Fingerprint Image Postprocessing Using Windowing Technique

Anam Tariq[1], M. Usman Akram[2], Sarwat Nasir[3], and Rabia Arshad[4]

[1,2,4]Department of Computer Engineering
[3]Department of Telecommunication
[1,2,4]College of Electrical and Mechanical Engineering
[3] College of Signals
National University of Science & Technology,Rawalpindi, Pakistan
anamtariq12@yahoo.co.uk, usmakram@gmail.com, sarwatnasir83@gmail.com, rabiakundi2007@gmail.com

Abstract. Performance of Automatic Fingerprint Identification System (AFIS) is greatly affected by its matching rate, means it should have low False Acceptance Rate (FAR) and False Rejection Rate (FRR). Minutiae based fingerprint matching techniques are normally used for fingerprint matching. In this paper, we present a new technique for fingerprint image postprocessing. This postprocessing is used to eliminate a large number of false extracted minutiae from skeletonized fingerprint images. We propose a windowing postprocessing method that takes into account the neighborhood of each minutia within defined window and check for minutia validation and invalidation. The results are confirmed by visual inspections of validated minutiae of the FVC2004 reference fingerprint image database. Experimental results obtained by the proposed approach show efficient reduction of false minutiae.

1 Introduction

Unique fingerprint of a person can be really helpful for identification purposes [1]. In AFIS, small portions of fingerprints are processed using complex algorithms [2]. 13 types of minutiae are used like empty cell, ridge ending, ridge fork, island, dot, broken ridge, bridge, spur, enclosure, delta, double fork, trifurcation, and multiple events [3]. Four main groups of minutiae are classified by the American National Standard Institute: terminations, bifurcations, crossovers and undetermined. But all of these are a combination of two fundamental features, ridge endings and bifurcations known as minutiae. Ridge ending or termination is defined as the point where a ridge ends abruptly, and bifurcation is the point where a ridge diverges into branches [1].

Most AFIS are minutiae based [1]. The minutia location and orientation can be uniquely determined using the method of least squares in a grayscale minutiae image [4].

The AFIS proposed in [5] uses syntactic processing to remove undesirable artifacts: two disconnected ridges are connected if their distance is less than a

A. Campilho and M. Kamel (Eds.): ICIAR 2008, LNCS 5112, pp. 915–924, 2008.

given threshold and endpoint directions are almost the same; wrinkles are detected by analyzing information on neighboring branch points. A three-step false minutiae identification process is adopted in [6]: (i) it repaired ridge breaks using the ridge directions close to the minutiae; (ii) Short ridges minutiae are dropped; (iii) crowded minutiae in a noisy region are dropped. False minutiae which are caused by noise or imperfect image processing are eliminated by a method described in [7]. A binarised image postprocessing technique was proposed in [8]. Spikes caused by the skeletonizing process are removed by the adaptive morphological filter proposed [9].

In this paper, we present a new windowing technique for false minutiae elimination. Our technique detects and removes false minutiae including spikes, holes, bridges, ladders and spurs. Our proposed technique decides that a candidate minutiae is false or not depending on its neighboring pixel values.

This paper is organized in five sections. Section 2 presents the steps for preprocessing and feature extraction from fingerprint images. Section 3 contains the proposed technique for fingerprint postprocessing. Experimental results of our technique are discussed in section 4 followed by conclusion in section 5.

2 Preprocessing and Feature Extraction

Preprocessing of fingerprint image means enhancing the fingerprint image using segmentation and orientation field estimation [1] while feature extraction means

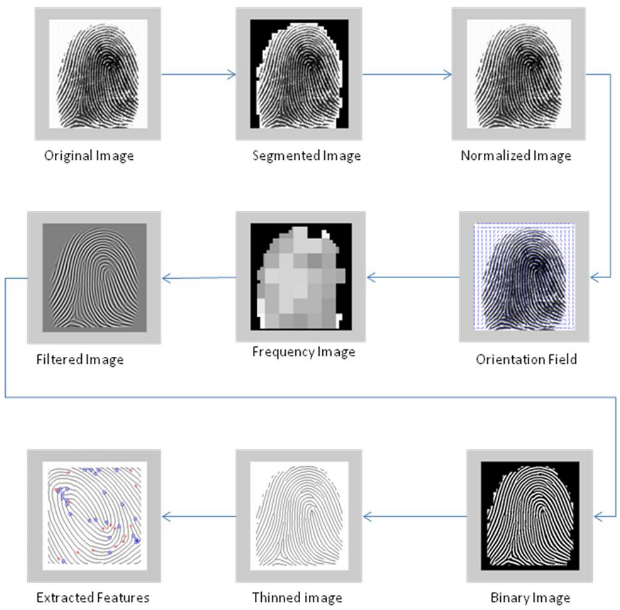

Fig. 1. Complete flow of preprocessing and feature extraction from an fingerprint image

extracting minutiae from a thinned image [1]. Figure 1 shows the complete flow of preprocessing and feature extraction.

