

An Effective Method for Fingerprint Classification

Ching-Tang Hsieh^{1*}, Shys-Rong Shyu¹ and Kuo-Ming Hung^{1,2}

¹*Department of Electrical Engineering, Tamkang University,
Tamsui, Taiwan 251, R.O.C.*

²*Department of Information Management, Kainan University,
Taipei, Taiwan, R.O.C.*

Abstract

In this paper, we present a fast and precise method for fingerprint classification. The proposed method directly extracts the directional information from the thinned image of the fingerprint. We use an octagon mask to search the center point of the region of interest and consider both the direction information and the singular points in the region of interest to classify the fingerprints. In the system, not only is the amount of computation reduced but also can the extracted information be used for identification on AFIS. The system has been tested all 4000 fingerprint images on the NIST special fingerprint database 4. The classification accuracy reaches 93.425% with no rejection for 4-class classification problem.

Key Words: Biometrics, Fingerprint Classification, Wavelets, Poincare Index, Singular Point Detection

1. Introduction

Fingerprints are the innate feature of mankind. The patterns of fingerprints become fixed much before a person reaches 14 years. Due to the invariance and uniqueness of fingerprints, they are used as the tool for crime investigation by the government as far back as A.D. 1897. So far, fingerprints have been used for more than 100 years as the most popular biometric signs of identity. With the large fingerprints database, fingerprint classification is an important subproblem for the automatic fingerprint identification system (AFIS) and automatic fingerprint recognition system (AFRS) [1]. Fingerprint classification improves the computational complexity substantially by reducing the number of candidate matches to be considered, so an effective indexing scheme can be of great help.

For fingerprint images, there are two special features from the structure of flow lines of ridges and valleys: the core and the delta points (also referred to as singular points or singularities), and they are highly stable and

also rotation- and scale-invariant. Henry classes [2], which classify the fingerprint images into arch, tented arch, left loop, right loop or whorl (see Figure 1), are defined by using the accurate position and type of such singular points, and Table 1 gives the rules. Most current classification methods, no matter what they are, structural methods or network-based methods, are based on the extraction of such singular points [3,4–6]. However, the methods using the singular points have the draw-

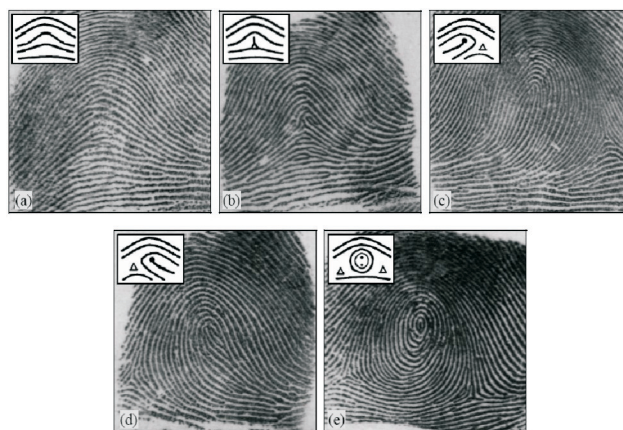


Figure 1. Major Henry fingerprint classes, (a) arch, (b) tented arch, (c) left loop, (d) right loop, (e) whorl.

*Corresponding author. E-mail: hsieh@ee.tku.edu.tw

Table 1. Fingerprint pattern classes and the corresponding number and type of singular points

Pattern class	Core	Delta
Arch	0	0
Tented arch	1	1 (middle)
Left loop	1	1 (right)
Right loop	1	1 (left)
Whorl	2	2

backs below:

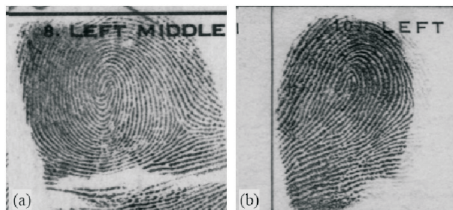
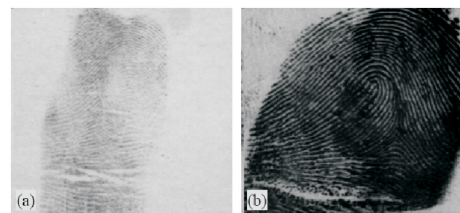
- (1) The information on the regions of singular points on original input images cannot be lost. Because these methods classify the fingerprints according to the number of the singular points, they will not be able to classify the images that lose the information of the singular points regions. (See Figure 2)
- (2) The detection of singular points extremely relies on the quality of input images. While the quality of the fingerprint images varies a lot due to noise, and even different regions show different quality in one image. So, it is difficult to classify such images according to the rules in Table 1. (See Figure 3)

Besides the singularities based methods mentioned above, in the past research there is another common approach, a structure based approach, which uses the estimated orientation field in a fingerprint image to classify the fingerprints into one of the predefined five classes. Wilson et al. [7] test their classifier on NIST-4 database [8] and an accuracy of 92.2 percent with 10 percent rejection is reported. The neural network used in [7] was trained on images from 2,000 fingers (one image per finger) and then tested on an independent set of 2,000 images taken from the same fingers. They further reported a later version of the algorithm [9] tested on the NIST-14 database, which is a naturally distributed database, resulting in a better performance. A similar approach, which

improves the enhancement and the feature extraction on fingerprint images, is also reported. Karu and Jain [3] first determine the singularity region of the orientation flow field from the ridge line map to get a coarse classification. This is followed by a fine classification based on ridge line tracking around singularities. Jain et al. [4] also propose a multi-channel approach to classify the fingerprints. They use the Gabor filter to filter the “disk” area around the pre-defined center of interest and extract the feature vector with four directions for neural network classifier. They train and test the system using 2000 fingerprints from NIST4 database separately. On 4-class classification the system accuracy is 94.8% with 1.8% rejection but the computation is heavier than the proposed method. Another approach using hidden Markov models for classification is presented in [10], but it requires the reliable estimation of ridge locations, it is difficult to perform in noisy images.

In our method of fingerprint classification, we extract the only 2D feature vector of the orientation field directly derived from the thinned images. We first use the enhancement algorithm of fingerprint images presented in *Pattern Recognition*, Vol. 36 [11]. We compute the orientation following the skeleton of thinned image, so it is more effective without another enhancement for AFIS. Then we use an octagon mask to fast search the center point of the region of interest and extract the feature from the lower half of the octagon mask. Finally, we simplify the feature vectors and use the two-stage classifier to analyze the features and output the result of classification.

The rest of the paper is organized as follows. In Section 2, we briefly review the algorithm of fingerprint enhancement and describe the computation of orientation field. In Section 3, we detailed search for the centers of the regions of interest using the octagon mask and feature extraction for classification. Then, Section 4 pre-

**Figure 2.** The information of singular points is lost, (a) whorl type loses a delta point, (b) left loop loses a delta point.**Figure 3.** The poor quality images, (a) the low contrast image, (b) the dim image.

sents our two-stage classification scheme and Section 5 shows the experimental results tested on the NIST-4 database. The conclusions are presented in Section 6.

