

Fingerprint image Pre- post processing Methods for minutiae extraction.

Asst. Prof. Dr.Fayadh. M. Abed College of Computer and
mathematics science. Tikrit University.

Dr.Adnan Maroof College of Computer and mathematics science.
Tikrit university

Abstract

A set of reliable techniques is presented in this paper. These techniques include enhancement and minutiae extraction that involve the clarity of the ridge and valley structures of the input fingerprint images based on the frequency and orientation of the local ridges and thereby extracting correct minutiae.

Fingerprint image enhancement algorithms are evaluated and implemented to understand how the enhancement algorithm works and how well it is. Two types of features are presented in this job. (i) global ridge associated with central region of the fingerprint and (ii) minutiae associated with local ridge and furrows structure. Extraction of minutiae, involves a series of image enhancement such as normalization, ridge orientation estimation, filtering, thresholding, thinning and False minutiae removal. After applying image enhancement and minutiae extraction techniques on a set of images, we took as a sample, the results was perfect. The minutiae has been removed and a results, we got was in better condition.

طرق معالجة بصمة الإصبع في استخلاص التفاصيل الدقيقة

في هذا البحث استخدمت مجموعة من التقنيات لاستخلاص التفاصيل الدقيقة لبصمة الإصبع، وتتضمن مرحلتين التحسين والاستخلاص والتي تهدف إلى إظهار ووضوح هيكلية بصمة الإصبع (الحافات والمناطق المدصورة بينها). التقنية تعتمد على حساب التردد والاتجاهات ومن ثم استخلاص الدقائق المطلوبة. خوارزمية تحسين صورة بصمة الإصبع استخدمت وتم برمجتها لفهم كيفية عمل وتحسين الصورة. في هذا البحث قدمت نوعين من الخصائص في تصنيف واستخلاص لبصمة الإصبع الأولى الحافات الرئيسية مع مركز البصمة والدقائق المرتبطة في الحافات والشقوق والثانية استخلاص الدقائق لبصمة الإصبع، حققت الدراسة نسبة نجاح جيدة. استخلاص الدقائق لبصمة الإصبع الدراسة حققت نسبة شملت سلسلة من عمليات تحسين الصورة (تحويل الصورة الملونة إلى صورة ثنائية، تحديد اتجاهها، المرشحات، تحديد العتبة، ومن ثم التنحيف) وأخيراً أزاله الشوائب. النتائج المستنبطة من استخدام تقنيته تحسين الصورة وإزاله الدقائق الغير مرغوب بها حققت نتائج جيدة.

الاستاذ المساعد الدكتور فياض محمد عبد الدليمي :كلية الحاسبات والرياضيات -

جامعه تكريت

الدكتور عدنان عبد القادر معروف الياسين : كلية الحاسبات والرياضيات- جامعه

تكريت

Nomenclatures:

B : Pixel value

$C(x, y)$: Coherence Image

CB : Segmentation Mask

$I(i, j)$: fingerprint Image

i, j : Index

K : Index of local window

M : no. of pixel in x-direction

$M(k)$: mean grey-level value of block k

$M0$: Mean desired value of the image

N : no. of pixel in y-direction

$N(I, j)$: Image Normalization

$M0, V0$ are set to empirical values 100.