2.1 Preprocessing

Image preprocessing normally consists of following steps:

Segmentation. Segmentation is done to extract fingerprint image from background. In AFIS, the processing of the surrounding background in fingerprint image is not necessary and consumes more processing time in all stages. Cutting or cropping out the region that contains the fingerprint feature (ROI) minimizes the number of operations on the fingerprint image [10].

Normalization. Normalization is performed to remove the effect of sensor noise and gray level background which are the consequence of difference in finger pressure. Normalization is used to standardize the intensity values in an image by adjusting the range of gray-level values so that it lies within a desired range of values [1].

Orientation Field Estimation. The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. The orientation estimation is a fundamental step in the enhancement process as the subsequent Gabor filtering stage relies on the local orientation in order to effectively enhance the fingerprint image [1].

Frequency Estimation. In addition to the orientation image, another important parameter that is used in the construction of the Gabor filter is the local ridge frequency. The frequency image represents the local frequency of the ridges in a fingerprint [11].

Gabor Filtering. Ridge orientation and ridge frequency parameters are used to construct the even-symmetric Gabor filter. A two-dimensional Gabor filter consists of a sinusoidal plane wave of a particular orientation and frequency, modulated by a Gaussian envelope [12].

Binarisation. Binarisation is the process that converts a gray level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image [1].

Thinning. The final image enhancement step performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide [1].

2.2 Feature Extraction

A critical step in studying the statistics of fingerprint minutiae is to reliably extract minutiae from the fingerprint images. The minutiae extraction technique that has been implemented is based on the widely employed Crossing Number method. The binary image is thinned as a result of which a ridge is only one pixel wide. The 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. Crossing number of pixel 'p' is defined as half the sum of the differences between pairs of adjacent pixels defining the 8-neighborhood of 'p'. Mathematically it is calculated using equation 1 [1].

$$CN(p) = \frac{1}{2}\sum_{i=1}^{8} |val(p_{imod8}) - val(p_{i-1})| \tag{1}$$

Where p_0 to p_7 are the pixels belonging to an ordered sequence of pixels defining the 8-neighborhood of p and val (p) is the pixel value. Crossing numbers 1 and 3 correspond to ridge endings and ridge bifurcations respectively. An intermediate ridge point has a crossing number of 2 as shown in Figure 2. The minutiae obtained from this algorithm must be filtered to preserve only the true minutiae.

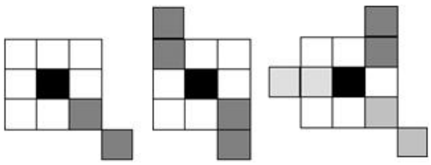

Fig. 2. From left to right: cn (p)=1, cn (p)=2 and cn (p)=3 representing a ridge ending, a non-minutiae region and a bifurcation

3 Proposed Technique

Our proposed technique deals with the skeleton fingerprint image obtained from preprocessing steps and use the minutiae neighboring information in an image to validate ridge endings and bifurcations. The minutiae points obtained by crossing number method contains a bunch of false minutiae. These false minutiae may occur due to noisy conditions in fingerprint image. These type of false minutiae are classified as spurs, ridge breaks, short ridges, holes/islands, bridges and ladders [1]. These unwanted minutiae points need to be removed in the postprocessing stage. Our postprocessing technique examined the minutiae points and tests their validity by taking thinned image into account. It takes the w×w neighborhood of that minutiae where

$$w = 2d + 1 \text{ where } d = \text{ Local ridge Distance} \tag{2}$$

The local ridge distance defines the average ridge distance in each region of the image [13]. Figure 3 shows the flow chart of our proposed technique.

3.1 Minutiae Invalidation

1st step of our postprocessing technique is to invalidate the minutiae with false minutiae structures, once these configurations have been recognized.

Elimination of False Minutiae due to Spurs, Bridges and Ladders. For each candidate bifurcation point,

1. Take a window of size w x w.
2. Place the candidate ridge bifurcation at the center of window.
3. Move in the direction of connected branches of the bifurcation.
4. If an end point of any traveled branch is detected within the window, eliminate that ending point and the candidate bifurcation which occurred due to spur.
5. If a bifurcation of any traveled branch is detected on the boundary of the window, eliminate that bifurcation and the candidate bifurcation which occurred due to bridge.
6. If two bifurcations are detected along traveled branches within or on the boundary of the window, eliminate that bifurcations and the candidate bifurcation which occurred due to ladders.

