} + \frac{1}{1+2^2} \right) \right) \right) \right) \\ = 0.34;$$

(2) for Fig. 4(b), by repetition,

$$C^K(S, T) = \frac{1}{2} \times \left( 2 \times \left( \frac{1}{20} \times \left( 20 \times \frac{1}{1+1^2} \right) \right) \right) \\ = 0.50.$$

Therefore, the two sets of data shown in Fig. 4(b) are, as expected, more similar according to the distance-weighted correlation values.

### C. A Fast Algorithm for Computing Distance-Weighted Correlation

Without a better algorithm, the computational cost for distance-weighted correlation values can be very high, as is analyzed in the following. Let  $S$  and  $T$  be two binary images whose similarity to each other in terms of a distance-weighted correlation value is to be computed. Let  $N$  be the common size in pixel number of  $S$  and  $T$ , and  $N_S$  and  $N_T$  be the numbers of black pixels in  $S$  and  $T$ , respectively. Before a distance-weighted correlation value  $C^K(S, T)$  can be computed,  $S$  has to be overlapped with  $T$  with respect to a fixed origin. Totally,  $N$  distinct overlappings have to be tried in the worst case. For each overlapping, we have to compute the weight  $w_p^K$  for each black pixel in  $S$  and in  $T$ , and then sum up all the weights. Let  $t_1$  be the maximum time to compute  $w_p^K$  and  $t_2$  be the time for an addition; then totally we need approximately, for the worst case, the following time to compute the  $C^K(S, T)$  values for all overlappings:

$$t_A = N(N_S + N_T)(t_1 + t_2).$$

In the remainder of this section, we propose a fast algorithm which requires approximately the following time to compute all the  $C^K(S, T)$  values for all overlappings:

$$t_B = (N_S + N_T)t_1 + N(N_S + N_T)t_2.$$

And so the time saved is equal to

$$t_A - t_B = (N - 1)(N_S + N_T)t_1.$$

which is great when  $N$ ,  $N_S$ ,  $N_T$ , and  $t_1$  are large. In other words, almost all the time required to compute the weights can be saved in the algorithm to be proposed in the following. This is another advantage of using the distance-weighted correlation as a similarity measure.

Let all the black pixels in image  $S$  be marked  $x$ , and all the blank pixels in  $S$  4-adjacent to  $x$  be marked  $y$ . When image  $T$  is overlapped on  $S$ , the minimum

distance  $d_p$  of each black pixel  $p$  in  $T$  overlapped by an  $x$  is 0, and that of each pixel in  $T$  overlapped by a  $y$  is 1. Therefore, if we can create in advance a new "distance map" from  $S$  in which each black pixel in  $S$  is replaced by distance 0, and each other pixel  $p$  is replaced by the distance  $d(p, p')$  from  $p$  to a nearest black pixel  $p'$  in  $S$ , then the computation of  $d_p$  for each black pixel  $p$  in  $T$  will be made very easy. We propose the use of the curve reexpansion method [8] for this purpose. "Reexpansion" means that the distance map is created by expanding the skeletons which in turn result from shrinking the character strokes (see discussion in Sect. 4). However, instead of marking symbols or using the measure  $d_p$  directly, the weight  $w_p^K$  is introduced. The reexpansion is done by executing the following steps sequentially:

- (1) Assign the weight  $w_p^K = 1$  to each black pixel in  $S$  (since  $d_p = 0$ ).
- (2) Assign the weight  $w_p^K = 0.5$  to each white pixel  $p$  4-adjacent to a black pixel in  $S$  (since  $d_p = 1$ ).
- (3) Assign the weight  $w_p^K = 0.33$  to each white pixel  $p$  diagonal to a black pixel in  $S$  (since  $d_p = \sqrt{2}$ ).
- (4) Assign weights similarly to other pixels with distances  $d_p$  from smaller to larger till  $d_p = K$ .
- (5) Assign the weight  $w_p^K = 0$  to all other blank pixels  $p$  in  $S$ .

Figure 5 shows an example of weight assignment after reexpansion.

After the above reexpansion is done for both images  $S$  and  $T$ , the two images are put to overlap each other according to some origin. Then, we can observe that

- (1) for each black pixel  $s$  (assigned with weight 1 in the above algorithm) in  $S$ ,  $w_s^K$  is just the weight assigned to the corresponding pixel in  $T$  which overlaps  $s$ ; and that
- (2) for each black point  $t$  (assigned with weight 1 in the above algorithm) in  $T$ ,  $w_t^K$  is just the weight assigned to the corresponding pixel in  $S$  which overlaps  $t$ .