$V0$: Variance desired value of the image

$V(k)$: The variance for block k

W : Size local window

θ : Minutiae Orientation

Abbreviation

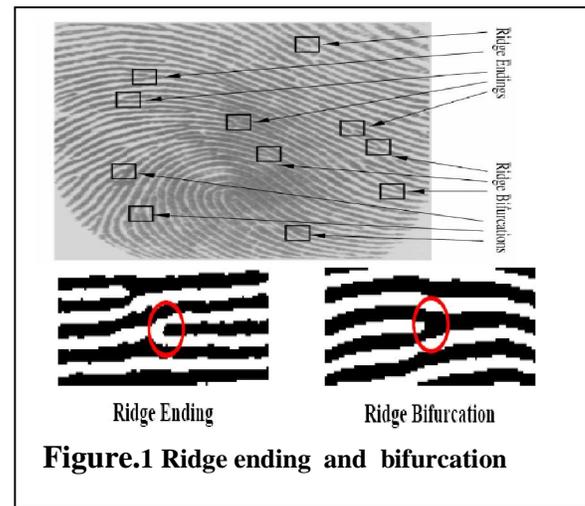
Bif.: Bifurcation

CN: Cross Number

Term: Termination

1. Introduction

Fingerprint-based identification is one of the biometric techniques we use a lot nowadays. A fingerprint is a series of ridges and furrows on the surface of the finger [1], where a ridge is a single curved segment, and a valley is the region between two adjacent ridges. Minutiae points are these local ridge characteristics that occur either at a ridge ending or bifurcation as shown in Fig.1. A ridge ending is the point where the ridge ends abruptly, however the ridge bifurcation is the point where the ridge splits into two or more branches. It is hard to detect the minutiae due to the low quality in fingerprint images. This study shows a set of reliable techniques to solve this problem. The process is summarized as follows: Fingerprint normalization, Local Ridge Orientation extraction, Fingerprint filtering, Image Thresholding, Image Thinning, Minutiae Extraction and False Minutiae Removal.



2. Fingerprint Representation

Fingerprint representations rely on locations of the minutiae and the directions of ridges at the minutiae location. It includes information about locations of critical points (e.g., core and delta). This study covers local and global representations in addition to vector drawing. Representations predominantly based on ridge endings or bifurcations (collectively known as minutiae (see Figure 1) are the most common, primarily due to the following facts: (i) minutiae capture much of the individual information, (ii) minutiae-based representations are storage efficient, and (iii) minutiae detection is relatively robust to various sources of fingerprint degradation. The steps taken in this study are Pre-Processing, Minutiae extraction and Post-Processing. The preprocessing is shown in Figure 2.0

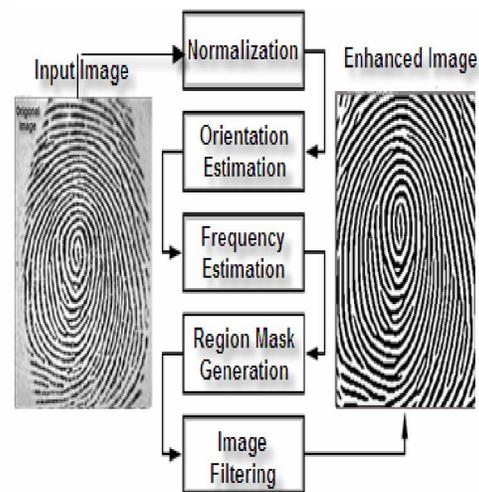


Fig.2 Fingerprint Pre-Processing Diagram(Enhancement)

3. Fingerprint Recognition

In this study we will concentrate on the steps taken for image enhancement and minutiae extraction.

3.1 Fingerprint Enhancement processing (Segmentation and normalization)

Variance threshold is used in segmentation process. Two steps we should take for enhancement (segmentation and normalization). In segmentation the block variance is calculated according to the following formula[2]:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=1}^{W-1} (I(i, j) - M(k))^2 \quad (1)$$

Let t be the mean threshold. If $v(k)$ is less than threshold (t) the block is deleted.

Normalization is defined as a function of variance as shown in Eq.2

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i, j) - M)^2}{V(I)}}, & \text{if } I(i, j) > M \\ M_0 - \sqrt{\frac{V_0(I(i, j) - M)^2}{V(I)}}, & \text{Otherwise} \end{cases} \quad (2)$$

The mean and variance of gray level fingerprint image I , are defined as

$$V(I) = \frac{1}{N * M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I(i, j) - M(I))^2, \quad (3)$$

$$M(I) = \frac{1}{N * M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I(i, j),$$

3.2 Coherence

Coherence is a measure of how well the gradients of the fingerprint image are pointing in the same direction. In a window of size $W \times W$ around a pixel, the coherence is defined as[3]:

$$Coh = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \quad (4)$$

Where

$$G_{xx} = \sum_w G_x^2, G_{yy} = \sum_w G_y^2, G_{xy} = \sum_w G_x G_y$$

A captured fingerprint image usually consists of two components, the *foreground* and the *background*. The foreground is the component originated from the contact of a fingertip with the sensor. The noisy area at the borders of the image is called the background. The part of the image that belong to the foreground or to the background is determined by the segmentation algorithm.

