IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 113-117

"AN EFFICIENT METHOD FOR FINGERPRINT RECOGNITION FOR NOISY IMAGES"

Vijay V. Chaudhary[1] and S.R. Suralkar[2]

[1]Department of E&TC (Digital Electronics), SSBT's COET Bambhori, Jalgaon India, *E-mail: vc_mp@yahoo.com*

[2]Department of E&TC, SSBT's COET Bambhori, Jalgaon India, *E-mail: shekhar_srs@rediffmail.com*

ABSTRACT

Fingerprint recognition has been successfully used by law enforcement for more than a century. The technology is now finding many other applications such as identity management and access control. Fingerprint recognition is an effective and speedy method towards accurate root cause determination. The skin on our palms and soles exhibits a flow-like pattern of ridges and valleys. These papillary ridges on the finger, called friction ridges, help the hand to grasp objects by increasing friction and improving the tactile sensing of surface textures. The "Friction Ridge Patterns" sidebar describes the nature and origin of these characteristics [1].

Another important issue in fingerprint recognition is core point positioning in noisy images. This paper describes the methodology for design a fingerprint recognition system that is capable of good identity of fingerprint and able to reject fingerprint for rescanning.

Keywords: Fingerprint recognition, Gabor filter, matching, pixel orientation.

1. INTRODUCTION

Fingerprint matching is among the most important and reliable methods for the identification of a person. There are two main applications involving fingerprint matching: Fingerprint verification and fingerprint identification. While the goal of fingerprint verification is to confirm the identity of a person, the goal of fingerprint identification is to establish the identity of a person. In general, fingerprint identification involves comparing a query fingerprint with a large number of fingerprints stored in a database, which is time consuming. To reduce search time and lower computational complexity, fingerprint classification is often employed to partition the database into smaller subsets [2]. The input image in fingerprint recognition is to be in good quality, if the image is in good quality noise is reduced. Other wise we need some image enhancement techniques to improve the quality of image for good results.

2. PROPOSED WORK

For our proposed algorithm following steps are as below.

2.1 Image Enhancement

Image enhancement is one of the most important processes in fingerprint recognition or identification. Most of the cases the scanned image is of poor quality and they did not give the proper results so before processing that image we need to enhance that image and convert in to good form (Ref figure 1). There are many parameters for image enhancement like contrast enhancement, noise reduction, filtering and so on. After this operation, the image is segmented (Ref figure 2) and background is separated from fingerprint image.