Elimination of False Minutiae due to Holes/Islands. Islands are closed paths these are basically false ridge bifurcations. For each candidate bifurcation point,

1. Take a window of size w x w.
2. Place the candidate ridge bifurcation at the center of window.
3. Move in the direction of connected branches of the bifurcation.
4. If one bifurcation is detected along two traveled branches within or on the boundary of the window, eliminate that bifurcation and the candidate bifurcation which occurred due to holes/islands.

Elimination of False Minutiae due to Breaks and Short Ridges. For each ridge ending point,

1. Take a window of size w x w.
2. Place the candidate ridge ending point at the center of window.
3. For that candidate ending point if there exits another ending point within the window then remove both of them as they occurred due to ridge break.
4. Move in the direction of connected branch of the ridge ending.
5. If another ridge ending is detected within the window, eliminate that ending and the candidate ending which occurred due to short ridges.

Figure 4 shows most common false minutiae structures.

3.2 Minutiae Validation

Here we present a technique to validate a minutiae (ridge ending and bifurcation).

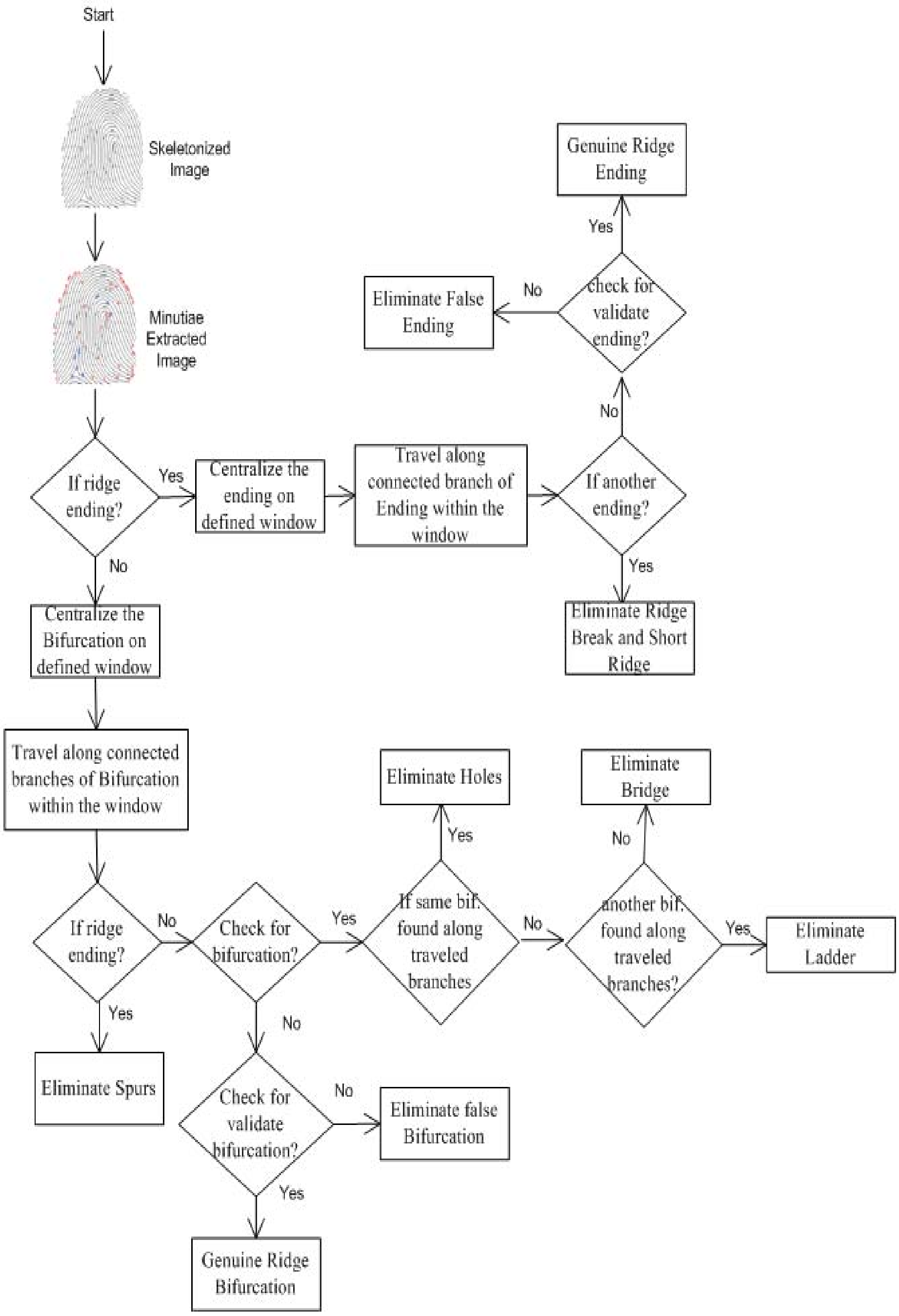

Fig. 3. Flow Chart of Postprocessing Technique

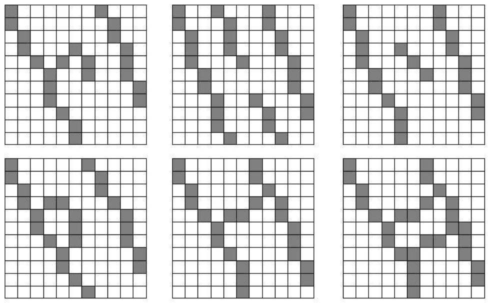

Fig. 4. Most common false minutiae structures, From left to right: Ist row: Spur, Ridge break and Short ridge, 2nd row: Hole/Island, Bridge and Ladder

Ridge Ending Validation. For each candidate ridge ending point,

1. Take a window of size w x w.
2. Place the candidate ridge ending point at the center of window.
3. The central pixel of window where endpoint is located should be initialized with -1 (Fig. 5).
4. Move in the direction of connected branch of the ending point. Assign 1 to all the pixels of the defined window while traveling along that connected branch (Fig.5).
5. Move all around the border of the window in clockwise direction.
6. Count 0 to 1 transitions while traversing along the border.
7. If count = 1, then the candidate ridge ending is validated as a genuine ridge ending.