Since a fingerprint mainly consists of parallel line structures, the coherence will be considerably higher in the foreground than in the background. For most fingerprint sensors, the ridge-valley structures can be approximated by black and white lines, while the background, where the finger does not touch the sensor, is rather white. This means the mean gray value in the foreground is in general lower, i.e. darker gray, than it is in the background while the variance of the ridge-valley structures in the foreground is higher than the variance of the noise in the background.

The resulting gradient coherence image is smoothed by the use of a Gaussian smoothing filter to give a coherence image. The smoothed image is then binarized to produce a segmentation mask. The binarization is carried out by global thresholding as:

$$C_B(i, j) = \begin{cases} 1 & C(i, j) > M_c - 0.5S_c \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

M_c is the global mean of the coherence image and S_c is its global standard deviation. Holes in the segmentation mask are removed using morphological post processing, This mask on pixel-wise multiplication with the fingerprint image gives the segmented image. coherence and segmentation mask images are shown in Fig.3 .

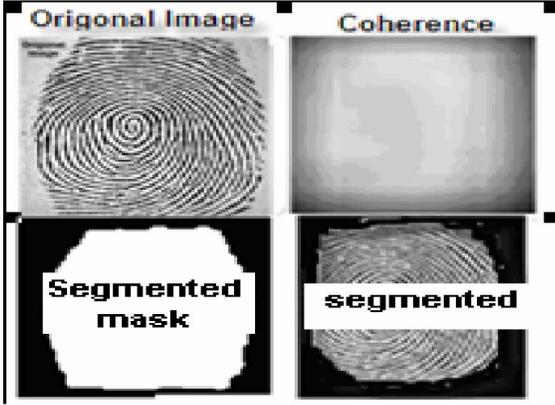


Fig.3 Image segmentation steps

4, Estimation of Orientation Field

A major common step in both core point detection and finger print matching is the orientation or direction of the ridge estimation. The orientation field of a fingerprint image represents the directionality of ridges in the fingerprint image

A number of methods have been proposed to estimate the orientation field of fingerprint images [4,5]. The first step of the ridge extraction algorithm is to estimate the local orientation field according to the following steps:

- Divide the input fingerprint image into blocks of size $w \times w$.
- Compute the row and column gradient (Sobel operator is used) at each pixel in each block [6].
- Estimate the local orientation at each pixel using the following algorithm

A pixel orientation map is computed according to the reports of [7]. For pixel (i, j) , we apply Sobel operators to get

$$G_x = B(i+1, j-1) + 2B(i+1, j) + B(i+1, j+1) \\ - B(i-1, j-1) - 2B(i-1, j) - B(i-1, j+1) \\ G_y = B(i-1, j+1) + 2B(i, j+1) + B(i+1, j+1) \\ - B(i-1, j-1) - 2B(i, j-1) - B(i+1, j-1)$$

$$V_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2G_x(u, v)G_y(u, v),$$

$$V_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (G_x^2(u, v) - G_y^2(u, v)) \quad (6)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_x(i, j)}{V_y(i, j)} \right)$$

where $\theta(i, j)$ is the least square estimate of the local orientation at the block centered at pixel (i, j) .

d. Smooth the orientation field in a local neighborhood using a Gaussian filter. The orientation image is converted first into a continuous vector field, which is defined as:

$$\Phi_x(i, j) = \cos(2\theta(i, j)), \Phi_y(i, j) = \sin(2\theta(i, j))$$

where Φ_x and Φ_y are the x and y components of the vector field, respectively. After the vector field has been computed, Gaussian smoothing is then performed as follows:

$$\Phi'_x(i, j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} G(u, v) \Phi_x(i - uw, j - vw),$$

$$\Phi'_y(i, j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} G(u, v) \Phi_y(i - uw, j - vw)$$

where G is a Gaussian low-pass filter of size $w_\Phi \times w_\Phi$.

e. The final smoothed orientation field (*Orient*) at pixel (i, j) is defined as:

$$Orient(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\Phi'_y(i, j)}{\Phi'_x(i, j)} \right)$$

this process is shown in Fig.4.