An illustration is shown in Fig. 6. Figure 6(a) shows the overlapping of two images with crosses as pixels from one image and circles as those from the other. We see that the minimum distance  $d_x$  of the cross  $x$  is 2; thus the weight  $w_x^K$  assigned to  $x$  is  $1/(1 + 2^2) = 0.2$ . Figures 6(b) and (c) show the distance maps (reexpansion results)

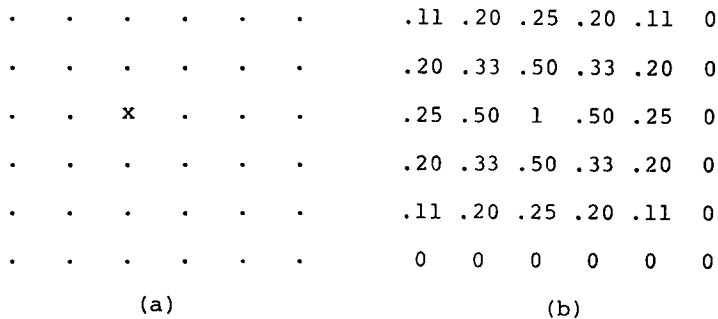


FIG. 5. Example of weight assignment. (a) Before assignment; "x" denotes a black pixel, "." denotes a white pixel. (b) After assignment.



of the two images, with (b) for circles and (c) for crosses, respectively. If Fig. 6(c) is overlapped exactly on Fig. 6(b), we see that the cross, which is assigned weight 1 in Fig. 6(c), is overlapped by a pixel in Fig. 6(b) with weight 0.20 which is exactly the weight  $w_x^K$  of this cross.

Thus, instead of finding the closest pixels for all black pixels, the computation of  $C^K(S, T)$  values can be done simply by reexpansion, overlapping, and summing the weights. We now analyze the complexity of the above proposed algorithm. For a fixed reference image, say,  $S$ , the distance map of  $S$  can be created in advance. This requires time  $N_S t_1$ , where  $t_1$  is the maximum time to compute the weight  $w_p^K$  for a pixel  $p$ . Then, for each input image  $T$ , we first also create the distance map of  $T$ . This requires time  $N_T t_1$ . Now, to compute the  $C^K(S, T)$  values for all  $N$  overlappings in the worst case, we need approximately time  $N(N_S + N_T)t_2$ , where  $t_2$  is the time for addition. Thus, the total time needed in the above algorithm approximately is

$$(N_S + N_T)t_1 + N(N_S + N_T)t_2$$

which is exactly equal to the value  $t_B$  stated previously. Actually, from the viewpoint of real-time identification, even the time  $N_S t_1$  for creating the distance map for the reference image  $S$  can be excluded from  $t_B$  above because this map for  $S$  need only be created once for ever before any seal identification work.

### 3. A SYSTEM FOR SEAL IDENTIFICATION

A seal identification system has been designed and developed around a PDP11/40 minicomputer to simulate the seal identification steps required in the money

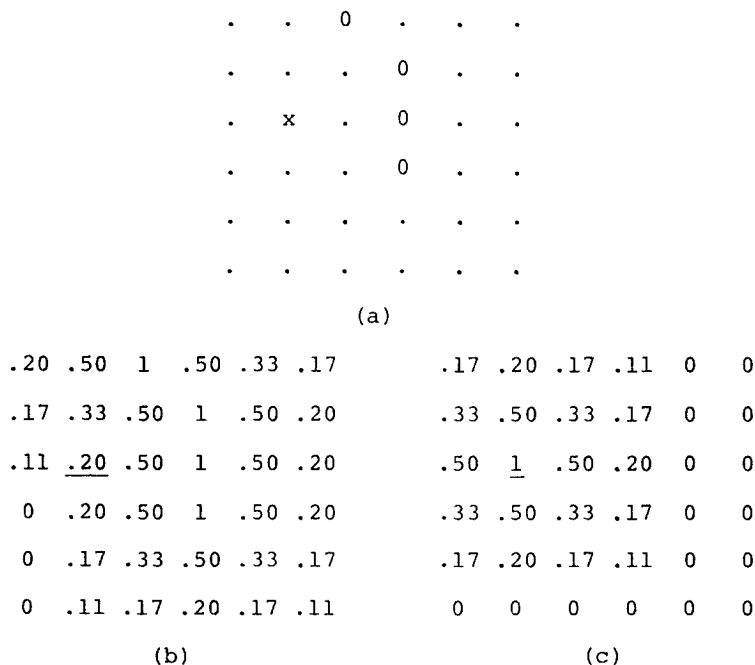


FIG. 6. Illustration of reexpansion.