2. Orientation Field Computation

For the fingerprint classification, the global texture information is more important than the local direction. For an automatic fingerprint identification/recognition system, the algorithm we presented in [11] has shown good results of enhancement, so we combine the fingerprint classification AFIS/AFRS to reduce computation. The detailed processing is presented in the flowing sub-sections.

2.1 Enhancement Algorithm Review

In an earlier work in *Pattern Recognition*, Vol. 36 [11], we propose an effective enhancement method that combines the texture unit of global information with the ridge orientation of local information. Based on the characteristic of hierarchical framework of multiresolution representation, the wavelet transform is used to decompose the fingerprint image into different spatial/frequency sub-images by checking the “approximation region”. In considering the global information, all the texture units within an approximation sub-image are processed by textural filters. Based on the hierarchical relationship of each resolution, all other sub-images are processed according to the related texture units within the approximation sub-image. Similarly, the directional compensation is implemented by voting technique in considering the orientation of local neighboring pixels. After these two enhancement processes, the improved fingerprint image is obtained by the reconstruction process of wavelet transform. The detailed processing of each step is described as follows:

Step 1: Normalization

The purpose of normalization process is to adjust the gray values of an input fingerprint image so that all images have the specific mean and variance values. Let $I(x, y)$ denote the gray value intensity at pixel (x, y) , M and V be the estimated mean and variance of the input fingerprint image, respectively, and $N(x, y)$ be the normalized gray value intensity at pixel (x, y) . For all the pixels, the normalized image is defined as [12]

$$N(x, y) = \begin{cases} Ms + \sqrt{\frac{Vs \times (I(x, y) - M)^2}{V}} & \text{if } I(x, y) > M, \\ Ms - \sqrt{\frac{Vs \times (I(x, y) - M)^2}{V}} & \text{otherwise,} \end{cases} \quad (1)$$

where MS and VS are the predefined mean and variance values, respectively. Normalization operation not only preserves the clarity of the ridge and valley structures but also changes the mean and variance of the image to the defined constant values.

Step 2: Wavelet decomposition

A normalized image is decomposed into some different spatial/frequency sub-images by specified filters for constructing the hierarchical relationship, and the decomposition level is established by checking the approximation region. In this paper we use Haar transform to excise wavelet transform in extracting multiresolution feature, and the wavelet transform flowchart is shown in Figure 4.

Step 3: Global texture filtering

All the approximation and detailed sub-images are converted into texture spectrum domain containing texture unit of all pixels. The global information is used to alleviate the problem of over-inking and under-inking in parts of the fingerprint image with the statistics of all texture units. As described before, the ridge structures of fingerprint can be viewed as some regular textures. This means that we can find many local structures that are similar to other local structures within all the fingerprints. The textural filtering technique is based on the statistics of all texture units. Hence, we can apply this filtering process to improve the quality of the damaged images. The wavelet-based textural filtering method can be described as follows:

1. Decompose the input fingerprint image into some

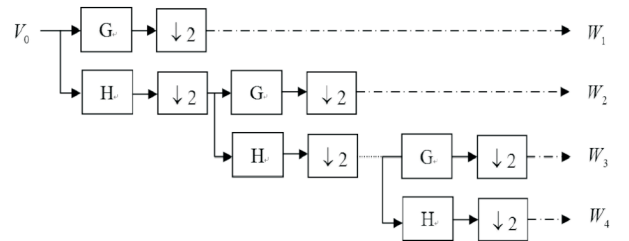


Figure 4. Filter-bank structure implementing a discrete wavelet transform.

sub-images according to the selected optimal decomposition process described in Step 2.

2. Label texture units for all pixels in the approximation sub-image. The texture information for a pixel can be extracted from a neighborhood of 3×3 pixels. Table 2 illustrates the concept. Hence, the texture unit can be labeled by using the following formula:

$$P_{TU} = \sum_{i=1}^8 E_i \times 3^{i-1}, P_{TU} \in \{0, 1, 2, \dots, 6560\} \quad (2)$$

where E_i is the i th element of the texture unit PTU . E_i can be derived from each neighborhood of 3×3 pixels by

$$E_i = \begin{cases} 0 & \text{if } P_i \leq (P_0 - \delta) \\ 1 & \text{if } (P_0 - \delta) < P_i \leq (P_0 + \delta) \\ 2 & \text{if } P_i > (P_0 + \delta) \end{cases} \quad (3)$$

where P represents the gray value of the central pixel, P_i denotes the gray value of the neighboring pixel i , and δ represents a small positive value. Based on the characteristic of hierarchical relationship of wavelet transform, the texture units of all pixels in the other detailed sub-images are labeled with respect to the related location of the approximation sub-image.

3. Perform averaging operation for all the neighborhoods of 3×3 pixels that belong to the same texture unit within each sub-image. Thus, an averaged standard sample of 3×3 pixels for each texture unit can be obtained. By performing the averaging operation over the whole sub-image with respect to the texture unit, the spectral and textural noise can be visibly removed.

Replace all the coefficients in the transformed image by the averaged standard values with respect to PTU . For further modification, nine neighborhoods or nine texture unit sets around a given pixel are also considered. For

Table 2. A 3×3 neighborhood mask of P

P_0	P_1	P_2
P_7	P	P_3
P_6	P_5	P_4

each standard sample set, the element that overlaps the original center pixel will be selected to contribute to the averaging operation of the nine elements. The process is illustrated in Figure 5.

Step 4: Local directional compensation

The local orientations of all pixels are estimated by using the predefined directional masks, so that the continuity of the ridge structure can be preserved by means of the relation with their neighboring regions. Thus, the process can eliminate the problem of broken ridges. In our evaluation, a modified method from the paper by Karu and Jain [3] is used to find the orientation of all pixels. The orientation of the pixel can be estimated by using the following formula:

$$D = \begin{cases} \text{Direction}(S_{\max}) & \text{if } (8C + S_{\max} + S_{\min}) > \frac{3}{16} \sum_{i=1}^{16} S_i, \\ \text{Direction}(S_{\min}) & \text{otherwise,} \end{cases} \quad (4)$$

where C denotes the gray value of center pixel and S_i represents the sum of the gray values located on the i th direction. $S_{\max}$ and $S_{\min}$ are, respectively, calculated as

$$S_{\max} = \max_{i=1, \dots, 16} S_i \quad (5)$$

$$S_{\min} = \min_{i=1, \dots, 16} S_i \quad (6)$$

In our implementation, in order to reduce the computation, 4 predefined directional masks are used. The four directional masks are illustrated in Figure 6. The orientation of each pixel can be calculated by Eq. (4). These ori-

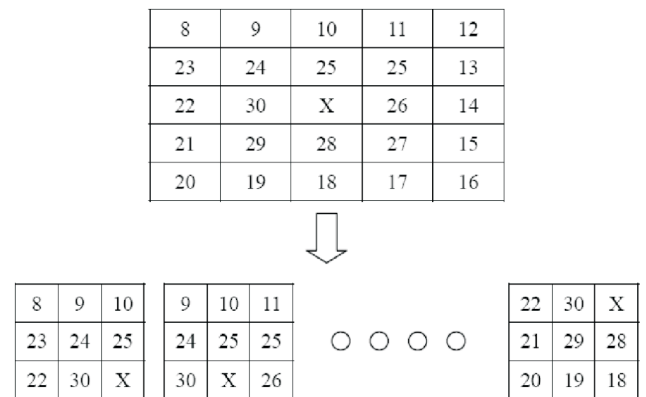


Figure 5. Nine considered neighborhoods and the nine standard values used for the average calculation.

