International Journal of Electronics Engineering, 1(2), 2009, pp. 273-276

Fingerprint Image Enhancement Using Fourier Filtering Technique

Pooja Gupta[1] & Jaspreet Kaur[2]

[1]*Department of ECE, M.M. University, Mullana, Haryana, INDIA*
[2]*Department of ECE, B.B.S.B.E.C., Fatehgarh Sahib, Punjab, INDIA*

Abstract: *Fingerprint verification is the most important stage in fingerprint verification. The robustness of the recognition system can be improved by incorporating an enhancement stage prior to feature extraction. Also, the definition of a noise in a generic image and fingerprints are widely different. The noise in a fingerprint consists of breaks in the directional flow of ridges. The filtering algorithm decomposes the input fingerprint image into a set of filtered image. From, the filtered image, the orientation is estimated and a quality mask which distinguishes the recoverable and unrecoverable corrupted regions in the image is generated. The input fingerprint image is adaptively enhanced in the recoverable regions. The experimental results show that the proposed algorithm reduces the influence of noise on the ridges and valleys.*

Keywords: *Fingerprint, Enhancement, Feature Extraction, Filtering, Orientation, Ridges, Valleys.*

1. INTRODUCTION

In an ideal fingerprint image, ridges can be easily detected and minutiae can be precisely located from the thinned ridges. The ridge structures in poor quality fingerprint images are not always well defined and hence they cannot be correctly detected. This leads to following problems: (a) a significant number of unreliable minutiae patters can be created, (b) a long percentage of minutiae can be ignored and (c) large errors in their localization (position and orientation) can be introduced [7]. A number of techniques to enhance fingerprint images have been proposed, which take advantage of filtering techniques of fingerprint enhancement [1, 2, 3, 4, 6].

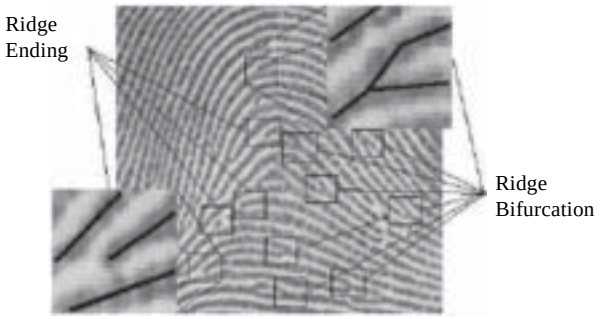

Figure 1: Examples of Ridge Ending and Bifurcation

2. RELATED WORK

Another approach based on directional filtering kernel is by Hong *et al.* The main stages of their algorithm are as follows:

*Corresponding Author: excellent_pooja@yahoo.com

1. Normalization: This procedure normalizes the global statistics of the image, by reducing each image to a fixed mean and variance.
2. Orientation Estimation: This step determines the dominant direction of the ridges in different parts of the fingerprint image. This is a critical process and errors occurring at this stage are propagated into the frequency estimation and filtering stages.
3. Frequency Estimation: This step is used to estimate the inter-ridge separation in different regions of the fingerprint image.
4. Segmentation: In this step, a region mask is derived that distinguishes between 'recoverable', 'unrecoverable' and 'background' portions of the fingerprint image.
5. Filtering: Finally using the context information consisting of the dominant ridge orientation and ridge separation, a band pass filter is used to enhance the ridge structure [6].

The enhancement consists of a filtering stage followed by a thresholding stage. The filtering stage produces a directionally smoothed version of the image from which most of the unwanted information ('noise') has been removed, but which still contains the desired information (i.e. the ridge structure and minutiae). The thresholding stage produces the binary, enhanced image [1].

Sherlock and moron perform contextual filtering completely in the Fourier domain. Each image is convolved with precomputed filters of the same size as the image. The precomputed filter bank (labeled PF0, PF1... PFN) are oriented in eight different direction in intervals of 45°, they are applied to the original image, yield a set of directionally filtered images. These are called 'prefiltered images' [6].

The filtered image is then built up by selecting, for each pixel position, the pixel value from the prefiltered image whose direction of filtering corresponds most closely to the actual ridge orientation at that position (see Figure 2). In order to perform this selection operation, knowledge of the actual ridge orientation is required, and this is obtained by estimation from the original image. Finally, the thresholding stage binarises the directionally filtered image using a local average as the threshold surface.