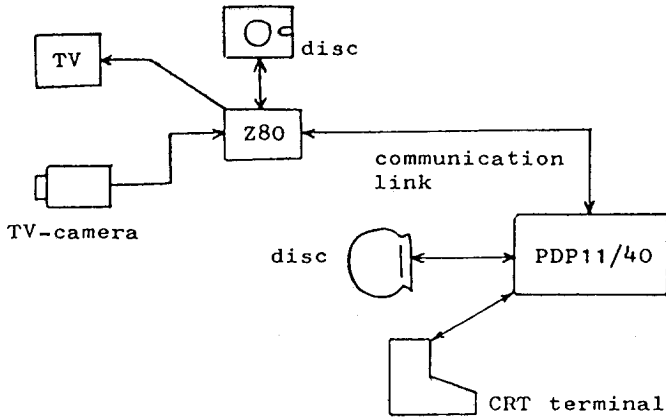


FIG. 7. Hardware configuration of proposed seal identification system.

withdrawing procedure in a banking system. The hardware configuration is shown in Fig. 7. A TV camera controlled by a Zilog Z80 microcomputer is used for scanning seal images, and a communication link between the Z80 and the PDP11/40 is used to transmit image data into the system. Each image taken is an array of 128 by 128 pixels, with each pixel gray value quantized into an 8-bit number (i.e., with 256 gray levels). The proposed seal identification procedure can be roughly divided into two stages—the training stage and the identification stage—which are briefly described in the following. More details will be discussed in the next section.

#### A. Processing on Reference Seal Images—Training Stage

Whenever a customer wants to open a new account, he is required to prepare a seal, and an original seal copy must be left in the system as the reference for later identification of other seal imprints set on money withdrawing lists. The reference seal copy is then processed according to the flow diagram as shown in Fig. 8. At first, the seal copy is scanned into the system as the reference image. Next, the image is thresholded into a binary picture. To facilitate identification, the binary picture is then rotated into its normal orientation, followed by a thinning process used to extract the skeletons of the character strokes. In the next step, the geometric center of all the strokes, called the *mean point* here, is found, which is used later for registration with input unknown seal images. The skeletons found above are then reexpanded into a distance map which, as discussed previously, facilitates the computation of distance-weighted correlation values in the identification stage to be described next. Finally, the reference image together with necessary information for later identification are stored. This completes the training stage of the proposed seal identification procedure.

#### B. Procedure for Identifying Unknown Seal Copies—Identification Stage

The flow diagram of the identification stage of the proposed system is shown in Fig. 9. After the unknown seal copy to be identified is scanned into the system, relevant information of the corresponding reference seal image is retrieved. Processing steps similar to those executed in the training stage are then performed on the unknown seal image except that at the final steps, the expanded skeletons (i.e., the

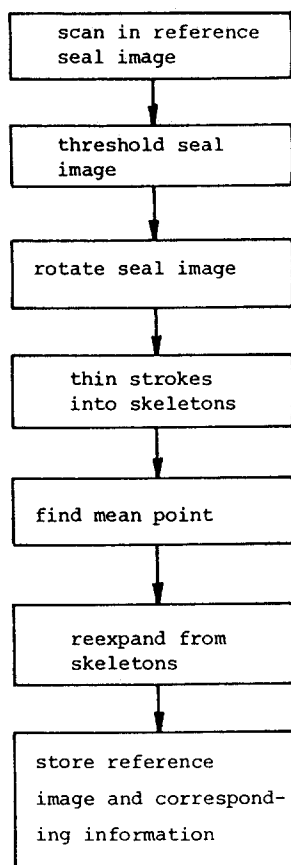


FIG. 8. Flow diagram for the operations on reference seal images.

distance map) of the reference image is retrieved, and registered with those of the unknown image according to their mean points to compute the distance-weighted correlation values for final decision making.

#### 4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section, we describe the details of the intermediate steps of both the training and the identification stages, which mostly are nontrivial. For illustration, the four seal images shown in Fig. 10 and their processed images will be used throughout this paper to demonstrate the intermediate results of the proposed identification procedure. The seal image in Fig. 10(a) will be used as the reference, and Figs. 10(b), (c), and (d) will be used as the seal images to be identified. The images shown in Figs. 10(a) and (b) are produced from a single seal, while 10(c) and (d) are produced from two other distinct seals. So, the expected results are to accept Fig. 10(b) and reject 10(c) and (d).