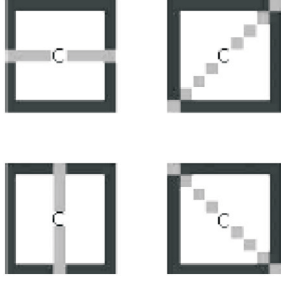


Figure 6. The 4 pre-defined directional masks.

orientations are sensitive to noise and, consequently, a smoothing technique is needed for constructing a robust enhancement algorithm. Based on the characteristic of fingerprint, it is obvious that most of the pixels in a local neighborhood have the same orientation. Thus, the voting mechanism is used in this chapter to modify the orientation of pixels that are corrupted by the scars. After smoothing the orientation of all pixels in the approximation sub-image, the pixels located in the specific direction as shown in Figure 8 are sorted by their wavelet coefficients. Then the value of the center pixel is replaced by the median value of these sorted coefficients. All detail sub-images derived from the wavelet transform are also modified with reference to the approximation sub-image just as the way described before. By using this technique, the broken ridge structure can be effectively reconditioned.

Step 5: Wavelet reconstruction

When all the sub-images are modified by the technique of texture filtering and directional compensation, the final improved fingerprint image is obtained by using the reconstruction process. Finally, we employ the thinning algorithm proposed by Zhou et al. [13] for thinning the enhanced images for the satisfactory performance and reasonable computational time. The results of the above steps are shown in Figure 7. Once the thinning process is completed, the minutiae can be extracted from the thinned image by using the Crossing Number (CN) on AFIS/AFRS. For the fingerprint classification, it is not necessary to perform another enhancement just for the classification. So we derive the orientation field from the thinned image directly to reduce the computation of system with large database. The method of orientation field computation is described in next subsection.

2.2 Orientation Field Computation

Although in the image enhancement of above session we can get the directional information on each pixel, the enhance step just use the 3×3 block to compute the directional field and there is only four direction on each pixel. For more detail on the direction field, we use a simple method to get the directional information from the thinned image. Because the thinned image has shown the clear characteristic of ridges and valleys on fingerprints, we use a 5×5 mask to compute the orientation following the skeleton of thinned image. The orientation is quantified by the 5×5 mask to 8 directions, S_1 to S_8 , as shown in Figure 8, there are $0(\pi)$, $\pi/8$, $\pi/4$, $3\pi/8$, $\pi/2$, $5\pi/8$, $3\pi/4$ and $7\pi/8$, respectively. And the pixels of each block belong to the direction D_p as

$$D_p = \text{Direction} (\max_{i=1, \dots, 8} S_i) \quad (7)$$

and

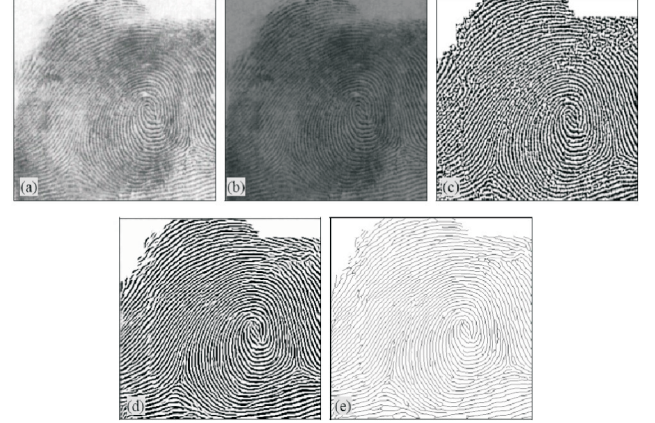


Figure 7. Results of the enhancement, (a) original fingerprint image, (b) normalized image, (c) the result of global texture unit filtering, (d) the result of local directional compensation, (e) thinned image.

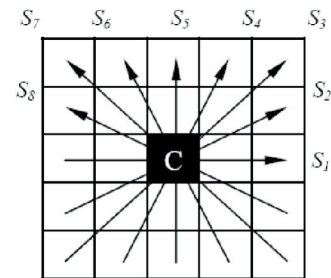


Figure 8. The 8 directions 5×5 mask represented, C is the center pixel of the mask.

$$S_i = \sum_{(x,y) \in M} p_i(x,y) \times c(x,y) \quad (8)$$

where M is the 5×5 mask, (x, y) is all the pixels in the mask, $p_i(x, y)$ is the gray value of pixel (x, y) in the i th directional mask and $c(x, y)$ is the gray value of the corresponding pixel (x, y) in the thinned image, respectively. When the computation is completed, each pixel near the skeleton will be given a directional value that they belong to. Figure 9 shows the examples of the processing. Then we divide the image into $(\text{height}/5) \times (\text{width}/5)$ blocks and assign a directional value D_b to the block by

$$D_b = \text{Direction} \left(\frac{\sum_{(x,y) \in \text{block}} \text{Direction}(x,y)}{N} \right) \quad (9)$$

where N is the number of the pixels with direction values in the block.

At this stage, the orientation field has been estimated but is still noisy, we need to smooth it and interpolate values for the noisy region. We use a 5×5 mask to smooth and interpolate the field by replacing the directional value of center point with the directional value that appears in the mask for most times. After the smoothing,

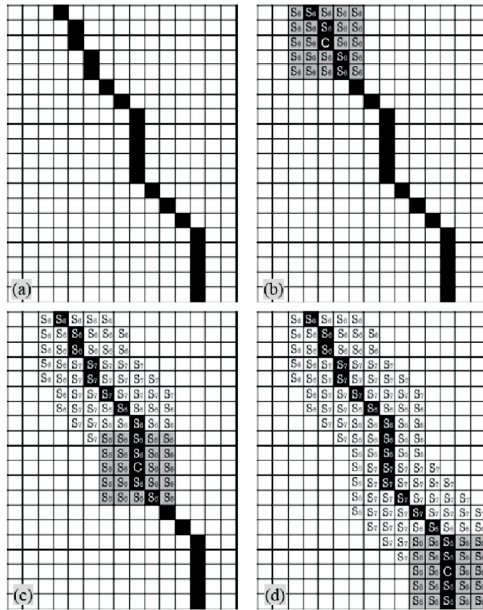


Figure 9. The example processing of the orientation estimation, (a) the partial skeleton of the thinned image, (b), (c), (d) the sequence of processing.

the orientation field is ready for the extraction of the feature for classifying. The result is shown in Figure 10. Figure 11 shows the comparison between the original image and the orientation field.