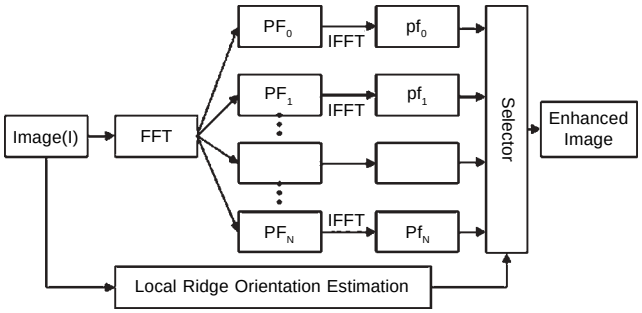

Figure 2: Block Diagram of Fingerprint Enhancement Algorithm

In Figure 3(a); this spectrum has twin peaks. The location of these peaks in the Fourier domain characterizes two fingerprint features-the frequency and the direction of the ridge flows in the area. The frequency of ridges is indicated by the distance between the peaks and the direction of the ridges is indicated by the direction of the line connecting the peaks. For low quality images such as those in Figures 3(b), and 3(c), however, the peaks are not distinct. In Figure 3(b) the image is blurred and the peaks in (b') are also blurred, and in (c) alphabetical letters appear in the image and as a consequence other peaks appear in (c') [3].

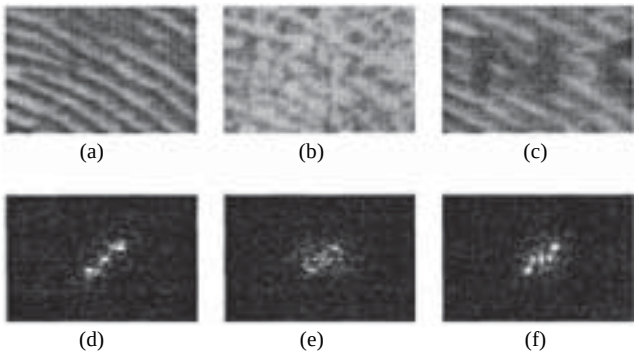

Figure 3: Fourier Power Spectra of Fingerprint Images. (a), (b) and (c) are Original Images and (a'), (b') and (c') are Fourier Power Spectra of (a), (b) and (c), Respectively.

In fingerprint image processing, one has the advantage of a well defined local frequency and orientation of information. In forming the Fourier transform of the image, the immediate neighborhood of each point contributes (to first approximation) a single frequency component. The LRS determines its distance from the Fourier origin, and the LRO determines its angular position [1].

Using polar coordinates (ρ, Φ), express the filter as a separable function

$$H(u, v) = H(\rho, \Phi) = H_{radial}(\rho)H_{angle}(\Phi) \quad (1)$$

Here (ρ, Φ) shows the polar co-ordinates in the Fourier domain and $(U, V) = (\rho \cos \Phi, \rho \sin \Phi)$ shows the orthogonal co-ordinates. $H_{radial}(\rho)$ depends upon LRS, and $H_{angle}(\Phi)$ upon LRO. Any good classical one-dimensional bandpass filter would be adequate for $H_{radial}(\rho)$; the Butterworth filter was chosen because its implementation is simpler than such alternatives as the Chebyshev or elliptic filter, especially if it is desired to vary the filter order *n*.

Therefore the following bandpass filter was designed as $H_{radial}(\rho)$:

$$H_{radial}(\rho) = \sqrt{\frac{[\rho\rho_{BW}]^{2n}}{(\rho\rho_{BW})^{2n} + (\rho^2 - \rho_0^2)^{2n}}} \quad (2)$$

Where ρ_{BW} and ρ_0 are the desired bandwidth and centre frequency. A value of $n = 2$ worked well and was used throughout.

$$H_{angle}(\phi) = \left\{ \cos^2 \frac{\pi(\phi - \phi_c)}{2\phi_{BW}} if\ |\phi| < \phi_{BW} \right\} \quad (3)$$

Where Φ_{Bw} is the 'angular bandwidth' of the filter, i.e. the range of angles for which $|H_{angle}(\Phi)| \geq 0.5$, and Φc is its 'orientation', i.e. the angle at which $|H_{angle}|$ is maximum.

3. IMPLEMENTATION OF FILTER

3.1. Using Prefiltered Images:

Direct implementation of the filter as an explicit function of position is computationally prohibitive. Instead, a small but representative set of values for the parameters LRS and LRO are chosen, thereby defining a small set of stationary filters. Applying these to the image yields a set of 'prefiltered images'. The final image is built up by interpolating for each position between pixel values from the prefiltered images corresponding to LRS and LRO values closest to the true values. The true values are estimated from the raw image [8].