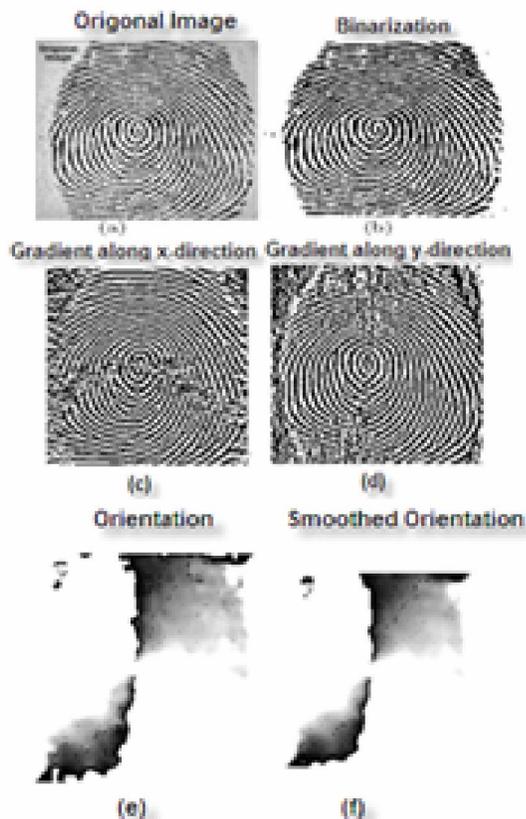


Fig.4 Estimation of image orientation

5. Minutiae Extraction

Minutiae Pattern Detection

From a thinned image, we can classify each ridge pixel into one of the following categories according to its 8- connected neighbors. A ridge pixel is called : An *isolated point* if it does not have any 8-connected neighbor,

- an *ending* if it has exactly one 8-connected neighbor,
- An *edgepoint* if it has two 8-connected neighbors,
- A *bifurcation* if it has three 8-connected neighbors, and
- A *crossing* if it has four 8-connected neighbors.

Two kinds of minutia pixels, either ending or bifurcation, are first located by checking the 8 neighbors of a ridge pixel. Figure 5c. shows candidates of minutia pixels in a ridge map. Due to broken ridges, hair effects, and ridge

endings beyond the margins, a post processing [8] is used to remove the following spurious minutia pixels with the result shown in Figure 5(d).

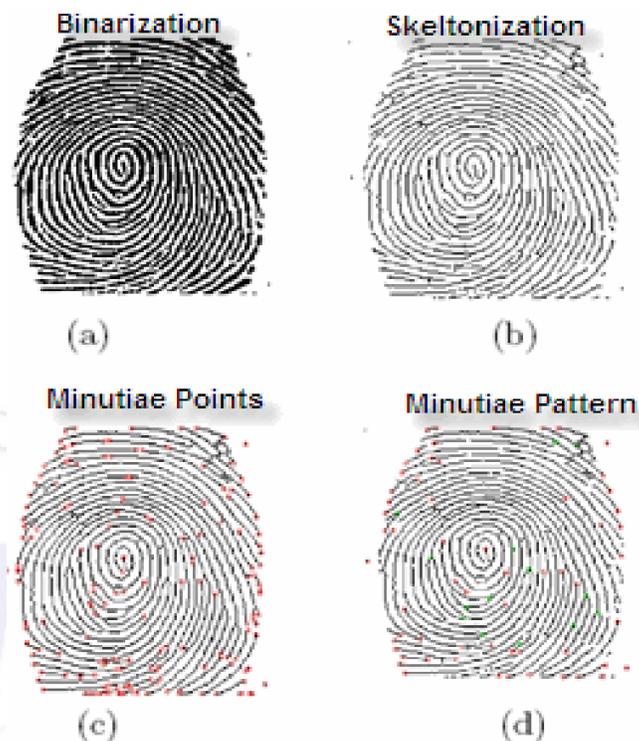


Fig.5 Minutiae Extraction

Minutiae Pattern

Spurious minutia pixels include endings that lie on the margins of the region of interest, two closest endings of the same ridge orientation, ridge ending and bifurcation are connected and two bifurcations that are too close. The whole pattern is shown in Fig.6 The closeness here is defined as 8 pixels wide. After removing the spurious minutia pixels, a minutiae pattern, consisting of approximately 20-40 ridge endings and bifurcations associated with corresponding ridge orientations, is obtained for matching pre-stored minutiae patterns in a database.