Figure 1: Enhanced Input Image

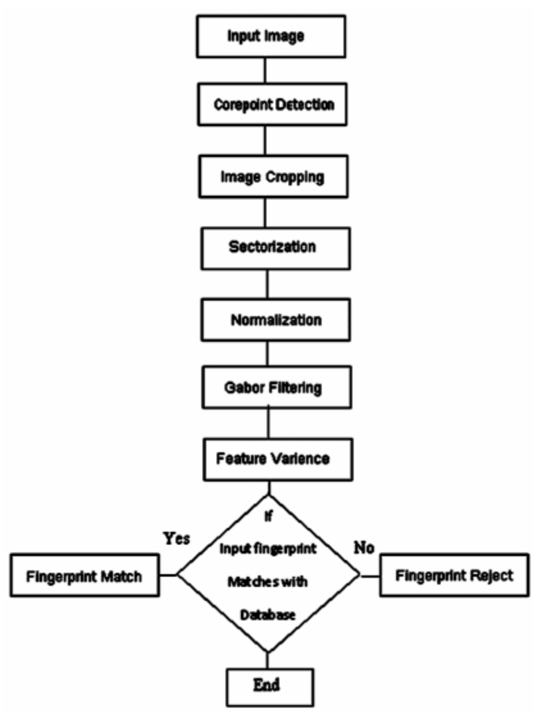

Figure 2: Flow Chart for Proposed Work

This can be performed using a simple block-wise variance approach, since background is usually characterized by a small variance. Image is first binary closed, then eroded, in order to avoid holes in fingerprint image and also undesired effect at the boundary (between fingerprint and background). The image segmentation is repeated several times, up to a desired condition is satisfied. This is done in order to avoid undesired boundary effects between fingerprint and background. The condition which has to be satisfied is chosen in the following way: the enhanced image is divided into non-overlapping blocks of given sizes (usually 32 × 32 or 64 × 64). [2] The whole enhanced image is filtered with a complex filter. Let Cf_max be the maximum value of the filtered image in the current region of interest. For each non-overlapping block we calculate the relative maximum Cf_rel. We finally consider a logical matrix F whose element (*I*, *J*) is equal to 1 if (*I*, *J*) is a block relative maximum and this value is equal or greater than a threshold value (usually 0.6*Cf_max in our simulations); *F* (*I*, *J*) is equal to 0 in all the other cases (i.e. if *F* (*I*, *J*) is not a block relative maximum or a block relative maximum smaller than the threshold value). If the number of non-zero elements of the logical matrix *F* is above a threshold value, the parameters for image segmentation are re-calculated and the whole process is repeated once again

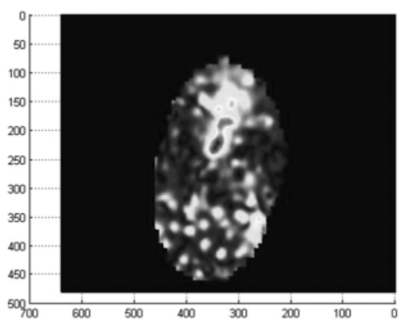

Figure 3: Output of the Enhanced Fingerprint Image After Filtering

2.2 Pixel wise image orientation

The orientation image represents an intrinsic property of the fingerprint images and defines invariant coordinates for ridges and furrows in a local neighborhood [4]. The orientation field of a fingerprint image represents the directionality of ridges in the fingerprint image. The block orientation could be determined from the pixel gradient orientations based on, say, averaging, voting, or optimization [5]. (Ref figure 4).

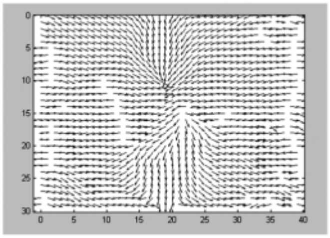

Figure 3: Pixel Wise Orientation of Input Image

Once the sum of values centered at pixel (*I*, *J*) has been calculated sum of *Y* pixels and *Z* pixels, in order to calculate the sum centered at pixel (*I*, *J* + 1) we simply subtract from the previous sum the *y* area and add the *X* area in this way it is possible to save a lot of computation. In other words,

$$\text{Sum}\,(i, j) = y + z$$

$$\text{Sum}\,(\text{in}, j + 1) = \text{sum}\,(i, j) - y + x$$

The main steps for the calculation of orient direction from the normalized image are:

The orientation image needs to be converted to continuous vector field using equation 1 and 2:

$$\varnothing x(i, j = \cos(2\theta(i, j)) \qquad (1)$$

$$\varnothing y(i, j = \sin(2\theta(i, j)) \qquad (2)$$

Where $\varnothing x$ and $\varnothing y$ are *x* and *y* components of vector field respectively.

All possible directions should get converted into eight directions in the range of 90 degrees to – 67.5 degrees. We round the value of the obtained direction to the nearest values of the desired range. We are considering only this range of values for the orientation field calculation. With this, a fairly smooth orientation field estimate can be obtained.

2.3 Core Point Positioning in Noisy Image

Accurate and reliable core point detection in fingerprint images is a critical issue that affects the performance of many fingerprint classification and recognition systems. Fingerprints have many conspicuous landmark structures and a combination of them could be used for establishing a reference point.

Figure 5: Core Point Positioning of Input Image

We define the reference point (Ref figure 5) of a fingerprint as the point of maximum curvature of the concave ridges (Ref Figure 6) in the fingerprint image [2].

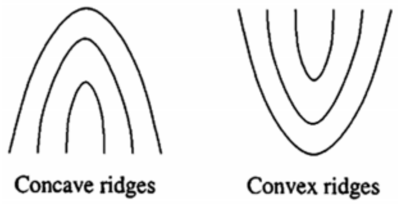

Figure 6: Concave and Convex Ridges in a Fingerprint Image

When the finger is positioned upright [2].

In This function the coordinates of core point.

x out : x-coordinates,

y out : y-coordinates

This routine is based on a novel, improved algorithm for core point localization. The parameters for image segmentation are determinate. Gradient magnitude is calculated from ($gx^2 + gy^2$) and theta is calculated from a tan($gx/gy + e^{-5}$).