3. Feature Extraction

In the stage of feature extraction, the Poincare index, which is derived from continuous curves, is the most popular approach for the singularities based classification [1,3,5,6]. However, as mentioned above, the approach is especially hard to deal with the noisy or dim images that lose the information about the regions of singularities. In our method, we find a center of a similar circle from the orientation field and extract the features around the center for classification. We also use the Poincare index but only for detecting that whether the region of interest includes any singular points or not, and the computation will be addressed in a later section. In

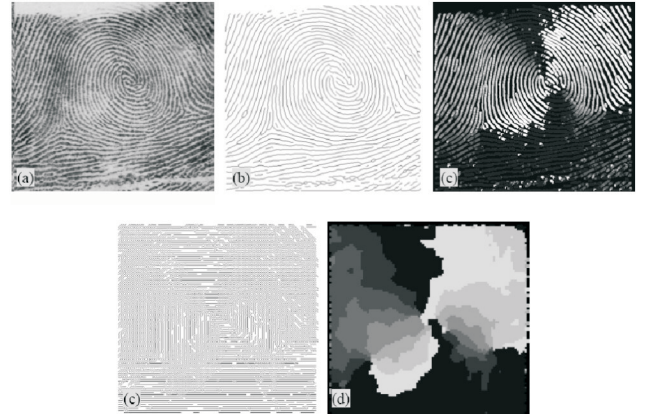


Figure 10. The result of computation by the 5×5 mask, (a) the original image, (b) the thinned image, (c) the result of the first stage computation, (d) the orientation field, (e) the smoothed orientation field.

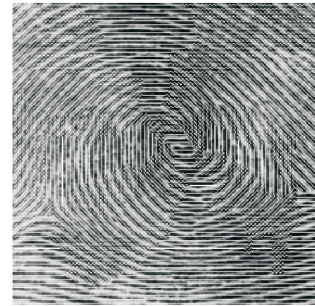


Figure 11. The comparison between the original image and the orientation field.

the next subsection, we first make a discussion on the structures of each class of fingerprints and then describe the detail of the feature extraction.

3.1 The Discussion on the Structures of Each Class

Before extracting the feature, we have to know what feature is helpful to classification and which feature has the most discriminability for fingerprint pattern. So, here we make the discussions about the regular characteristic of each fingerprint class first. We divide the region around the center into four blocks, left upper, left lower, right upper and right lower, and we discuss the directional features on each block for 4-class problem below:

- Arch (and tented arch): the orientations of texture around the center are symmetrical and the orientations of each block are all downward and outward. (see Figure 12)
- Left loop: the texture orientations of the two blocks above the center are symmetrical and downward, and the orientations of the two lower blocks are both towards the left. (see Figure 13)
- Right loop: the texture orientations of the two blocks above the center are symmetrical and downward, and the orientations of the two lower blocks are both towards the right. (see Figure 14)
- Whorl: the texture orientations of the left and right blocks are symmetrical each other and downward,

and the orientations of the blocks above the center and that are under the center are outwards and inwards, respectively. (see Figure 15)

By observing the appearances described above, we could draw the conclusion that the orientations of the blocks above the center are consistent in that they are all symmetrical and are downward and outward and the orientations of the blocks under the center have the discriminability of each fingerprint class.

Hence, we use the information above the center to locate the center point of the region of interest and extract the feature from the information under the center point to classify them. The detailed process is described in the next subsection.

3.2 Center Point Location

After we accomplished the computation of the orientation field, we still have to determine the region of interest (ROI) to extract the most representative features for classification. Locating the center point of the region of interest is an essential step that can influence the accuracy of the classification. For the different classifier, the method used to locate the center point is different, too. Wang et al. [14] used the Poincare index to locate the core of singular point as the center of the interest region and extract the gray values of the 7 by 7 blocks around

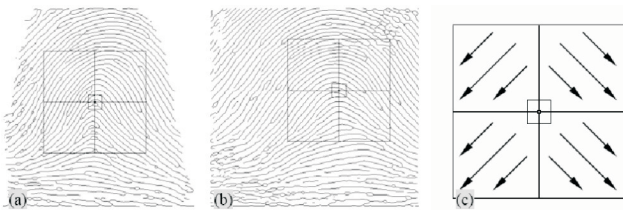


Figure 12. The orientation characteristic of arch class, (a), (b) the example patterns, (c) the orientations of four blocks around the center.

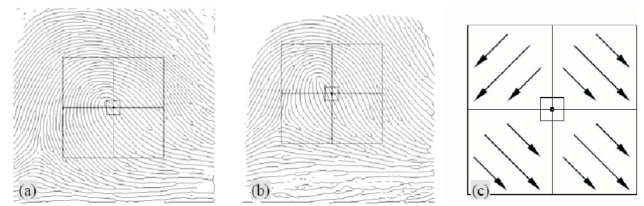


Figure 14. The orientation characteristic of right loop class, (a), (b) the example patterns, (c) the orientations of four blocks around the center.

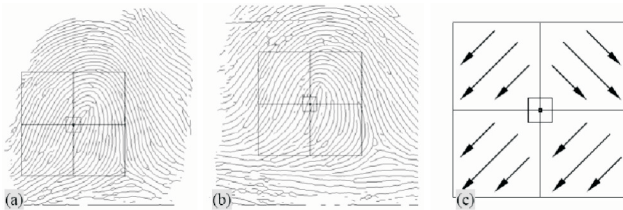


Figure 13. The orientation characteristic of left loop class, (a), (b) the example patterns, (c) the orientations of four blocks around the center.

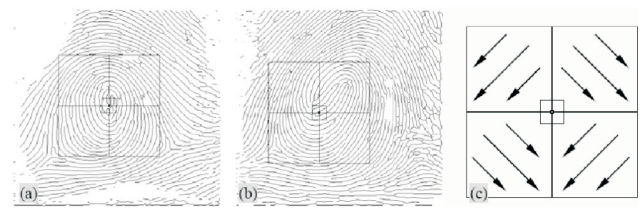


Figure 15. The orientation characteristic of whorl class, (a), (b) the example patterns, (c) the orientations of four blocks around the center.

the center as the feature vectors for classification. Besides, against the drawbacks about the Poincare index mentioned above, Sha et al. [15] present a method using the relationship between one point and all points of the image to find the point similar to a center of a circle, but it is heavy to compute the relationship of each pixel in the whole image. Another approach using the intersection points of the boundaries of each direction in the orientation field to determine the singular points is presented by Rusyn and Ostap [16], but the method is unable to differentiate between the cores and the deltas.