##### A. Thresholding Seal Images

The first step in either stage (training or identification) of the proposed seal identification system is to threshold the input image into a binary picture. Several

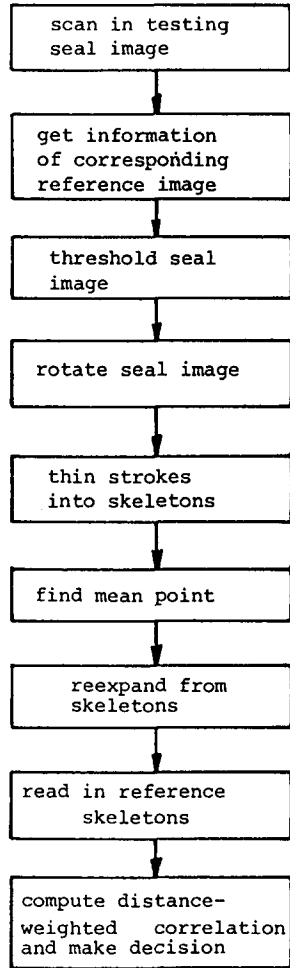


FIG. 9. Flow diagram for the identification steps of unknown seal images.

thresholding methods have been studied. The simplest way is to use a fixed threshold value for all input seal images. Obviously, this does not work because seal imprints are often produced from paints with different colors, resulting in scanned seal images with diverse mean gray values. Thresholding by relaxation [9, 10] has also been tried but was found too time-consuming to be practical for real-time identification. The method most appropriate for the application here is thresholding after gray scale normalization [8] which is described in the following.

Let  $H_T$  and  $H_S$  be the average gray values of an input image  $T$  and a reference image  $S$ . Gray scale normalization transforms the gray value  $g_p$  of each pixel  $p$  in  $T$  into a new gray value  $g_p - H_T + H_S$ . It is easy to see that the average gray value of the transformed image  $T'$  of  $T$  is  $H_S$ . This means that now both  $T'$  and  $S$  are approximately equally "dark." Therefore, we can then threshold  $T'$  with a fixed threshold value  $T_S$  identical to that for  $S$ , the reference image. For practical implementation in the proposed system, instead of transforming  $T$  into  $T'$ , we just threshold  $T$  directly with the threshold value  $T_S - H_T + H_S$ . After thresholding, a

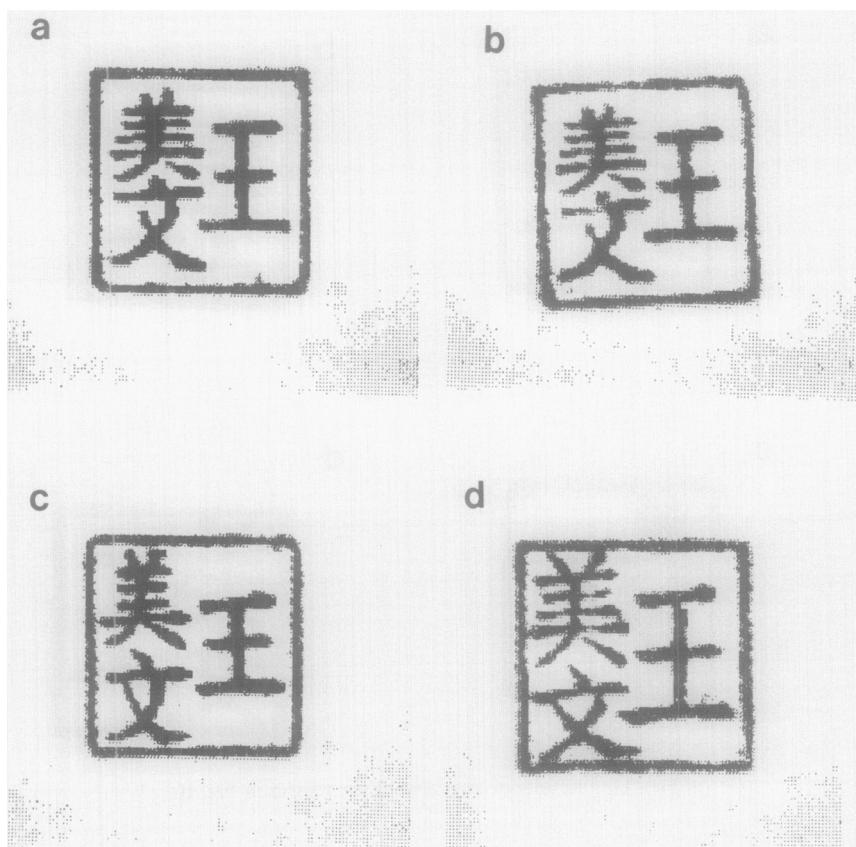


FIG. 10. Image of seal imprints. (a) Reference imprint. (b) An imprint to be identified, produced by the same seal as that of (a). (c), (d) Two other imprints to be identified, produced by two seals different from that of (a) and (b).

decision is made in the proposed system to reject a thresholded image which contains many fewer or many more black pixels than the corresponding thresholded reference image.

The image of Fig. 10 after thresholding is shown in Fig. 11. Since the number of black points in Fig. 11(d) is quite different from that in 11(a), the reference image, the seal imprint of 11(d) is rejected after thresholding, which means that it is regarded as a false seal copy if Fig. 11(a) is used as the reference.