The next step after enhancement of the image is the extraction of minutiae. The

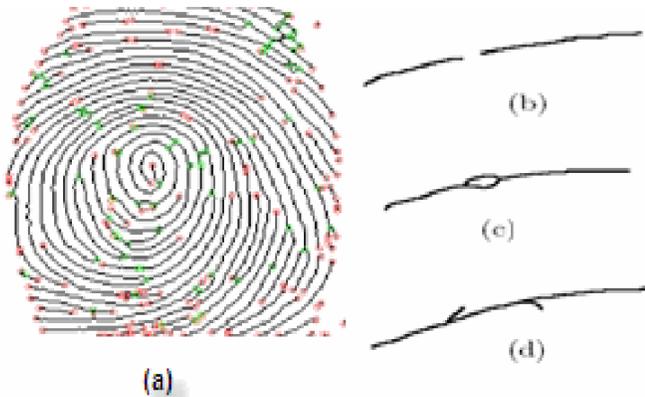


Fig 6 False Minutiae (a) Beyond margins (b) Broken ridges, (c) close bifurcation (d) Hair effect

minutiae points are thus those which have a pixel value of one (ridge ending) as their neighbor or more than two ones (ridge bifurcations) in their neighborhood. This ends the process of minutiae points extraction.

The minutiae extraction process, defined in [9], uses matrices of 3x3 pixels to search for typical minutiae, (ridge endings and ridge bifurcations). This method incurs many errors as there will be many ridge-ends in border regions and spurious bifurcations and ridge-endings inside the fingerprint. Most of them are clearly recognizable to the trained eye, but in the case of an automatic system, a series of rules should be used to delete the so-called “false-minutiae” [10]. Apart of these rules, is a Region-Of-Interest (ROI). In these areas the fingerprint is statistically less corrupt. This area makes all the ridge-ends located in border blocks invalid and marks them as false-minutiae because they are almost always caused by the fact that only a finite area of fingers can be in contact with the sensor surface that is acquiring the fingerprint image. This causes the apparent ending of many ridges in border regions and the ridge that’s ends one block a way from the borders must be ignored.

6. Crossing Number Theory

The enhanced fingerprint image is binarized and then used as input to the thinning algorithm to reduce the ridge thickness to one pixel wide. The skeleton image is used to extract minutiae points which are the points of ridge endings and bifurcations. The location of minutiae points along with the orientation is extracted and stored to form feature set. Eight connected pixels are used for extraction of minutiae points [6]. The Crossing Number (CN) method is used to perform minutiae extraction. This method extracts the ridge endings and bifurcations from the skeleton image by examining the local neighborhood of each ridge pixel using a 3x3 window. The CN for a ridge pixel P is given by

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}| \quad (7)$$

$$P_9 = P_1$$

where P_i is the pixel value in the neighbourhood of P . After the CN for a ridge pixel has been computed, the pixel can then be classified according to its CN value. A ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation as shown in Fig.7. For each extracted minutiae point, the following information is recorded:

- x and y coordinates,
- orientation of the associated ridge segment, and
- type of minutiae (ridge ending or bifurcation)

For a pixel P , its eight neighbouring pixels are scanned in an anti-clockwise direction as follows:

| | | |
|-------|-------|-------|
| P_4 | P_3 | P_2 |
| P_5 | P | P_1 |
| P_6 | P_7 | P_8 |

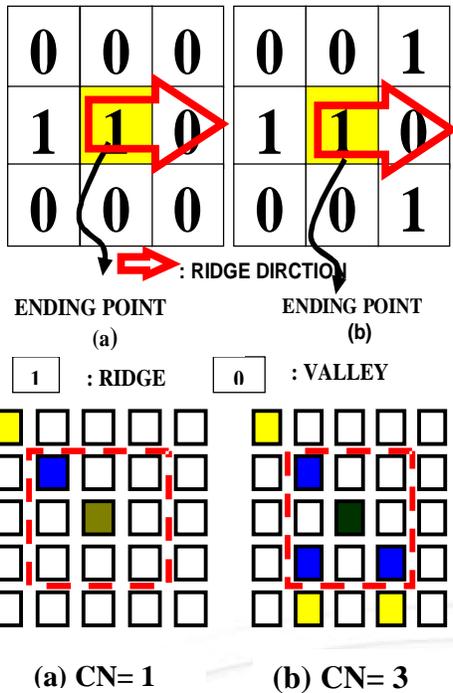


Fig 7. Termination and Bifurcation edges.