All the points found in this function are subdivided into subsets of points which are quite near each other. We have N subsets. For each subsets we have a certain number of candidates and we consider only those subset which will have with a number of candidates >= 3. For each of this subset we consider the subset with the greatest x-averaged coordinate. In this subset we consider the core point the candidate with the greatest x-coordinate. This is a good approximation in standard fingerprint image.

2.4 Filtering

The true ridge and furrow structures of a fingerprint image can be greatly accentuated by applying properly tuned Gabor filters. These accentuated ridges and furrow structures constitute an efficient representation of a fingerprint image [3]. The general form of a 2D Gabor filter is defined by (Ref figure 7).

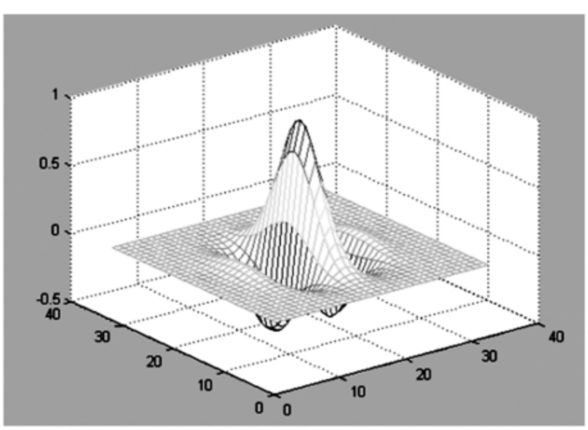

Figure 7: View of Gabor Filter for Input Image

$$g(x, y, f, \partial x, \partial y) = \exp[-1/2(x^2\theta k/\partial^2 x + y^2\theta k/\partial^2 y)] \times \exp(2\pi j f x\theta k) \quad (3)$$

Where

$$x\,\theta k = x\cos\theta k + y\sin\theta k \quad (4)$$

$$y\,\theta k = x\cos\theta k - y\sin\theta k \quad (5)$$

f = Frequency,

θk = Orientation of Gabor filter,

∂x, ∂y = Gaussian envelope along x and y axis.

By dividing the image into 8 × 8 sized blocks, then compute the image gradients $\delta x(i, j)$ and $\delta y(i, j)$ at each pixels (i, j). Calculate the local orientations at each pixel by finding principle axis of variation in image gradients, and Smooth the orientation field using Gaussian low pass filter.

The orientation field is used to obtain a logical matrix where pixel (I, J) is set to 1 if the angle of the orientation is <= $\pi/2$ after this computation we find that the border of this logical matrix, in the region of interest of fingerprint image. After this computation we calculate the complex filtering output of the enhanced fingerprint image [5]. It is not necessary to re-calculate it; we use the complex filtered image. Now we can find the maximum value of complex filtering output where the pixels of the logical image are set to 1. we repeat previous procedure for a wide set of angles (…, $\pi/2-3*\varphi$, $\pi/2-2*\varphi$, $\pi/2-1*\varphi$, $\pi/2$, $\pi/2+1*\varphi$, $\pi/2+2*\varphi$, $\pi/2+3*\varphi$, … where φ is an arbitrary angle). Each time I determine a point.

We subdivide all the points found in last step into subsets of points which are quite near each other. We will have said N subsets. For each subsets I will have a certain number of candidates and I consider only subset with a number of candidates >= 3. For each of this subset I consider the subset with the greatest x-averaged coordinate. In this subset we consider the core point the candidate with the greatest x-coordinate. This is a good approximation in standard fingerprint image. Pseudo match filtering is used to improve the Signal to Noise Ratio SNR of an image. Large alpha will increase the smoothness but also introduce artifacts.

2.5 Fingerprint Matching

Fingerprint matching refers to finding the similarity between two given fingerprint images. Due to noise and distortion introduced during fingerprint capture and the inexact nature of feature extraction, the fingerprint representation often has missing, spurious, or noisy features. Therefore, the matching algorithm should be immune to these errors [6].

The probability of making the correct verification is called the probability of correct verification or probability of verification the probability of making the false acceptance is called the false acceptance rate. The biometric system has decided that the user's claim is not correct and the claimed and the biometric identifier of the user belong to the same person. We say that a genuine individual has been falsely rejected. This error is called false rejection, and the probability of making the false rejection is called the false rejection rate. The probability of verification is equal to one minus the false reject rate [7].

