# A New Approach to Finger Print Matching Technique

***D. M. Akbar Hussain*, Henrik Legind Larsen*, Shaiq A. Haq***
*Department of Computer Science and Engineering
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg DENMARK
Phone: (+45) 7912 7729   Fax: (+45) 7512 7710
Professor and Chief Consultant IT, IT Research Centre
University of Engineering & Technology, Lahore PAKISTAN
Office Tel. +92(42)682-9397.

**Abstract:** A fingerprint is a pattern of ridges and valleys that exist on the surface of the finger. The uniqueness of a fingerprint is typically, determined by the overall pattern of ridges and valleys as well as the local ridge anomalies e.g., a ridge bifurcation or a ridge ending, which are called minutiae points. Designing a reliable automatic fingerprint matching algorithm is quite challenging. However, the popularity of fingerprint sensors as they are becoming smaller and cheaper, automatic identification based on fingerprints is becoming not only attractive but an alternative complement to the traditional methods of identification. The critical factor in the widespread use of fingerprints identification is, satisfying the performance e.g., matching the speed and accuracy requirements of the application. The widely used minutiae-based representation utilizes this discriminatory information available in a fingerprint for a matching. However, we extend this process of matching through a triangular feature to improve the matching process. The result shows that it has improvement in recognition process.

**Keywords:** Finger print recognition, Biometrics, Gabor Filter, Matching, Minutiae.

## 1. INTRODUCTION

Fingerprint Identification System is an important biometric technology. Feature extraction (minutiae) is a key step in a finger print matching system. However, because of imperfections in the image acquisition process, minutiae extraction methods are prone to miss certain information and could also give error in the location of the minutiae [1][2].

### 1.1 Background of Minutiae Based Method

Typically, each minutia is described by its location in the image *(x,y)*, orientation $\theta$ and the type *'t'* meaning if it is ridge ending or bifurcation etc., as shown in figure 1. In general some other attributes can also be included in the minutiae. Suppose ***T*** represents the template and ***I*** represent the input fingerprint with location, orientation and type:

$$T = \{m_1, m_2, ........m_m\} \qquad m_i = \{x_i, y_i, \theta_i, t_i\} \qquad i = 1....m \tag{1}$$

$$I = \{n_1, n_2, ........n_n\} \qquad n_j = \{x_j, y_j, \theta_j, t_j\} \qquad j = 1....n \tag{2}$$

Where m and n are numbers of minutiae in ***T*** and ***I,*** respectively. Now a minutia is considered matched with another minutia if the spatial distance $\boldsymbol{D_s}$, between them is equal or smaller that a given threshold $\boldsymbol{d_0}$ and direction difference $\boldsymbol{\theta_d}$ is equal or smaller than a given threshold $\boldsymbol{\theta_0}$ [3].

$$D_s(n_j, m_i) = (x_j - x_i)^2 + (y_j - y_i)^2 \leq d_0 \tag{3}$$

$$\theta_d(n_j, m_i) = \sqrt{\min(|\theta_j - \theta_i|, \quad 360^0 - |\theta_j - \theta_i|)} \leq \theta_0 \tag{4}$$

In order to maximize the number of minutiae matching ***T*** and ***I*** must be aligned. However, correct alignment some times requires, displacement in *x, y* and in rotation $\theta$ to be recovered. Another factor which is significant would be the scale in case the finger prints are obtained at different resolutions. Tolerating a higher number of transformations results in additional degrees of freedom to the finger print matcher, therefore, this issue has to be carefully evaluated as each degree of freedom bring whole

4th WSEAS International Conference on Applied Mathematic and Computer Science, AMCOS 2005: Rio de Janeiro, Brazil April: 25 – 27 2005.

set of possible alignments and may result in false matching.