6.2 Binarization and thinning

It is well known the minutiae extraction depends on binarization. Image binarization classifies each pixel into either "ridge" (darker) or "valley" (lighter). From an enhanced image $\{B(i, j)\}$ in Figure 2, we apply a binarization operation according to the following rules:

(1) Assign "ridge" to pixel (i, j) if $B(i, j) \leq P_k$, where P_k is the k th percentile of histogram of $\{B(i, j)\}$.

(2) Assign "valley" to pixel (i, j) if $B(i, j) \geq P_k$.

(3) The remaining pixels $\{(i, j)\}$ are classified as "valley" if $B(i, j) \geq T_{5 \times 5}$, and "ridge" otherwise; where $T_{5 \times 5}$ is the 30th percentile of 5×5 pixel values surrounding the pixel (i, j) . A result of binarization is shown in Fig.5.

6.1 Skeletonization

After binarization, a binary image contains of ridge and valley pixels as shown in fig 5a. The width of a ridge at this step is in average 8-12 (pixels) which is hard to extract the minutiae points. Skeletonization (fig.5b) is an iterative thinning process that reduces a binary image consisting of ridges and valleys to a ridge map of unit width as shown in Fig.8.

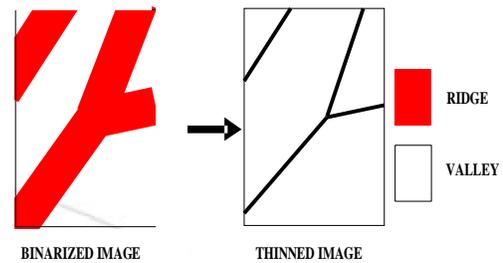


Fig. 8 Edge width and thinning

Feature extraction

There are two main types of features in a fingerprint: (1) Global ridge and furrow structures which form special patterns in the central region of the fingerprint, this type is used to classify a fingerprint and (2) Local ridge and furrow minute details. This type is used to identify a fingerprint (ridge endings and bifurcations, also known as minutiae). Basically, it is features (ridge endings and bifurcations, also known as minutiae), consists of three components: (i) Orientation field estimation, (ii) ridge extraction, and (iii) minutiae extraction and postprocessing. The overall flowchart of a typical algorithm, is depicted in Figure 9.

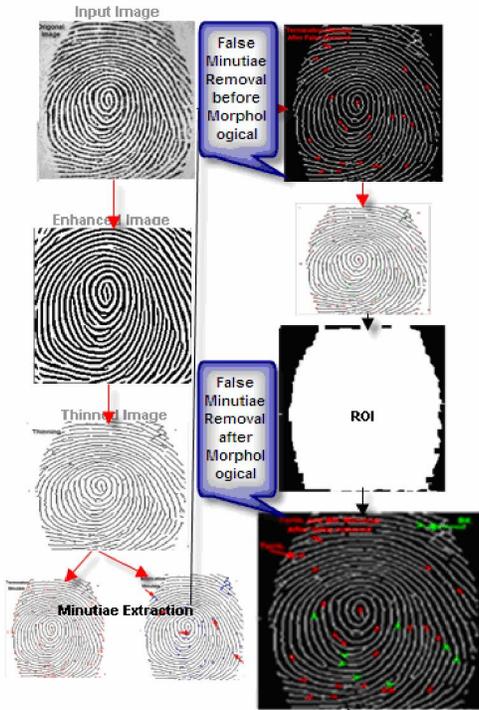


Fig.9 Flowchart of the minutiae extraction algorithm

7. Fingerprint image post processing

The minutiae points obtained in the above stage may contain many spurious minutiae. rules are used to detect false-minutiae, like : broken ridges, short ridges, hole structures and bridges.

False minutiae is generated by many factors such as noisy images, and image artifacts created by the thinning process. when the

minutiae are extracted, it is necessary to employ a post processing stage in order to validate the minutiae. Figure 10 illustrates some examples of false minutiae structures, which include the spur, hole, triangle and spike structures [11]. It can be seen that the spur structure generates false ridge endings. however, both the hole and triangle structures generate false bifurcations. The spike structure creates a false bifurcation and a false ridge ending point.