Ridge Bifurcation Validation. For each candidate bifurcation point,

1. Take a window of size w x w.
2. Place the candidate ridge bifurcation at the center of window.
3. The central pixel of window where bifurcation is located should be initialized with -1 (Fig. 6).
4. Move in the direction of connected branches of the bifurcation. Assign 1,2 and 3 to all the pixels of the defined window while traveling along three different connected branches respectively (Fig. 6).

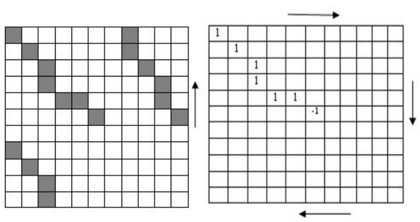

Fig. 5. From left to right: Ridge ending Validation, after applying values to connected branch

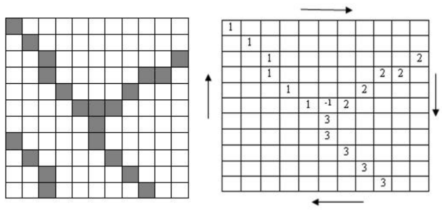

Fig. 6. From left to right: Ridge bifurcation Validation, after applying values to connected branches

5. Move all around the border of the window in clockwise direction.
6. Count 0 to 1, 0 to 2 and 0 to 3 transitions while traversing along the border.
7. If count = 3, then the candidate bifurcation is validated as a genuine bifurcation.

4 Experimental Results

The performance of our proposed technique is tested on FVC2004 [14] database. The database contains 40 different fingers and 8 impressions of each finger (40x8=320 fingerprints). The images in DB1, DB2, DB3 and DB4 are 640×480, 328×364, 300×480 and 288×384 respectively and each having a resolution of 500 dpi. Minutiae are extracted from three sets of images: (i) Direct skeletonized images (Raw Images), (ii) Segmented and then skeletonized images (iii) Enhanced skeletonized images using described preprocessing. Then our postprocessing algorithm is applied on that extracted minutiae and results are shown in Table 1. Table 1 shows the average number of minutiae and their standard deviation before and after applying our proposed technique. The goal of postprocessing is to minimize the number of false minutiae and to increase the probability of fingerprint matching. Table 2 shows the step by step statistical results of proposed technique when applied on preprocessed images and it gives average number of ridge endings and bifurcations. Figure 7 shows the minutiae extracted using cross number method and reduction in number of minutiae after applying postprocessing technique.