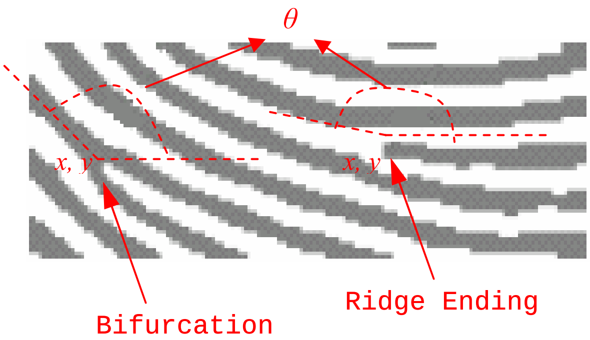


**Figure 1: Minutiae for Ridge Bifurcation and Ending**

## 2. Finger Print Matching Algorithm

A finger print matching algorithm has to execute a number of intermediate steps before the real matching process begins. A flowchart for these processes is shown in figure 2. In the following paragraphs a background for these processes with our implementation is provided.

### 2.1 Normalization

The image has to be normalized to a pre-specified mean and variance. Normalization process reduce variations in grey level values along ridges and valleys, the normalized image is given as:

$$N(i,j) = \begin{cases} \mu_0 + \sqrt{\dfrac{\sigma_0 (P(i,j)-\mu)^2}{\sigma}} & if\ P(i,j) > \mu \\ \mu_0 - \sqrt{\dfrac{\sigma_0 (P(i,j)-\mu)^2}{\sigma}} & otherwise \end{cases} \quad (5)$$

Where $P(i, j)$ is the grey level value at pixel $(i, j)$, $\mu$ and $\sigma$ are the estimated man and variance of P respectively. $N(i,j)$ is the normalized grey level value at pixel $(i, j)$, $\mu_0$ and $\sigma_0$ are the desired mean and variance values respectively.

$$\mu = \frac{1}{height*width} \sum_{i=0}^{height} \sum_{j=0}^{width} P(i,j) \qquad (6)$$

$$\sigma = \frac{1}{height*width} \sum_{i=0}^{height} \sum_{j=0}^{width} (P(i,j)-\mu)^2 \qquad (7)$$

### 2.2 Orientation

For the orientation, image is divided into blocks of size *(h x w)* e.g., in our case (3 x 3) and then gradients $\partial x\ (i, j)$ and $\partial y\ (i, j)$ at each pixel are calculated using Sobel operator. Sobel edge detection masks detect vertical and horizontal edges separately. And these directional edges are combined finally.

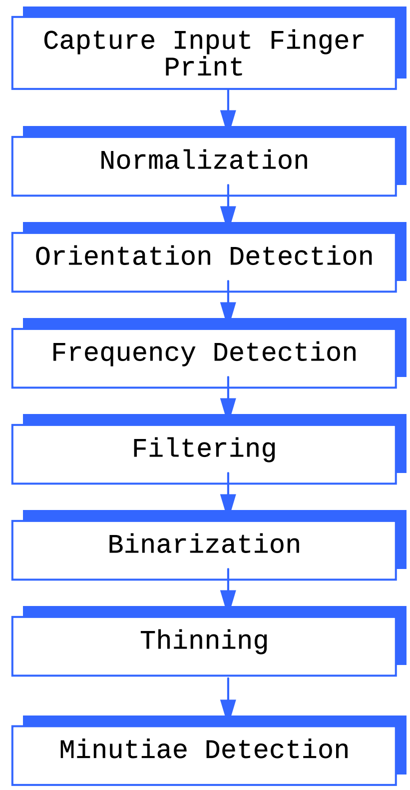


**Figure 2: Steps Leading to Minutiae Detection**

$\partial_x$ = (a2+2a5+a8)-(a0+2a3+a6)

$\partial_y$ = (a0+2a1+a2)-(a6+2a7+a8)

| | | |
|---|---|---|
| a0 | a1 | a2 |
| a3 | a4 | a5 |
| a6 | a7 | a8 |

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

$\partial_x$

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

$\partial_y$

Sobel masks for $\partial_x$ and $\partial_y$ , respectively. The least mean square estimate of the local ridge orientation centered at pixel $(i, j)$ is given by [3][4],