3.2. Regions of High Ridge Curvature:

High ridge curvature occurs where the LRO changes rapidly, i.e. near cores and deltas. Away from these singular points, ridge curvature tends asymptotically to zero. In regions of higher curvature, a wider range of orientations is present. The angular bandwidth of the directional filter must therefore increase near singular points. Away from singular points, the angular bandwidth is n/8. The angular bandwidth must equal *n* at singular points since all orientations are

present. Figure 4, shows an empirical piecewise linear relationship giving angular bandwidth as a function of distance from the nearest singular point. Values of $r_1 = 0$ and $r_2 = 20$ pixels are used throughout. Where necessary, the effect of a filter of wider bandwidth than n/8 is simulated by an appropriately weighted combination of pixel values from two or more prefiltered images [1].

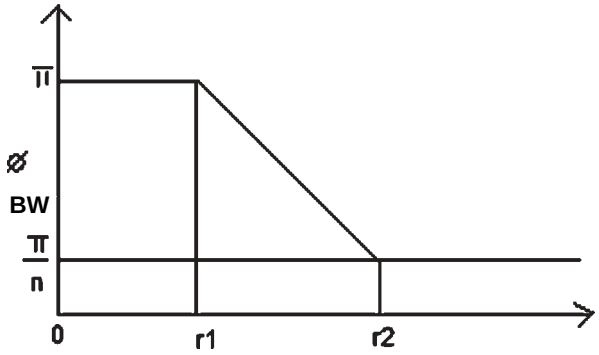

Figure 4: Filter Angular Bandwidths near Singular Points. The Graph Shows the Angular Bandwidth as a Function of the Distance *r* to the Nearest Core or Delta Point.

3.3. Algorithm for Estimating LRO at a Point:

LRO is determined directly from the image at a grid of points, with intermediate values obtained by interpolation. The algorithm for determining LRO at a point follows:

This window is rotated to 16 different orientations, $\Theta_i = i\Pi/16$, for $i = 0–15$. At each orientation a projection along the *y*-axis of the window is formed.

$$p_i(x) = \frac{1}{32}\sum_{Y=0}^{31} W_i(x, y) \quad x = 0-31 \qquad (4)$$

Where $W_i(x, y)$ is the data inside the window at angle Θ_i.

A second order butterworth bandpass filter removes noise from the projections. The total variation V_i of each filtered projection $fp_i(x)$ is evaluated as:

$$V_i = \sum_{x=0}^{31} f_{pi(x)}(x+1) - f_{pi}(x) \qquad (5)$$

the LRO estimate is given by $i_{\max}\Pi/16$, where $V_{i\max}$ is the maximum of 16 variations. This algorithm produces the correct value except the noisiest regions. The typical failure rate for inked fingerprints is approximately 0.5%.

Figure 5, projections of a window of fingerprint image data. The projection which exhibits the greatest variation corresponds to the orientation of the ridges within the window (here $\Theta = 2\Pi/8$) for clarity, eight rather than 16 projections are shown.

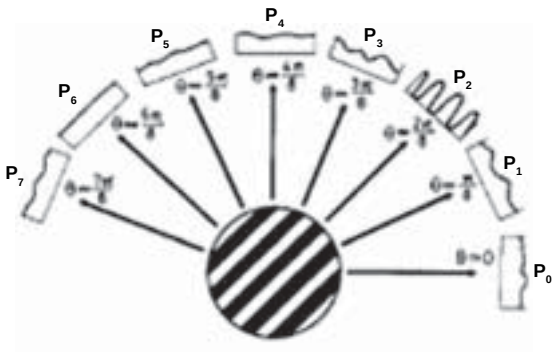

Figure 5:

4. FINGERPRINT ENHANCEMENT ALGORITHM

The enhancement algorithm consists of the directional filtering process described above, followed by a thresholding operation which converts the filtered image into a binary form. The input to this first stage consists of a 512 by 512 pixel raw fingerprint image and a 30 by 30 array of LRO values. The filtering algorithm follows.

1. Obtain 16 prefiltered images PF[i] from the raw fingerprint. Window the image, to reduce edge effects. Take the 2-D DFT of the windowed image.
2. Multiply this by the i^{th} filter function and inverse transform this to get the i^{th} prefiltered image PF[i].
3. Obtain the LRO at each point by interpolation between the 30 by 30 sampled values.
4. Determine the distance I from (x, y) to the nearest singular point. Using the function in Figure 3, determine the angular bandwidth that is required, and hence the number *j* of prefiltered images that must be combined.
5. Form the output pixel OUT (x, y) as a weighted average of the (x, y) pixels of the *j* prefiltered images nearest in orientation to the LRO. The result of applying this algorithm is a filtered image which has been smoothed in the direction of ridges.
6. Local average thresholding in a 32 by 32 pixel neighborhood is used. The threshold surface is

$$g_T(x, y) = \frac{1}{32^2}\sum_{i=0}^{31}\sum_{j=0}^{31} f(x = 16 + i, y = 16 + j) \qquad (6)$$

Where $f(x, y)$ is the filtered image.