### *B. Rotation of Seal Images*

Before two seal images can be matched for identification, they have to be rotated into an identical orientation. Again, several methods have been studied for this rotational normalization problem. The simplest method is to rotate one image into a lot of directions, each direction say with  $1^\circ$  apart, and then match with the other image to find an optimal match. However, this wastes too much time. Moreover, due to the requirement of high exactness in matching seal images, this method is impractical for the purpose of seal identification. Another general method for rotational normalization is to use the Hotelling transform [11]. Within each binary image, there always exists a standard or invariant direction with the maximum

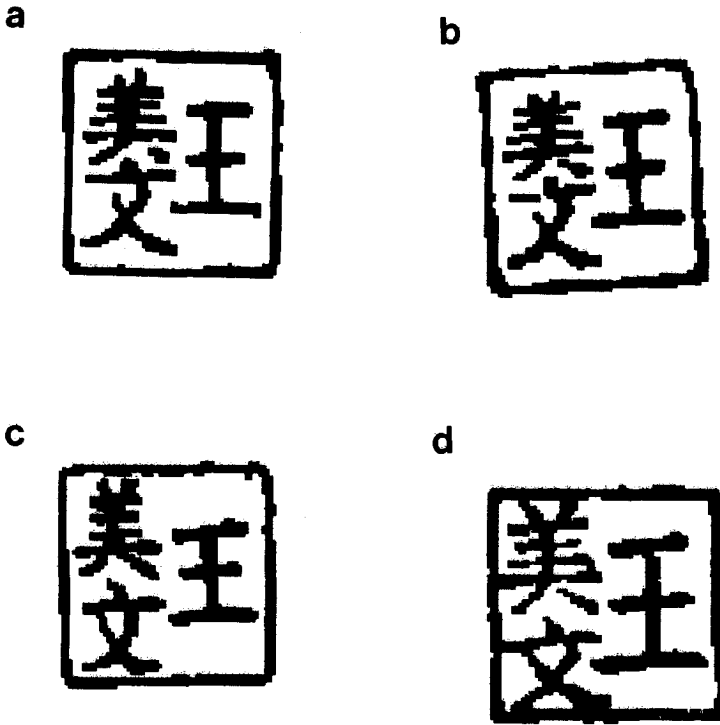


FIG. 11. Results of thresholding the images in Fig. 10.

variance of black pixels in the image. If this standard direction can be found for each image and the image is then rotated into that direction, the rotational normalization problem certainly is solved. However, the standard directions of seal images are found "unstable" possibly due to that high "randomness" of character strokes; a little change in a seal imprint, such a wider stroke, some noise, or a spot, etc., often results in a great change in the computed standard direction. This means that the Hotelling transform, though theoretically sound, is not a suitable method for rotational normalization of seal images.

In this paper, the problem of rotational normalization is solved by using a method modified from Hsieh and Fu [12], which was proposed for normalizing the orientations of IC chips. In this method, a set of imaginary equally spaced vertical (or horizontal) lines are superimposed on the image to be rotated. The constant interval between every two adjacent lines is  $G$ , as shown in Fig. 12. Let these lines be denoted as  $L_i$ ,  $i = 1, 2, \dots, \lfloor \text{NB}/G \rfloor$ , where NB is the width of the image in pixel number. If  $L_i$  intersects the skeleton of one of the seal borders at  $p_i = (x_i, y_i)$ , then the slope of the intersected border, measured at  $L_i$  and  $L_{i+1}$ , is

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}.$$

After  $m_i$  is computed for each  $i$ , if it is greater than or equal to 1, it is removed because it is assumed previously that a seal imprint is not allowed to vary in

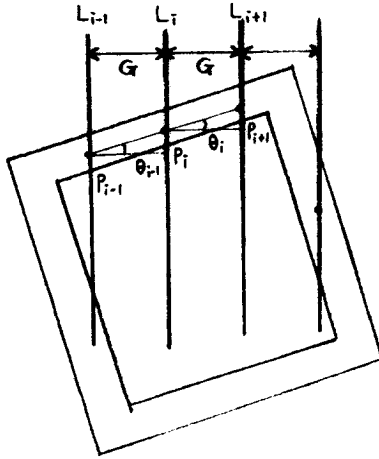


FIG. 12. Illustration of normalizing IC orientation.

direction more than  $\pm 45^\circ$  from its normal orientation. Next, an optimal slope  $m_k$  is chosen among all  $m_i$  as the slope of the whole border such that the absolute error

$$E_k = \sum_{i \neq k} |m_k - m_i|$$

is minimized.