In this subsection, we present a fast method to search and locate the center of the region of interest using an octagon mask. In the octagon mask, for the region of each triangle we assign different score with different conditions for the corresponding pixels of the orientation field. As shown in Figure 9, the mask we used is designed by considering the similar structure of a circle and the values 1, 3, 5 and 7 represent the direction S_1 , S_3 , S_5 and S_7 respectively, as shown in Figure 16. We attempt to find the point that is a similar center of a circle from the orientation field of fingerprints, so that the summation of the correlation between the mask and a pattern, if that is an ideal circle, will be the highest. And considering the symmetric of center of fingerprints as mentioned above, we assign different weight for each triangle of the octagon. We increase the weight of upper and both side triangles to emphasize the characteristic of the center point, and the weight of each triangle is shown as Figure 17.

The center point location algorithm includes following steps:

Step 1: Let D be the orientation field with $D(i, j)$ representing the direction at pixel (i, j) .

Step 2: Define an octagon mask M including four direc-



Figure 16. The mask for capturing the center point; X denotes the evaluated position of possible center point.

tions and eight weights.

Step 3: Initialize a label image L with the same size of D . L is used to indicate the center point of the region of interest.

Step 4: For each pixel (i_c, j_c) in D , define a local region S with the same size of M . The values of summation $L(i_c, j_c)$ can be estimated by using the following formula:

$$L(i_c, j_c) = \sum_{(i,j) \in M, (i_m, j_m) \in S} |dir(i, j) - dir(i_m, j_m)| \times weight \quad (10)$$

where $dir(i, j)$ is the direction at pixel (i, j) in M and $dir(i_m, j_m)$ is the direction at pixel (i_m, j_m) in S . The weight is defined as Figure 17.

Step 5: Search for the minimum value and assign the coordinates as the center point of the region of interest. If there are two equal minimum values in L , we will assign the coordinates of the middle point of them as the center point.

Figure 18 shows the result of the center point location and Figure 19 is the result of location on noisy or dim images. We could note that for the stained images the algorithm still works well.

3.3 Feature Extraction

As mentioned in section 3.1, the orientation under the center point can most demonstrate the characteristics of each fingerprint class most, so we extract the directional feature of the lower portion of the octagon for classification. Figure 20 shows the region we extract the feature from.

We determine the similarities of each direction in triangles to deduce the feature as the input of classification by using the following formulas:

In the left half the L -score is estimated by

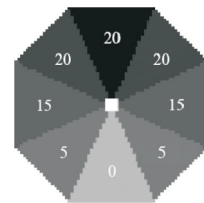


Figure 17. The weight of each triangle of the octagon.

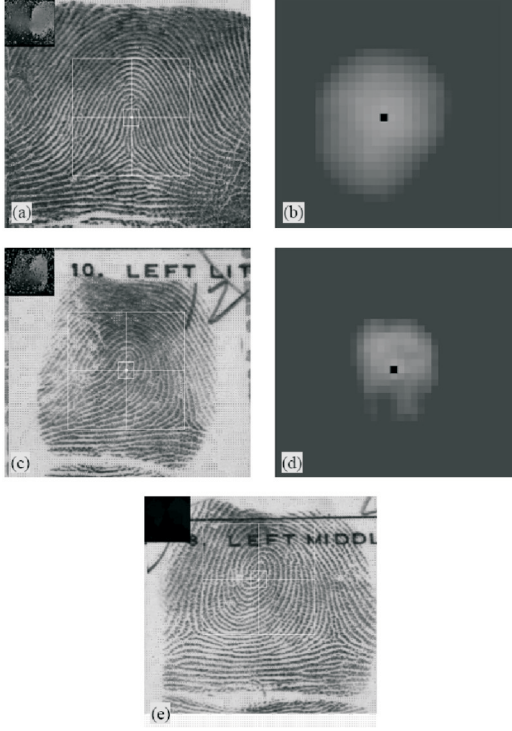


Figure 18. The results of the center point location, (a) the result of an arch fingerprint, (b) the label image of (a), (c) the result of a left fingerprint, (d) the label image of (c), (e) the result of a whorl fingerprint.

$$L-score = \sum_{i,j \in L_M} S_{i,j} \times weight \begin{cases} S_{i,j} = 10 & \text{when } dir(i,j) = 3, \\ S_{i,j} = 5 & \text{when } dir(i,j) = 2 \text{ or } 4, \\ S_{i,j} = -10 & \text{when } dir(i,j) = 7, \\ S_{i,j} = -5 & \text{when } dir(i,j) = 6 \text{ or } 8, \\ \text{others, } S_{i,j} = 0, \end{cases} \quad (11)$$

In the right half the $R-score$ is estimated by

$$R-score = \sum_{i,j \in R_M} S_{i,j} \times weight \begin{cases} S_{i,j} = -10 & \text{when } dir(i,j) = 3, \\ S_{i,j} = -5 & \text{when } dir(i,j) = 2 \text{ or } 4, \\ S_{i,j} = 10 & \text{when } dir(i,j) = 7, \\ S_{i,j} = 5 & \text{when } dir(i,j) = 6 \text{ or } 8, \\ \text{others, } S_{i,j} = 0, \end{cases} \quad (12)$$

where $dir(i,j)$ is the direction of the corresponding pixel in the orientation field and there are also different

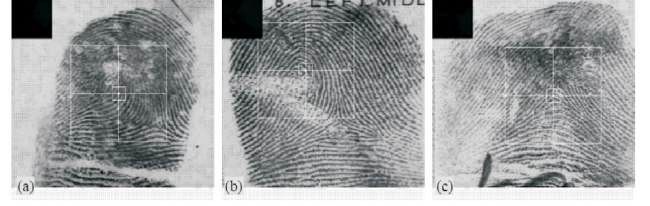


Figure 19. Some examples of the location on poor quality images, (a) a large number of information is lost, (b) the information of the core is lost, (c) the texture on fingerprint is stained.

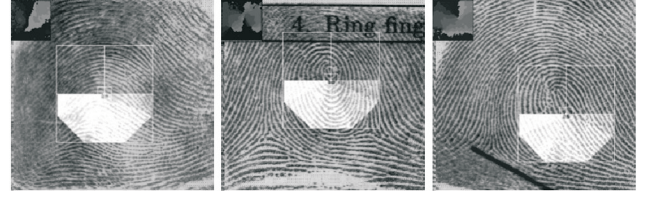


Figure 20. The regions we extract the features from.

weights in each triangle as shown in Figure 21. Finally, we can obtain a 2D feature vector ($L-score$, $R-core$) and then we use the vectors for classification.

4. Classification

Because the derived features describes the directional information on blocks (left lower and right lower), if there are more than one direction in a block, the features will not be able to show the clear feature with some direction. In other words, they will get a low score on the block that includes the singular points. For this, we detect the singular points in the block with the low score by Poincare index in first stage and classify them into one of three classes of right loop, left loop and whorl. For the others with the clear directional characteristics else, we classify them into one of four classes by the Quadrant of ($R-score$, $L-score$) in second stage. The flowchart of classification is shown in Figure 22.