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\partial_y(i,j)}{\partial_x(i,j)}\right) \quad (8)$$

## 2.3 Ridge Frequency

The ridge frequency is made by the ridges and valleys in the local neighbourhood as a sinusoidal shaped plane wave. The determination of the frequency is made by dividing the normalized image *'N'* into blocks of certain width and length *(w x w)*, in our implementation it is 10x10. For every block centered at *(i, j)* create an orientation window of some size (*l x w),* 32x10 as shown in figure 3. Now calculate *x* signature *x[0], x[1], ……x[l-1]*, of the ridges and valleys, for each block centered at *(i, j)* as:

$$x[k] = \frac{1}{w}\sum_{n=0}^{w-1} N(u,v) \qquad k = 0,1,..(l-1) \quad (9)$$

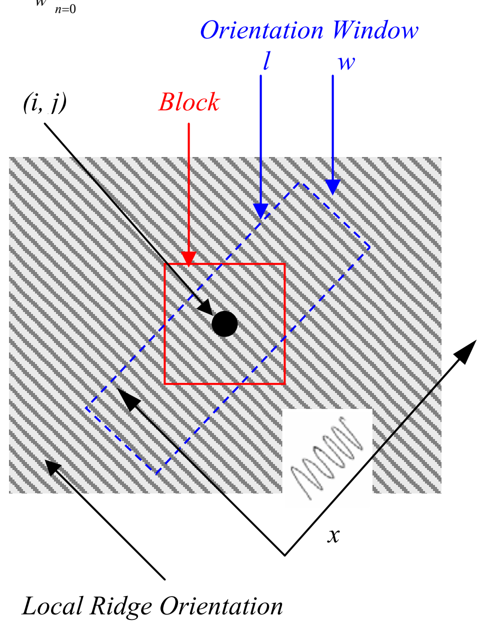


**Figure 3: Block and Oriented Window Description**

$$u = i + \left(d - \frac{w}{2}\right)\cos\theta(i,j) + \left(k - \frac{l}{2}\right)\sin\theta(i,j) \quad (10)$$

$$u = j + \left(d - \frac{w}{2}\right)\sin\theta(i,j) + \left(k - \frac{l}{2}\right)\cos\theta(i,j) \quad (11)$$

Therefore, if there are no minutiae points in the oriented window then x signature will form a discrete sinusoidal shape wave having the same frequency as of ridges and valleys in the oriented window. The frequency *f(i, j)* can be computed by finding the number of pixels (average=*γ(i, j)*) between the two peaks as:

$$f(i,j) = \frac{1}{\gamma(i,j)} \quad (12)$$

For a finger print image scanned at 500 dpi typically has frequency in range [1/3, 1/25] [3].

## 2.4 Filtering

Filtering is required to remove undesired noise from the fingerprint image [5]. Gabor filter is typically, used as a band-pass filter to remove the undesired noise and to keep preserve true ridges and valleys structures. It is defined as:

$$h(x,y:\varphi,f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\varphi^2}{\delta_x^2} + \frac{y_\varphi^2}{\delta_y^2}\right]\right\}\cos(2\pi f x_\varphi) \quad (13)$$

$$x_\varphi = x\cos\varphi + y\sin\varphi \quad (14)$$

$$y_\varphi = -x\sin\varphi + y\cos\varphi \quad (15)$$

Where $\varphi$ is the orientation of the Gabor filter, f is the frequency of a sinusoidal plane wave, $\delta_x$ and $\delta_y$ are the space constants of the Gaussian envelope along x and y axes respectively. In order to implement the Gabor filter these three parameters frequency, orientation and the space constants have to be provided. The two parameters are obviously obtained from the local orientation and the local ridge frequency just described earlier in the text. However, the space constants involve a compromise. Larger value means more rectification of noise, but on the other hand more chances of creating spurious ridges and valleys. Smaller value means it may create less spurious ridges and valleys but filtering of noise is affected. Therefore, the best option could be, obtain an average values by empirically from a set of data images. In our implementation we have taken a value of 3 for both constants. The enhanced image ***E(i, j)***, can be obtained by:

$$E(i,j) = \sum_{u=-\frac{w_g}{2}}^{\frac{w_g}{2}}\sum_{v=-\frac{w_g}{2}}^{\frac{w_g}{2}} h(u,v:\theta(i,j),\varphi(i,j),N(i-u,j-v)) \quad (16)$$

where $\theta$, $\varphi$, $N$ and $w_g$ are the orientation, frequency, normalized image and the size of the Gabor filter (in our implementation it is 10), respectively.

### 2.5 Binarization

Binarization of the image basically, facilitates further processing. Because, true information is binary; ridges against background. This process takes the greyscale image and converts it into the binary image. However, as the fingerprint image do not posses same contrast characteristics having a single intensity threshold is inappropriate. The contrast characteristics even can vary if the finger is pressed more at the center or sides. Therefore, typically a locally adaptive thresholding is often used. The implementation here assumes a uniform contrast therefore, an average color value is determined from the image which is used as a threshold to transform the pixels.

### 2.6 Thinning

Before the minutiae detection the binary image is passed through a thinning process which converts the widths of ridges to a single pixel. Importantly, a reasonable thinning process should maintain connectivity and also produces minimum artefacts. These artefacts are primarily from erroneous bifurcations with one very small offshoot.

## 3. Feature (Minutiae) Extraction

Minutiae detection process take the thinned image and extract the minutiae points from that image. Ridge ending could be found where a thin line terminates and bifurcation could be found at the junction of three lines. As it is said in the previous paragraph there will be extraneous minutiae points because of the processing so far especially, filtering and thinning. These extra points are eliminated here using a threshold which for example in the case of bifurcation determines if any of the two branches of it is shorter than a certain value so it is not considered a valid minutia point. Also, in the case of ending for example found opposite to each other, therefore, this is likely to be the same line. Ending at the boundary of the image are also not valid candidates.

## 4. New Strategy for Matching

For each valid minutiae point attributes, position, orientation [4] and the type is determined and stored in file. In a typical minutiae matching method each minutiae of the subject is compared with a given set of minutiae and if for each individual minutiae the spatial distance and the orientation is less then a threshold, it is assumed to be a likely pair of the same finger. Obviously, not all the minutiae will obey this rule because of the reasons sited earlier, so a matching score giving a percentage of similarity is used to declare a match of the two finger prints. In our implementation we use minutiae of similar kind (bifurcation and ending) to create multiple triangles as shown in figure 4, only 3 are shown but typically it ranges from 10 to 100. After that we take the stored reference file having feature extraction attributes.