In many cases, there may exist some gaps in the borders of the seal frame and some of the superimposed lines may be drawn through these gaps. One such case is shown in Fig. 13. Then, the slopes calculated will be seriously erroneous. To avoid this type of errors, the calculation of slopes is repeated for several values of  $G$ , which is the gap between every two adjacent superimposed lines. For each  $G = G_j$ , an optimal slope  $m^{(j)}$  is calculated by the method discussed previously. Then, another overall optimal slope  $m^{(l)}$  is chosen from all calculated  $m^{(j)}$  such that the absolute error

$$E^{(l)} = \sum_{j \neq l} |m^{(l)} - m^{(j)}|$$

is minimized. To improve accuracy further, the above calculation is performed for all four borders of a seal imprint, and the final slope  $m$  is chosen similarly as above (in the sense of the minimum absolute error) from the four computed slopes of the borders. After the slope  $m$  is found, the image is then rotated according to  $m$  into its normal orientation. Figure 14 shows the result of rotating the image of Fig. 11(b). It is noted that the images in 11(a) and (c) are already in their normal orientation, and the image in 11(d) has already been rejected after thresholding.

### C. Thinning Strokes into Skeletons and Finding Mean Points

One way to find the skeletons of character strokes is to thin the strokes [8]. Thinning is done by continuously removing the outside pixels of the strokes until the

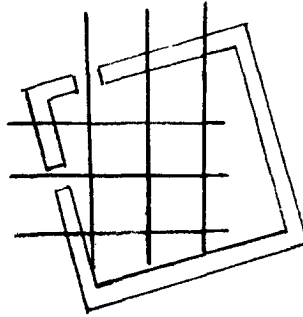


FIG. 13. An example to show the gaps in the borders of a seal imprint.

widths of the strokes become 1 everywhere. To thin an image iteratively, many steps are needed. In the odd-numbered steps, north and west simple border pixels of the strokes are removed, while in the even-numbered steps, south and east simple border pixels are removed. This is necessary because in some cases, simultaneous removal of all north and south border pixels will cause a stroke to vanish completely. On the other hand, the border pixels removed must be *simple*. A pixel is said to be simple if and only if

(1) it is neither an isolated pixel nor an end pixel (i.e., at least two of its four neighbors are black); and

(2) the number of components (i.e., disconnected parts) of the image is not changed by the removal of the pixel. Connectedness is defined using 8 neighbors in this paper.

The results of thinning the images in Figs. 11(a), 14(b), and 11(c) are shown in Fig. 15.

For two images to be matched conveniently, two corresponding pixels, which we call *mean points*,  $M_S = (S_x, S_y)$  and  $M_T = (T_x, T_y)$ , are computed for the reference image  $S = \{p_s(i, j) | i, j = 1, \dots, NB\}$  and the input image  $T = \{p_t(i, j) | i, j = 1, \dots, NB\}$ , respectively, where NB is the width of either image in pixel number. These two mean points are used for registering  $S$  and  $T$  with respect to spatial translation to avoid exhaustive shifting as is normally required in template matching. More specifically, the input image  $T$  is shifted by  $T_x - S_x$  in the  $x$  direction and

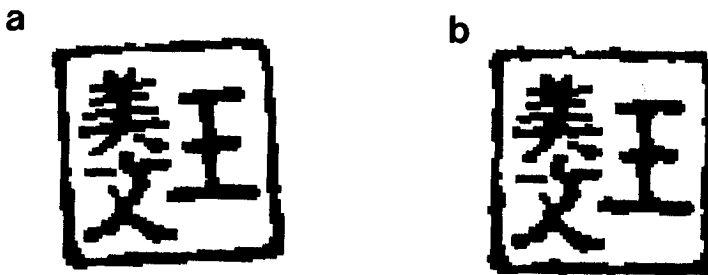


FIG. 14. The result of rotating the image of Fig. 11(b). (a) Before rotation. (b) After rotation.