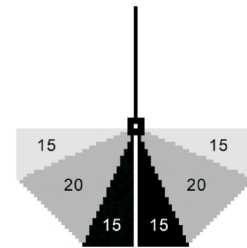


Figure 21. The weights for feature extraction.

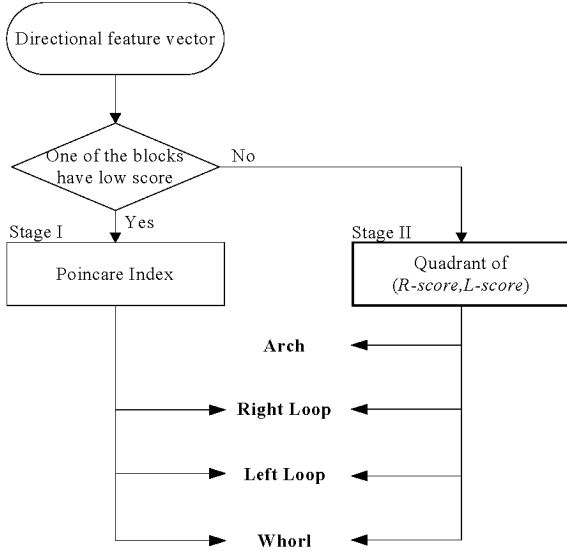


Figure 22. The flowchart of our fingerprint classification.

4.1 First Stage: Filtering the Image Including Singularities

We use the Poincare index to filter the images including singular points in the stage. As for fingerprints images, a double care point has a Poincare as 1 in a core point $1/2$ and a delta point $-1/2$. The Poincare index at pixel (x, y) , which is enclosed by a digital curve, the boundary of the blocks in the paper, can be computed as follows:

$$Poincare(x, y) = \frac{1}{2\pi} \sum_{k=0}^{N_{\Psi}} \Delta(k) \quad (13)$$

where

$$\Delta(k) = \begin{cases} \delta(k), & \text{if } |\delta(k)| < \frac{\pi}{2}, \\ \pi + \delta(k), & \text{if } \delta(k) \leq -\frac{\pi}{2}, \\ \pi - \delta(k), & \text{otherwise,} \end{cases} \quad (14)$$

and

$$\delta(k) = \Theta(\Psi_x(i'), \Psi_y(i')) - \Theta(\Psi_x(i), \Psi_y(i)) \quad (15)$$

$$i' = (i+1) \bmod N_{\Psi} \quad (16)$$

Θ is the orientation field, $\Psi_x(i)$ and $\Psi_y(i)$ denote the i th

point coordinate (x, y) of the enclosed curve Ψ .

There are three cases that the block probably includes the singular points, it is described as follows:

1. The case of left loop; when the loop feature is not apparent, the directional features are usually unclear due to the fact that some block includes the delta points as shown in Figure 23. Hence if we find that the directional feature towards the left in L -score is higher than a predefined threshold and a delta point detected by Poincare index in the right block, it is classified as the class of left loop at the first stage.
2. The case of right loop; the same as left loop, when the loop feature is not apparent, the directional features are usually unclear due to the fact that some block includes the delta points as shown in Figure 24. Hence if we find that the directional feature towards the right in R -score is higher than a predefined threshold and a delta point detected by Poincare index in the left block, it is classified as the class of right loop at the first stage.

The case of whorl; as shown in Figure 25 and Figure 26, when the texture of a whorl is large, the region of in-

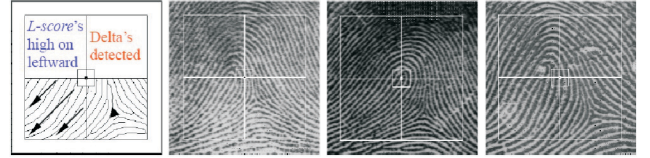


Figure 23. The left loop includes a delta point in the right block and the L -score is high leftward.

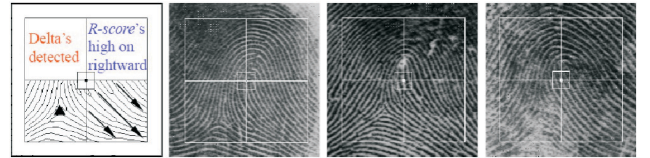


Figure 24. The right loop includes a delta point in the left block and the R -score is high on rightward.

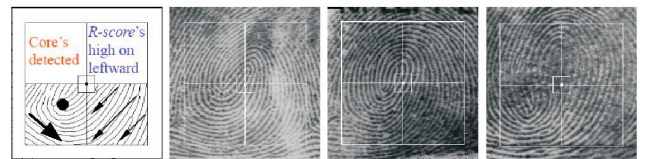


Figure 25. The whorl includes a core point in the left block and the R -score is high on leftward.

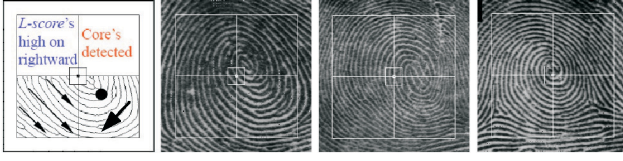


Figure 26. The whorl includes a core point in the right block and the L -score is high on rightward.

terest we derived in section 3.2 probably includes a singular point. Hence if we find that the directional feature towards the left in R -score or towards the right in L -score is higher than a predefined threshold and a core point detected by Poincare index in the corresponding block, it is classified as the class of whorl at the first stage. Furthermore, in a few situations there the core point will be replaced by a delta point and we also assign that as the class of whorl as shown in Figure 27.

4.2 Second Stage: Determining the Quadrant of (R -score, L -score)

At the second stage, first we define the sign of direction $S_3(\nearrow)$ as positive and the direction $S_7(\nwarrow)$ as negative. By that the coordinates of (R -score, L -score) will be mapped into the four quadrants of (x , y)-plane and as the mention in Section 3 we can separate the fingerprints easily into the four classes at the second stage. Figure 28 shows the four classes in the four quadrants. After the second stage classification by (R -score, L -score), it is all done for the fingerprint classification. The examples of our results are shown in Figure 29 and that shows the singular points detected at the first stage such as (c), (d), (f) and (g). And Figure 30 shows that the proposed method also can classify the noisy and dim images correctly. The letters A, R, L and W in these figures represent the class of arch, right loop, left loop and whorl, respectively.

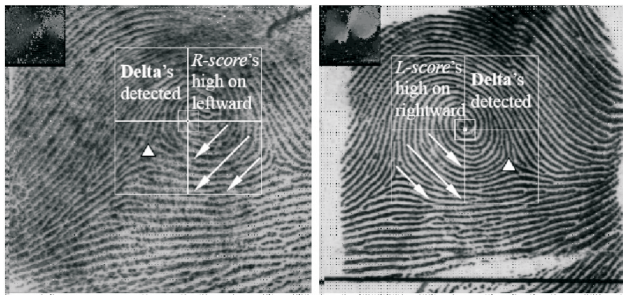


Figure 27. The situations that the core point is replaced by a delta in the case of whorl.