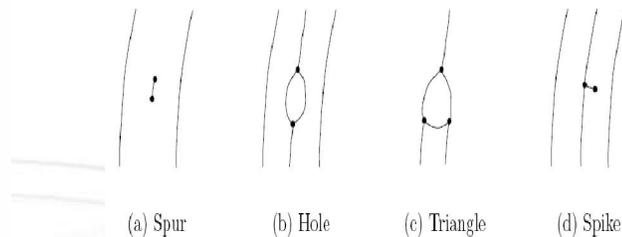


Figure.10 False minutiae structure

The majority of the image post processing in literature are based on a series of structural rules used to eliminate spurious minutiae. One such approach is the one proposed by Ratha et al. [12], which performs the validation of minutiae based on a set of heuristic rules. For example, a ridge ending point that is connected to a bifurcation point, and below a certain threshold distance is eliminated. This heuristic rule corresponds to removal of the spike structure shown in Figure 10 (d). Additional heuristic rules are then used to eliminate other types of false minutiae. Furthermore, a boundary effect treatment is applied where the minutiae below a certain distance from the boundary of the foreground region are deleted.

7.1 False minutiae extraction

The steps taken to remove false minutiae are follows. first: if the distance between a termination and a bifurcation is smaller than D , (D is calculated by Euclidian Formula).second, if the distance between two bifurcations is smaller than D , the third one is, if the distance between two terminations is smaller than D . Two morphological step should be taken to determined ROI. These steps are erosion and

closing. Different approaches may be used to determine the angle of termination and bifurcation. The Termination can be estimated through analyzing the position of the pixel one time for termination and three times for bifurcation on the boundary of a 5 x 5 bounding box and compare this position to the Table variable. The Table variable gives the angle in radian as shown in table 1. three times.

| | | | | |
|-----------|-----------|----------|----------|----------|
| $3\pi/4$ | $\pi/3$ | $\pi/2$ | $\pi/3$ | $\pi/4$ |
| $5\pi/6$ | $\pi/6$ | 0 | 0 | $\pi/6$ |
| π | 0 | 0 | 0 | 0 |
| $-5\pi/6$ | 0 | 0 | 0 | $-\pi/6$ |
| $-3\pi/4$ | $-2\pi/3$ | $-\pi/2$ | $-\pi/3$ | $-\pi/4$ |

Table.1 Window 5x5 Angle position

7.2 Labeling Minutiae

In marking minutia points, the following steps are taken, first: if the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch as shown in Figure 7(a). second If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending as shown in Figure 7(b).

8.0 Results and discussion

It is well known that fingerprint images are in poor quality. The performance of a fingerprint image matching algorithm relies critically on the quality of the input fingerprint images. The object of this test is to show the result of each stage we obtain using the enhancement algorithm, so we can determine how well each stage is. the poor ridge structure quality in fingerprint images are not always well-defined and hence they can not be correctly detected. This leads to the following problems: (i) a significant number of spurious minutiae may be created, (ii) a large percentage of genuine minutiae may be ignored, and (iii) large errors in minutiae localization (position and orientation) may be generated. To ensure that the performance of the minutiae extraction algorithm is robust

with respect to the quality of fingerprint images, an enhancement algorithm which can improve the clarity of the ridge structures is necessary. The process starts with normalization of input fingerprint image so that it has pre-specified mean and variance. The orientation image is estimated from the normalized input fingerprint image. Further, frequency image is computed from the normalized input fingerprint image and the estimated orientation image. After computation of frequency image the region mask is obtained by classifying each block in the normalized input fingerprint image into a recoverable or unrecoverable block. finally, a bank of Gabor filters which is tuned to local ridge orientation and ridge frequency is applied to the ridge and valley pixels in the normalized input fingerprint image to obtain an enhanced fingerprint image.

The second step is to find and to determine the type of fingerprint image gradients in x and y direction is calculated using Gaussian Filter. Estimate the local ridge orientation at each point by finding the principal axis of variation in the image gradients. Then Covariance data for the image gradients and Smooth the covariance data to perform a weighted summation of the data then Analytic solution of principal direction of Sine and cosine of doubled angles then Smoothed sine and cosine of doubled angles. all of the above operation are performed to Calculate 'reliability' of orientation data. the third step the technique of minutiae extraction.