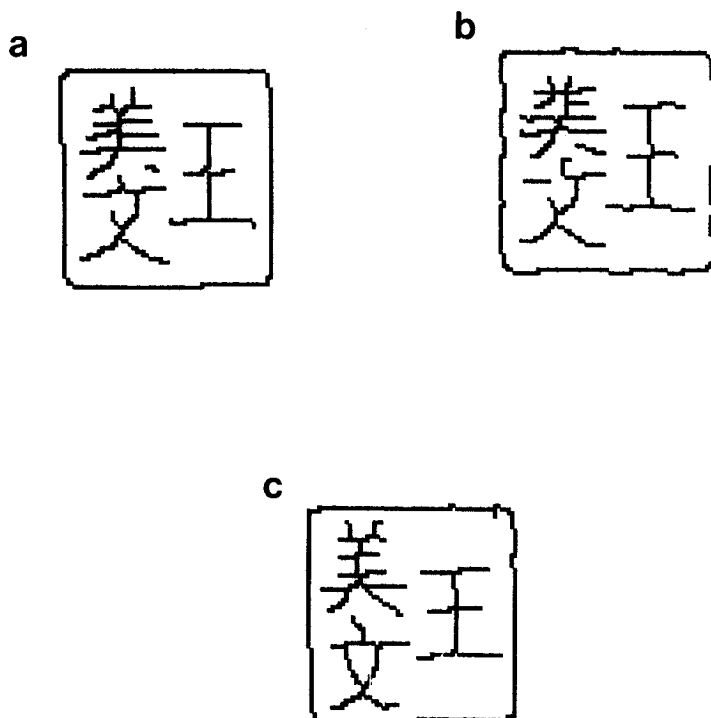


FIG. 15. Results of thinning the rotated images (Figs. 11(a), 14(b), and 11(c)).

$T_y - S_y$  in the  $y$  direction so that  $M_S$  and  $M_T$  overlap, and matching can then be done in a manner of 1-to-1 pixel comparison for each  $p_s(i, j)$  in  $S$  and each  $p_t(i + T_x - S_x, j + T_y - S_y)$  in  $T$ . In this study, the mean points  $M_S$  and  $M_T$  are determined to be the geometrically central points, measured from the four seal borders, of the reference and the input seal imprints, respectively.

#### D. Matching of Skeletons

The match of skeletons of character strokes is done by overlapping two images  $S$  and  $T$  according to their mean points  $M_S$  and  $M_T$  and then calculating the distance-weighted correlation values by the method proposed in Section 3.C. Before this is done, all the seal border pixels are first removed. The reason to do so is twofold. First, it is easier to engrave two seals with almost identical square borders in size and in width. Next, the number of border pixels in a thinned image is large so that if they are not removed before matching, the degree of difference between two distinct seal images will be much reduced.

On the other hand, since the mean points  $M_S$  and  $M_T$  computed originally need not be integers, to increase precision of matching, overlapping of the two images is performed on all eight neighbors of  $M_S$ . That is,  $M_S$  is shifted to any of its eight neighbors, and a corresponding distance-weighted correlation value  $C^K(S, T)$  is computed. The one with the greatest  $C^K(S, T)$  values computed is chosen as the point of best match.

The optimal results of overlapping Fig. 15(b) with 15(a) and overlapping 15(c) with 15(a) are shown in Figs. 16(a) and (b), respectively. The dotted lines denote the



FIG. 16. Optimal results of overlapping.

black pixels of Fig. 15(a), and the black lines denote those of 15(b) and (c), respectively. It is easy to see that the two images in Fig. 16(a) are very similar to each other while those in 16(b) are not, and this agrees with human inspection. In the experiment, the  $C^K(S, T)$  values computed for Figs. 16(a) and (b) are 0.75 and 0.50, respectively.

After the  $C^K(S, T)$  value is calculated, three types of decision are made accordingly:

(1) if  $C^K \geq C_2$ , the input image is accepted;

(2) if  $C^K \leq C_1$ , the input image is rejected;

(3) if  $C_1 < C^K < C_2$ , the input image is classified as "ambiguous" and is displayed for visual inspection by the operator to reduce incorrect identification possibility,

where  $C_1$  and  $C_2$  with  $C_2 > C_1$  are two constants determined by experiments to be 0.705 and 0.735, respectively. With  $C_1$  and  $C_2$  so chosen, if we take the image shown in Fig. 10(b) to match with 10(a) (the optimal overlapping result is shown in Fig. 16(a)), it will be accepted, while the one in Fig. 10(c) (the optimal overlapping result is shown in Fig. 16(b)) will be rejected.

### E. Experimental Results

Eight distinct seals were used for identification in our experiment. Four of them were engraved by a single person, and the other four were engraved by four other distinct persons. The first four engraved by a single person look so similar in appearance that it is difficult to distinguish them simply by visual inspection without the help of other means. Each seal is then used to produce five seal imprints, one as the reference, and the other four as the data to be identified. The 8 reference data are processed according to the training steps shown in Fig. 8, and the remaining 32 imprints are processed according to the identification steps shown in Fig. 9. By matching each of the 32 testing imprints with the 8 reference imprints, totally 256 distance-weighted correlation values are computed. The result of the 256 matchings are shown in Table 1 from which we see that the rate of correct identification is