Table 1. Postprocessing Results I

Type of Image	Before Average	Before Std. Dev.	After Average	After Std. Dev.	Reduction Factor (%)
Raw Image	971	523	51	21	94.74
Segmented Image	536	271	47	19	91.23
Preprocessed Image	321	129	29	12	90.96

Table 2. Postprocessing Results II

Technique	*Endings Average*	*Bif. Average*	*Endings Red.(%)*	*Bif. Red.(%)*	*Total Red.(%)*
Feature Extraction	250	130	-	-	-
Ridge Breaks	203	130	18.8	0.0	12.3
Bridge, Ladder and Spur	164	57	19.2	56.1	33.6
Short Ridge	112	57	31.7	0.0	23.5
Island/Hole	112	36	0.0	36.8	12.4
Validate end point	25	36	77.6	0.0	58.7
Validate bifurcation	25	10	0.0	72.2	42.6

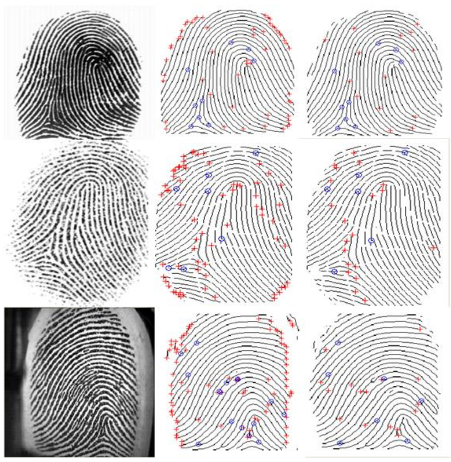

Fig. 7. 1st Col:Original Images, 2nd Col: Minutiae extracted using CN method, 3rd Col: Post-processed filtered Minutiae

5 Conclusion

In this paper, a windowing postprocessing technique for false minutiae elimination from skeletonized fingerprint image is proposed. The problem of minutiae extraction from skeletonized images has been derived from the fact that there are thousands of apparent minutiae, while actual are less than 100. Fingerprint images are enhanced using preprocessing and minutiae are extracted from them. Proposed technique successfully eliminates the false minutiae structures like spurs, bridges, ladders, ridge breaks, short ridges and holes/islands. The statistical analysis of the results obtained by the proposed approach shows the effective reduction of false minutiae. A very useful effect of the proposed minutiae extraction algorithm is that it performs correctly in dirty areas and dry images. The results are confirmed by visual inspection of images extracted from the FVC2004 fingerprint image database.

References

1. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, Heidelberg (2003)
2. Federal Bureau of Investigation: The science of fingerprints: classification and uses. U.S. Government Printing Office Washington DC (1984)
3. Osterburg, J., Parthasarathy, T., Raghavan, T.E.S., Sclove, S.L.: Development of a mathematical formula for the calculation of fingerprint probabilities based on individual characteristics. J. Am. Stat. Assoc. 72, 772–778 (1977)
4. Andrew Sr., W., Bolle, R.M., Ratha, N., Pankanti, S.: Fingerprint minutiae: a constructive definition. In: IEEE ECCV workshop on biometrics (2002)
5. Stosz, J.D., Alyea, L.A.: Automated system for fingerprint authentication using pores and ridge structure. In: Proc. SPIE 2277 (1995)
6. Chen, Z., Kuo, C.H.: A topology-based matching algorithm for fingerprint authentication 2 (1992)
7. Malleswara, T.C.: Feature extraction for fingerprint classification. Pattern Recognition 8 (1976)
8. Fitz, A.P., Green, R.J.: Fingerprint classification using a hexagonal fast Fourier transform. Pattern Recognition 29 (1996)
9. Ratha, N.K., Chen, S., Jain, A.K.: Adaptive flow orientationbased feature extraction in fingerprint images. Pattern Recognition 28 (1995)
10. Akram, M.U., Tariq, A., Jabeen, S., Khan, S.A.: Fingerprint Image Segmentation based on Boundary Values. In: 3rd International Conference on Computer Vision Theory and Applications (VISAPP 2008), Portugal (January 2008)
11. Hong, L., Wan, Y., Jain, A.: Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence archive 20 (August 1998)
12. Paul, A.M., Mary Lourde, R.: A Study on Image Enhancement Techniques for Fingerprint Identification. In: IEEE International Conference on Video and Signal Based Surveillance (AVSS 2006) (2006)
13. Kovascs-Vajna, Z.M., Frazzoni, M., Rovatti, R.: Fingerprint ridge distance computation methodologies. Pattern Recognition (1997)
14. Finger print Verification Contest 2004 (FVC 2004), http://bias.csr.unibo.it/fvc2004.html