The minutiae are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3 x 3 window. The CN value is then computed, which is defined as half the Sum of the differences between pairs of adjacent pixels in the eight-neighborhood. Using the properties of the CN as shown in Table 1, the ridge pixel can then be classified as a ridge corresponds to a ridge ending, and a CN of three corresponds to a bifurcation. The number of minutiae in a given area is also limited therefore the minutiae density must also

be kept in check. In order to filter out these false minutiae a three level-filtering process is applied:

Level 1: Removes the false ridge endings created as a result of the application of minutiae extraction algorithm at the ends of the thinned image.

Level 2: Removes the first five types of minutiae mentioned above using the rule based morphological minutiae filtering

Level 3: This stage limits the maximum number of minutiae present in the thinned image to a pre-specified threshold.

There are a lot of spurious minutiae.

The process of removing the false minutiae by comparing the suggesting distance with distance between the termination minutia and bifurcation we determined by the following steps.

step 1: if the distance between a termination and a bifurcation is smaller than D , we remove this minutiae.

step 2: if the distance between two bifurcations is smaller than D , we remove this minutia step 3: if the distance between two terminations is smaller than D , we

remove this minutia. The result of first step is done by removal the superior minutiae by using the Euclidian algorithm to calculate the distance between the termination and bifurcation minutiae. The second step is calculate the distance between the termination minutiae.

The final steps is calculating the distance between the bifurcation minutia.

The result of the above steps is shown in the following table.

| | |
|---------------------------------------|---|
| Distance | 6 pixel |
| Term.-bif. before processing | Number of Term.=164 Number of Bif.= 75 |
| Term.-bif. Distance After processing | Number of Term.= 60 Number of Bif.= 14 |
| Term.-Term. Distance After Processing | Number of Term.=24 |
| Bif.- bif. Distance After Processing | Number of Bif.=12 |

Conclusion and recommendation

All the methods and algorithms described in this study were implemented using MATLAB V6.5. The purpose is to illustrate the results of each stage in the enhancement algorithm and to assess how well each stage performs.

All the methods and algorithms described in this paper will be implemented in java.

A fingerprints is represented as an object has attribute and behaviors to enable the user to detect and manipulate false minutiae. This process includes creating fingerprint editor, a user can use to edit and process the image.

6.0 References

- [1] O. Baruch, "Line Thinning by Line Following," *Pattern Recognition Letters*, vol. 8, no.4, pp. 271-276, 1988
- [2] Mehtre, B. M. Fingerprint image analysis for automatic identification. *Machine Vision and Applications* 6, 2 (1993), 124–139.
- [3]. A.M. Bazen and S.H. Gerez, "Directional field computation for fingerprints based on the principal component analysis of local gradients," 1993
- [4] Jie Zhou and Jinwei Gu, "A Model-based Method for the Computation of Fingerprints' Orientation Field," *IEEE Transactions on Image Processing*, vol.13, no.6, p. 821-835, June 2004.
- [5] Jinwei Gu and Jie Zhou, "Analysis of Singular Points in Fingerprints based on Topological Structure and Orientation Field", Technical Report, 2005.
- [6] N. Ratha, K. Karu, S. Chen and A. K. Jain, A Real-time Matching System for Large Fingerprint Database, *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 18, No. 8, pp. 799-813,1996.
- [7] C.Y, Chang, Automatic Fingerprint Verification System. *M.S. Thesis, National Tsing Hua University*, Taiwan, June 2003.
- [8] J. Hollingum, "Automated Fingerprint Analysis Offers Fast Verification," *Sensor Review*, vol. 12, no. 3, pp. 12-15, 1992.

[9] T. Y. Zhang , C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", Communications of the ACM, v.27 n.3, p.236-239, March 1984.

[10] C.Arcelli and G.S.D.Baja, "A Width Independent Fast Thinning Algorithm," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 7, no. 4, pp. 463-474, 1984.

[11]Hong, L., Wan, Y., and Jain, A. K. 'Fingerprint image enhancement: Algorithm and performance evaluation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8 (1998), pp 777-789.

[12] N. Ratha, K. Karu, S. Chen and A. K. Jain, A Real-time Matching System for Large Fingerprint Database, *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 18, No. 8, pp. 799-813,1996.