$$r_1 = 1 - 22/256 = 91.4\%,$$

when all 8 ambiguous cases are assumed to be identified incorrectly, or is

$$r_2 = 1 - 14/256 = 94.5\%,$$

TABLE 1  
The Results of Matching 32 Test Images with 8 Reference Images

	C1 = 0.705				C2 = 0.735					
	BLANK = YES; N = NO; A = AMBIGUOUS									
	AS	BS	CS	DS	ES	FS	GS	HS		
A1	0.80	0.69	0.73	A	0.67	0.74	N	0.51	0.00	0.51
A2	0.78	0.68	0.66	0.69	0.79	N	0.50	0.00	0.52	
A3	0.80	0.69	0.68	0.70	0.80	N	0.51	0.00	0.49	
A4	0.74	0.63	0.64	0.62	0.66	0.46	0.00	0.56		
B1	0.69	0.78	0.60	0.00	0.66	0.50	0.00	0.52		
B2	0.69	0.87	0.63	0.75	N	0.67	0.48	0.00	0.50	
B3	0.68	0.86	0.64	0.74	N	0.66	0.48	0.00	0.50	
B4	0.66	0.70	N	0.61	0.61	0.63	0.45	0.00	0.53	
C1	0.76	N	0.63	0.84	0.70	0.74	N	0.58	0.00	0.00
C2	0.68	0.55	0.71	A	0.62	0.64	0.49	0.00	0.52	
C3	0.71	A	0.60	0.80	0.68	0.68	0.55	0.00	0.53	
C4	0.68	0.61	0.77	0.65	0.65	0.53	0.00	0.52		
D1	0.70	0.78	N	0.65	0.88	0.70	0.52	0.00	0.00	
D2	0.69	0.73	A	0.67	0.78	0.70	0.55	0.00	0.47	
D3	0.70	0.77	N	0.65	0.80	0.70	0.52	0.00	0.50	
D4	0.70	0.69	0.62	0.69	N	0.64	0.50	0.00	0.53	
E1	0.73	A	0.67	0.69	0.72	A	0.79	0.58	0.00	0.45
E2	0.77	N	0.66	0.73	A	0.70	0.80	0.56	0.00	0.46
E3	0.72	A	0.66	0.68	0.67	0.80	0.54	0.00	0.45	
E4	0.69	0.67	0.70	0.66	0.66	0.83	0.56	0.00	0.42	
F1	0.53	0.46	0.57	0.53	0.56	0.83	0.00	0.40		
F2	0.51	0.47	0.59	0.53	0.53	0.79	0.00	0.41		
F3	0.50	0.44	0.56	0.51	0.54	0.82	0.00	0.39		
F4	0.49	0.45	0.55	0.51	0.53	0.75	0.00	0.40		
G1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.00	
G2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.00	
G3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.00	
G4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.68	N	0.00
H1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.58	0.87	
H2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.68	N
H3	0.00	0.50	0.47	0.00	0.48	0.39	0.61	0.79		
H4	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.84		

otherwise. Using FORTRAN IV as the programming language, the time required for each identification is less than 3 min.

##### 5. CONCLUSIONS AND SUGGESTIONS FOR FURTHER DEVELOPMENT

In this paper, a combination of effective image processing and matching techniques is proposed to identify Chinese seal imprints. The experiment shows that the resulting identification is rather successful. The identification rate is more than 90% and the total CPU time spent is less than 3 min, for each identification. To improve the results, several suggestions for further development are given in the following.

(1) In this study, the shape of the seal is constrained to be square for the convenience and efficiency of rotational normalization and mean point finding. It is desirable to find other efficient algorithms for rotational normalization of the seal images of other shapes.

(2) It is also desirable that more powerful processing techniques be found to identify seal imprints with more flaws without increasing identification time too much.

(3) The identification system used in this study was not specially designed. If multiprocessors and special hardware can be constructed so that images can be processed in parallel and faster, the identification time can be reduced.

#### REFERENCES

1. G. Groner, J. Heafner, and T. Robinson, On-line computer classification of handprinted Chinese characters as a translation aid, *IEEE Trans. Electron. Comput.* **EC-16**, 1967.
2. W. Stallings, Approaches to Chinese character recognition, *Pattern Recognition* **8**, 1976.
3. R. N. Nagel and A. Rosenfeld, Steps toward handwritten signature verification, Proceedings, First International Joint Conference on Pattern Recognition, Washington, D.C., 1973.
4. B. Moayer and K. S. Fu, A syntactic approach to fingerprint pattern recognition, *Pattern Recognition* **7**, 1975.
5. B. Moayer and K. S. Fu, A tree system approach for fingerprint pattern recognition, *IEEE Trans. Comput.* **C-25**, 1976.
6. K. Rao and K. Balck, Type classification of fingerprints: A syntactic approach, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**, 1980.
7. P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Probability Theory*, Houghton-Mifflin, Boston, 1977.
8. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic, New York, 1976.
9. A. Rosenfeld and R. C. Smith, Thresholding using relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-3**, 1981.
10. S. Peleg, A new probabilistic relaxation scheme, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-3**, 1981.
11. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1977.
12. Y. Y. Hsieh and K. S. Fu, An automatic visual inspection system for integrated circuit chips, *Computer Graphics Image Processing* **14**, 1980.