5. EXPERIMENTAL RESULTS

The performance of a fingerprint feature extraction and matching algorithms depend heavily upon the quality of the input fingerprint image. The results of fingerprint image enhancement using Fourier filtering techniques has been shown. It can be seen that the enhancement improves the ridge structure even in the areas of high ridge curvature without introducing any artifacts. Visual inspection of several hundred images indicates that the method is very effective.

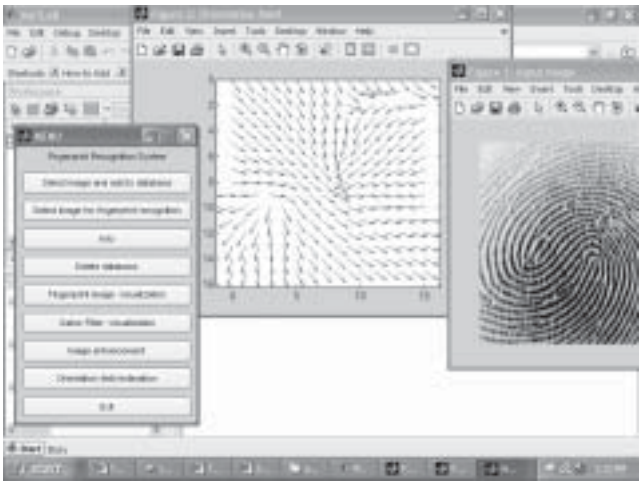

Figure 6: Result of Orientation Field Estimation.

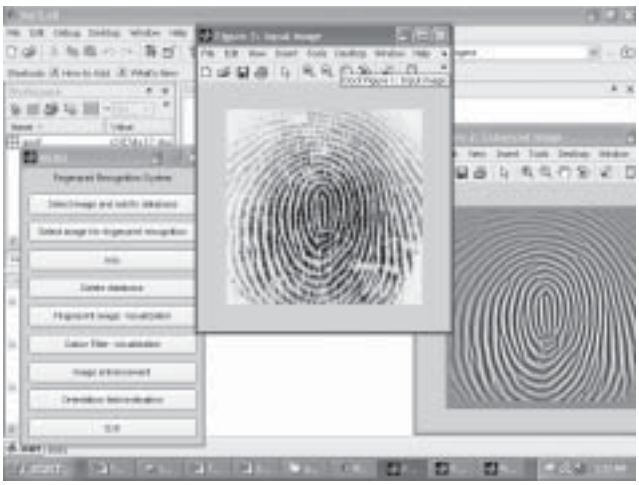

Figure 7: Result of Fingerprint Image Enhancement.

6. CONCLUSION

The fingerprint enhancement algorithm presented in this paper is of importance because it has been shown to produce significant improvements over the algorithm incorporated in a working AFIS. This demonstrates the usefulness of position-dependent Fourier domain filtering in the processing of fingerprint images within AFIS systems.

REFERENCES:

[1] B. G. Sherlock, D. M. Monro, K. Millard, 'Fingerprint Enhancement by Directional Fourier Filtering' in Proc. *IEEE Vision Image Signal Process.*, **141**, (2), (1994), 87–94.

[2] Oh, S. K.; Lee, J. J.; Park, C. H.; Kim, B. S. and Park, K.H., "New Fingerprint Image Enhancement Using Directional Filter Bank", in *Journal of WSCG*, **11**, (1), (2003).

[3] Kamei T. and Mizoguchi M., "Image Filter Design for Fingerprint Enhancement", in Computer Vision, Preceding Symposium, (1995), 109–114.

[4] Hong L.; Jian, A.; Pankanti, S. and Bolle, R.; "Fingerprint Enhancement" in *Applications of Computer Vision*, (1996), WAVC '96, Proceedings 3rd IEEE Workshop, (1996), 202–207.

[5] Chikkerur, S.; Wu C. and Govindaraju, V., "A Systematic Approach for Feature Extraction in Fingerprint Images", Hong Kong, ICBA, (2004).

[6] Chikkerur, S. S., "Online Fingerprint Verification System", M.Tech Thesis, State University of New York at Buffalo, (2005).

[7] Drahansky, I. M., "Biometric Security Systems Fingerprint Recognition Technology", Dissertation, Brno University of Technology, Faculty of Information Technology, (2005).

[8] Sherlock, B. G.; Monro, D. M., and Millard, K., "Algorithm for Enhancing Fingerprint Images", *Electron. Lett.*, **28,** (1992), 1720-1721.