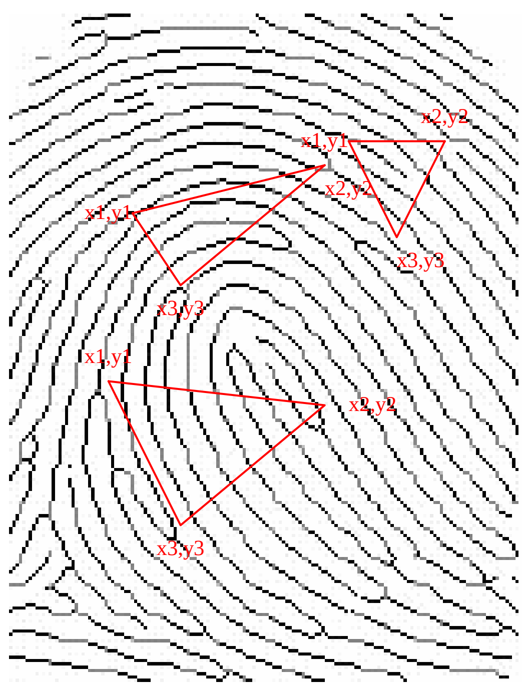


**Figure 4: Triangles**

We start from one of the vertex of a triangle and try to locate similar minutiae from stored reference file as typically done in the minutiae matching (distance and orientation). It is like an exhaustive search, if you don't find one try another vertex till you find one. After that as for each triangle starting from any vertex we have the knowledge of its side dimensions. Therefore, after locating a similar minutiae corresponding to that particular vertex, we try to locate other (2) minutiae which can make a similar triangle, using the dimensions obtained in the subjects image. Once we locate the other two vertices then a pattern matching technique is used staring from the center of the triangle as shown in figure 5. Basically, it compares the template comprising of firstly 8 pixels around the center point, secondly it compares 16 pixels (outer ring of 8) and so on it goes up to 4 levels depending upon the size of the triangle. After comparison between these two sets if the determined score is less than a threshold a match for those three minutiae and the template is declared.

## 5. Conclusion

The complete system was implemented using a finger print scanner FDC-FPC1010 daughter card with its platform support board DSK6713 from Texas Instruments. The resolution for this scanner is ~360 less than for example used typically by FBI or other law enforcing agencies. We ran tests with a moderate number of samples it shows an average improvement of about 20% compared with if we use only a standard minutiae matching method. One obvious penalty is it takes longer time to recognise a matching finger because of the processing involved, but then percentage wise it is better then the typical method. One of the limitation was not having a data base of various kind (category of ARCH, LOOP and WHORL) of finger prints so that an exhaustive matching experiments can be run to test the various aspects of the matching algorithm on these patterns. Higher resolution could also be another factor for further improvement in percentage matching.

## 6. References

[1] Jain,A., Hong,L. and Bolle,R.: On-Line Fingerprint Verification. IEEE-PAMI, Vol.19, No.4, pp. 302-314, Apr. 1997.

[2] Ratha,N., Karu,K., Chen,S., and Jain,A.: A Real Time Matching System for Large Fingerprint Databases. IEEE-PAMI, Vol. 18, No. 8, pp. 799-813, Aug. 1996.

[3] Hong,L., Wan,Y.and Jain,A.: Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 20: 777 -789. 1998.

[4] Marious Tico, Pauli Kuosmanen: Fingerprint Matching Using an Orientation Based Minutia Descriptor. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 8, 2003.

[5] Greenberg,S. Aladjem,M. and Kogan,D.: Fingerprint Image Enhancement using Filtering Techniques, Real-Time Imaging 8 ,227-236 (2002).

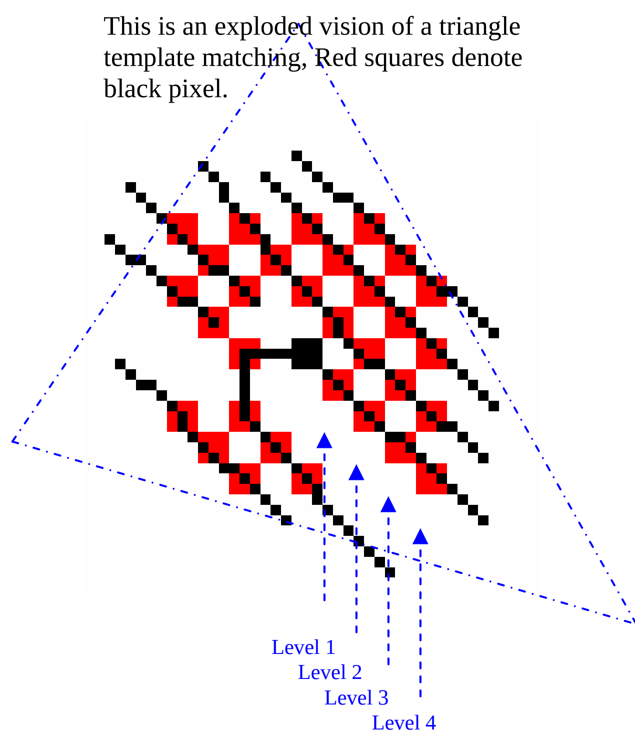


**Figure 5: Template Comparison Inside the Triangle**