3. SIMULATION RESULTS

For this simulation we use database DB1_B, DB2_B, DB3_B, DB4_B, PNG, VeriFinger_Sample_DB which has fingerprints from over 800 fingers from 100 different persons, each person has minimum eight impressions per finger [15,16]. This includes some good quality images and some noisy images [Ref fig.9]. We process on every image and calculate False Acceptance Rate and False Rejection Ratio. With a very small computation time for verification that is almost same for all matches. For 856 fingerprints we get False Rejection Ratio (FRR) of 21.49% and False Acceptance Rate (FAR) of 0.350 %.

Platform used for above simulation is MATLAB 7.8.0 (R2009a) 32 Bit, on Pentium Dual Core 2.30 GHz processor, which shows the better results for FAR and FRR (refer Table 1).

Table 1
FRR and FAR Values

FRR	*FAR*
21.49%	0.350%

Table 2
Simulation Results

Sr. No.	*Input Image*	*Fingerprint Recognize/ Fingerprint Reject*
1	Sample _Image_1	Fingerprint Recognize
2	Sample _Image_2	Fingerprint Recognize
3	Sample _Image_3	Fingerprint Recognize

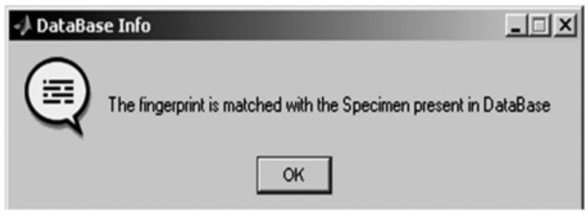

Figure 8: Result Display Window

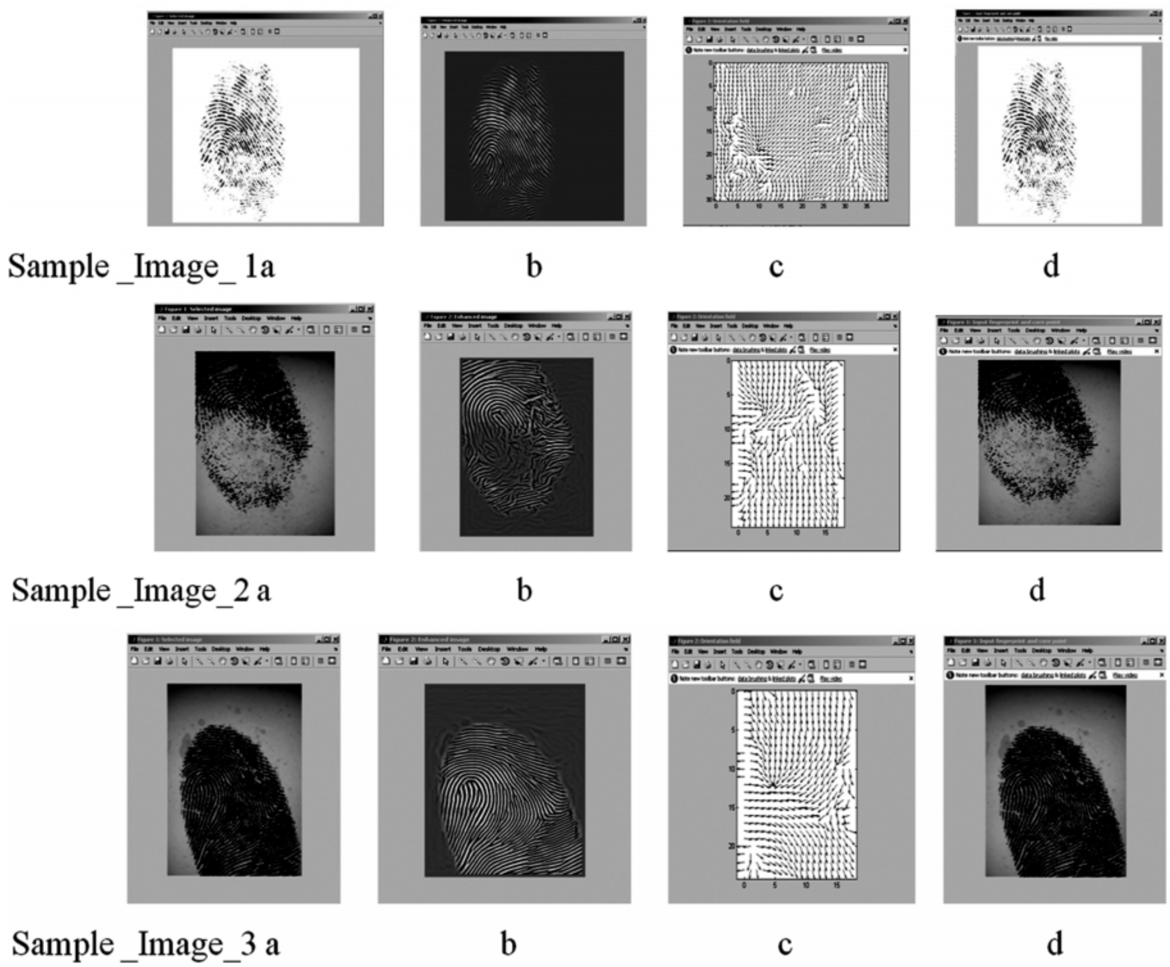

Figure 9: Simulation Results for Three Different Noisy Images, (*a*) Input Fingerprint Image, (*b*) Enhance Input Image, (*c*) Pixel Orientation of Input Image, (*d*) Core Point Detection in Input Image

4. CONCLUSION

From above simulation results we have proposed an efficient method for fingerprint recognition for noisy images. This technique exploits both the local and global characteristics in a fingerprint image to verify an identity. Each fingerprint image is filtered in a number of directions and we use pixel wise orientation with a fixed-length feature vector is extracted in the core point of the fingerprint. The matching stage computes the Euclidean distance between the template Fingerprint and the input Fingerprint, for the simulation we use Noisy images from database of over 800 fingerprints from 100 different persons, (Ref figure 9). Our proposed system shows better FAR results and we are able to achieve a verification accuracy which is only marginally inferior to the performance of a filter-based approach.