5. Experimental Results

In our experiment we use the NIST special fingerprint database 4 (NIST-4) that contains 4000 gray-scale fingerprints images from 2000 fingerprints, and they are distributed evenly over the five classes (arch (A), tented arch (T), left loop (L), right loop (R) and whorl (w)). The classification of each fingerprints are stored in the NIST IHead id field of each file. In our results of the experiment for the 4-class classification problem, the classes A and T are merged into one class and the class index of one image in the NIST-4 database does not necessarily belong to only one class. So we consider our result of the

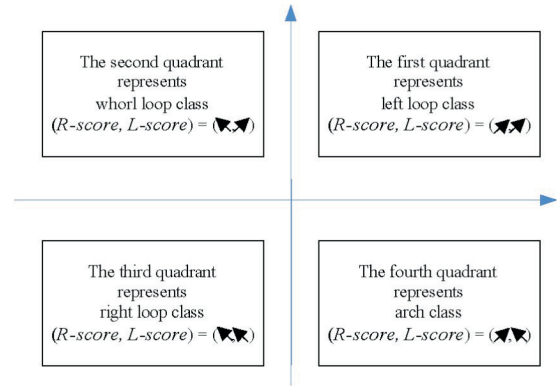


Figure 28. The four classes in the four quadrants.

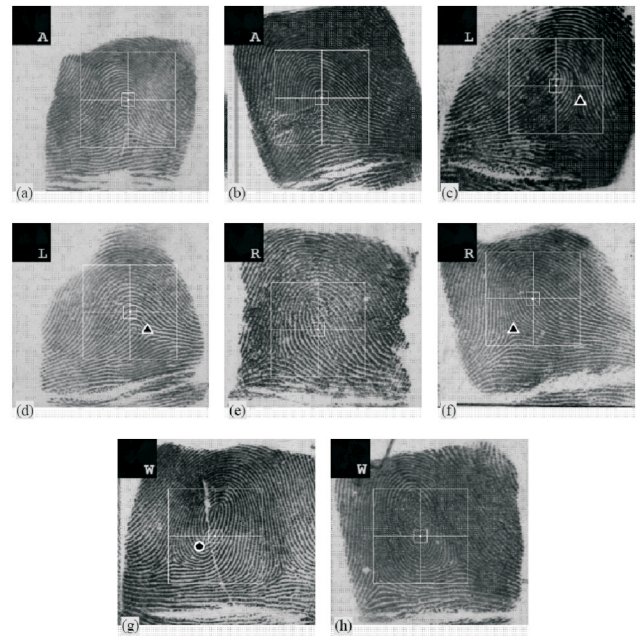


Figure 29. The results of fingerprint classification, and there are the singular points in (c), (d), (f) and (g).

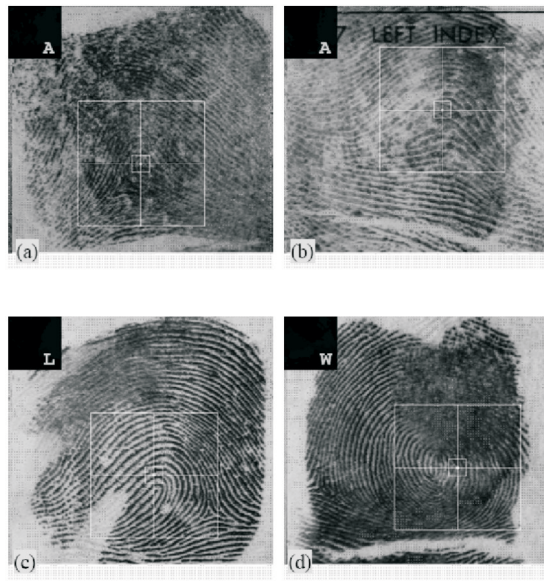


Figure 30. The results of the images with poor quality.

classification to be correct if it matches the one of the class index. This approach has also been adopted by other researchers [1,3]. We test our algorithm with all 4000 fingerprints in NIST-4 data base and the results are shown in Table 3 where the first column shows the class index in the image file header and the first row labels the assigned class using our method. The accuracy is 93.425% (3737/4000) with zero rejection in our system.

From the result table we can see that most of the misclassifications are between the loops and the whorls. The main reason is due to the rotation of loops and the similar pattern between loops and arch, and some of the misclassifications are due to the limitation of the mask size used to extract features; so that the mask cannot include the helpful features like singular points from the images with the larger pattern. Besides, some images have the wrong center of the region of interest located at first stage because of the poor image quality. (See Figure 31)

Table 3. Our experimental results for 4-class problem on NIST-4 database with no rejection

True	Exp. as				
	Whorl	Left loop	Right loop	Arch	Unknown
Whorl	752	19	8	1	0
Left loop	7	674	10	71	0
Right loop	5	4	801	76	0
Arch	4	32	26	1518	0

We also compare the results with the traditional method of singularity based on the same database as shown in Table 4. We can see that the major difference is that the number of the images that cannot be classified by the traditional classifier is up to 254 and the whorl type is up to 127. So the accuracy of the system with singularity based is only 87% for 4-class problem on the same database. There is a complementary texture for the classification of the whorl type using the pseudo ridge tracing presented by Zhang and Yan [17] in recent research and their classification accuracy reaches 92.7% for the 4-class problem on NIST-4 database without rejection as compared in Table 5.

However, there are still many poor quality images, like stained and dim images, on NIST-4 database and they are difficult to recognize even for human. If we discard such obscure images by introducing the rejection scheme, the accuracy of our classification will rise to 93.9% with 1.8 reject rate, 95.8% with 8.5% reject rate and 96.8% with 19.5% reject rate. Figure 32 shows the corresponding graph of the rejection rate and Figure 33