REFERENCES

[1] Ratha N., Karu K., Chen S., Jain A. (1996). "A Real-time System for Large Fingerprint Databases". *IEEE Trans Pattern Anal Mach Intell,* **18(8)**, 799-813.

[2] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti (2000). "Filterbank-Based Fingerprint Matching". *IEEE Trans, on Image Processing*, **9(5)**.

[3] Pradeep M. Patil, Shekhar R. Suralkar, Hemant K. Abhyankar, "Fingerprint Verification Based on Fixed Length Square Finger Code". *17th IEEE International Conference on tool with Artificial Intelligence (ICTAI'2005)*.

[4] Ravishankar Rao, "A Taxonomy of Texture Description", *Springer Verlag*.

[5] Kenneth Nilsson and Josef Bigun, "Localization of Corresponding Points in Fingerprints by Complex Filtering", *Pattern Recognition Letters*, **24**, (2003), 2135-2144.

[6] hthttp://www.csse.uwa.edu.aus /~pk/ Research/ Matlab Fn

[7] Mike Bone and James L. Wayman, "Evaluating Facial Recognition Technology for Drug Control Applications". Available online at: http://www.frvt.org/DLs/ONDCP2001.pdf.

[8] A.K. Jain, L. Hong, S. Pankanti, "An Identity Authentication System Using Fingerprints", *Proc. IEEE*, **85**, pp., Sept. 1997.

[9] A.K. Jain, R.M. Bolle, and S. Pankanti, "Eds., Biometrics: Personal Identification is a Networked Society". *Norwell, MA: Kluwer*, 1999.

[10] L. O'Gorman, "Fingerprint Verification", *in Biometrics: Personal Identification in a Networked Society*, A.K. Jain, R. Bolle, and S. Pankanti, Eds. Norwell, MA: Kluwer, 1999.

[11] D. Maio and D. Maltoni, "Direct Gray-scale Minutiae Detection in Fingerprints", *IEEE Trans. Pattern Anal. Machine Intell*., **19**, pp. 27-40, Jan 1997.

[12] A.K. Jain, S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification", *IEEE Trans. Pattern Anal. Machine Intell*., **21(4)**, pp. 348-359, 1999.

[13] L. Hong and A.K. Jain, "Classification of Fingerprint Images", *in 11th Scandinavian Conf. Image Analysis, Kangerlussuaq*, Greenland, June, 1999.

[14] S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching", *IEEE Trans. Pattern Anal. Machine Intell*., **18(4)**, pp. 377-388, 1996.

[15] http://bias.csr.unibo.it/fvc2000/download.asp

[16] http://www.advancedsourcecode.com/PNGfingerprint.rar