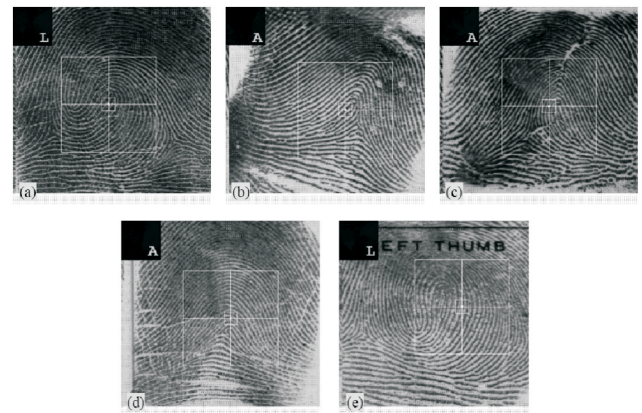


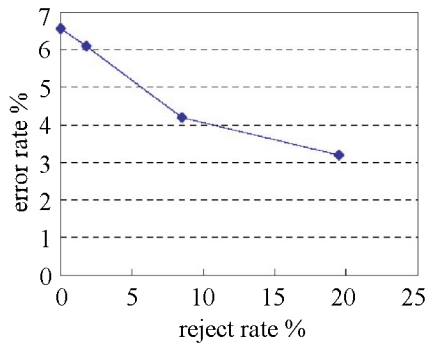
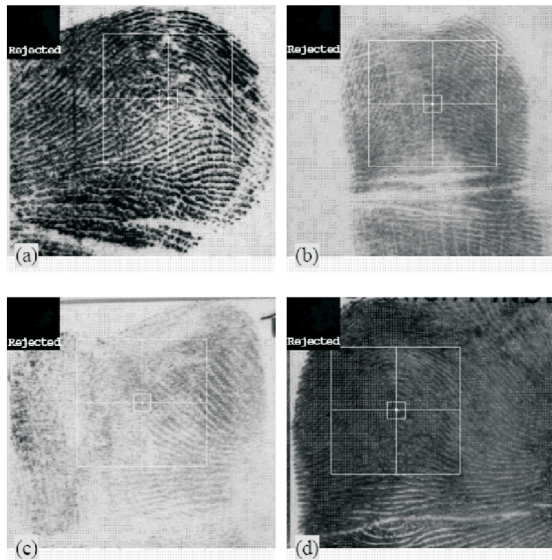
Figure 31. The misclassified images, (a) a whorl as a left loop, (b) a left loop as an arch, (c) a right loop as an arch, (d) a right loop as an arch, (e) a whorl as a left loop.

Table 4. The experimental results of traditional singular based method for 4-class problem with no rejection

True	Exp. as				
	Whorl	Left loop	Right loop	Arch	Unknown
Whorl	602	20	23	10	125
Left loop	24	669	6	35	50
Right loop	23	10	689	46	36
Arch	18	33	19	1519	43

Table 5. The experimental results with pseudo ridge tracing [17] for 4-class problem with no rejection

True	Exp. as				
	Whorl	Left loop	Right loop	Arch	Unknown
Whorl	727	27	26	8	7
Left loop	25	759	5	26	2
Right loop	24	14	740	41	4
Arch	22	38	18	1482	5

**Figure 32.** Four-class error-reject plots of the NIST-4 data base.**Figure 33.** The examples of the rejected images, (a) the noisy image, (b), (c) the faint images, (d) the over-inked image.

shows the examples of the rejected images with poor quality.

Finally, the average computational expense of 1000 fingerprints is recorded at PC Pentium 4 2.6 G and the result is shown in Table 6. We can note that the total time is

Table 6. The computational expense of our system

Procedure	Expense (s)
Image enhancement [10]	0.58
Orientation field computation	0.08
Feature extraction	0.22
Classification	0.0002
Total	0.8802

less than one second and the system only spends about 0.3 second on the procedures of the classification. In another word, if the classification is built on an automatic fingerprint identification system, it will have an effective classification only with little additional load.

6. Conclusion

We have proposed an effective method for the fingerprint classification, which uses both the orientation characteristic and singular points around the center of the region of interest. As the tables shown above, the results of the system test on NIST-4 database are encouraging. But there is still a lot more that can be done to further improve the results especially on the rotation and the obscure similar pattern.

Acknowledgements

This work was supported by the National Science Council under grant number NSC 94-2213-E-032-020.

References

- [1] Jain, L. C., Halici, U., Hayashi, I., Lee, S. B. and Tsutsui, S., *Intelligent Biometric Techniques in Fingerprint and Face Recognition* CRC Press, Boca Raton, FL, pp. 355–398 (1999).
- [2] Henry, E. R., *Classification and Uses of Finger Prints*, London: Routledge (1900).
- [3] Karu, K. and Jain, A. K., “Fingerprint Classification,” *Pattern Recognition*, Vol. 29, pp. 389–404 (1996).
- [4] Jain, A. K., Prabhakar, S. and Hong, L., “A multichannel Approach to Fingerprint Classification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions*, Vol. 21, pp. 348–359 (1999).
- [5] Meltem, B., Sakaraya, F. A. and Evans, B. L., “A Fin-

- gerprint Classification Technique Using Directional Image," *IEEE Asilomar Conf. on Signals Systems and Computers* (1998).
- [6] Kawagoe, M. and Tojo, A., "Fingerprint Pattern Classification," *Pattern Recognition*, Vol. 17, pp. 295–303 (1984).
- [7] Wilson, C. L., Candela, G. T. and Watson, C. I., "Neural Network Fingerprint Classification," *J. Artificial Neural Networks*, Vol. 1, pp. 203–228 (1993).
- [8] Watson, C. I. and Wilson, C. L., "NIST Special Database 4, Fingerprint Database," *Nat'l Inst. of Standards and Technology*, Mar. (1992).
- [9] Candela, G. T., Grother, P. J., Watson, C. I., Wilkinson, R. A. and Wilson, C. L., "PCASYS-a Pattern-Level Classification Automation System for Fingerprints," *Technical Report NISTIR 5647*, Apr. (1995).
- [10] Senior, A., "A Hidden Markov Model Fingerprint Classifier," *Proc. 31st Asilomar Conf. Signals, Systems and Computers*, pp. 306–310 (1997).
- [11] Hsieh, C. T., Lai, E. and Wang, Y. C., "An Effective Algorithm for Fingerprint Image Enhancement Based on Wavelet Transform," *Pattern Recognition*, Vol. 36, pp. 303–312 (2003).
- [12] Daubechies, I., "The Wavelet Transform, Time – Frequency Localization and Signal Analysis," *IEEE Trans. Inf. Theory*, Vol. 36, pp. 961–1005 (1990).
- [13] Zhou, R. W., Quek, C. and NG, G. S., "Novel Single-Pass Thinning Algorithm," *Pattern Recognition Letter*, Vol. 16, pp. 1267–1275 (1995).
- [14] Wang, S., Zhang, W. W. and Wang, Y. S., "Fingerprint Classification by Directional Fields," *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces. (ICMI'2002)*, p. 395 (2002).
- [15] Sha, L., Zhao, F. and Tang, X., "Improved Fingercode for Filterbank-Based Fingerprint Matching," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, pp. 895–898 (2003).
- [16] Rusyn, B. and Ostap, O., "Estimation of Singular Points in Fingerprints Image," *TCSET'2002*, Ukraine, Feb. 18–23, p. 236 (2002).
- [17] Zhang, Q. and Yan, H., "Fingerprint Classification Based on Extraction and Analysis of Singularities and Pseudo Ridge," *Pattern Recognition*, Vol. 37, pp. 2233–2243 (2004).

Manuscript Received: Mar. 12, 2007

Accepted: Apr. 28, 2008