An Algorithm for Fingerprint Image Postprocessing

Marius Tico, Pauli Kuosmanen
Tampere University of Technology
Digital Media Institute
P.O.BOX 553, FIN-33101, Tampere, FINLAND
tico@cs.tut.fi

Abstract

Most of the current fingerprint identification and verification systems perform fingerprint matching based on different attributes of the minutia details present in fingerprints. The minutiae (i.e. ridge endings and ridge bifurcations) are usually detected in the thinned binary image of the fingerprint. Due to the presence of noise as well as the use of different preprocessing stages the thinned binary image contains a large number of false minutiae which may highly decrease the matching performances of the system. A new algorithm of fingerprint image postprocessing is proposed in this paper. The algorithm operates onto the thinned binary image of the fingerprint, in order to eliminate the false minutiae. The proposed algorithm is able to detect and cancel the minutiae associated with most of the false minutia structures which may be encountered in the thinned fingerprint image.

1. Introduction

Fingerprints have long been used for personal identification. It is assumed that every person possess unique fingerprints [2] and hence the fingerprint matching is considered one of the most reliable techniques of people identification.

A fingerprint image exhibits a quasiperiodic pattern of ridges (darker regions) and valleys (lighter regions). The local topological structures of this pattern together with their spatial relationships determine the uniqueness of a fingerprint. There are more than 100 different types of local ridge structures that have been identified [2]. Nevertheless, most of the automatic fingerprint identification / verification systems adopt the model used by the Federal Bureau of Investigation [9]. The model relies on representing only the two most prominent structures: ridge ending and ridge bifurcation, which are collectively called minutiae.

Several methods of automatic minutiae extraction from fingerprint images have been proposed in the literature [1, 4, 6, 8]. Although rather different from one other, most of the proposed methods perform first a segmentation of the fingerprint ridges followed then by a ridge thinning process which reduces the width of each ridge to one pixel. The candidate minutiae are located in those pixels of the thinned binary image where the number of outgoing branches is different of 2. Due to different types of noise which may be present in the fingerprint image (e.g. under-inking, over-inking, scars or excessively worn prints) as well as due to the segmentation and thinning processes, a large number of false minutiae are encountered among the candidate minutiae detected in the thinned image. Therefore a postprocessing stage of minutiae purification is required before fingerprint matching.

The false minutiae may be identified in the thinned binary image either as part of false minutia structures (e.g. spikes, bridges, holes, breaks, spurs, ladder structures) or at the boundary of the image region where the fingerprint pattern is located (boundary effect). The problem of fingerprint image postprocessing for false minutiae elimination have been addressed before by different authors. Ratha, Chen, and Jain [6] use three heuristic rules in order to eliminate ridge breaks, spikes and boundary effect. Hung [3] propose a set of algorithms for detecting and removing spurs, holes and bridges. Hung's algorithms use the duality between the ridge and valley structures whose thinned versions are represented as graphs. A combined statistical and structural approach is proposed by Xiao and Raafat [10] in order to remove ridge breaks, and false bifurcations.

The postprocessing algorithm proposed in this paper is able to detect and cancel false minutiae which are included in false minutia structures like spikes, holes, bridges, ladder structures, and spurs. The algorithm analyzes the neighborhood of each candidate minutia in order to decide whether the minutia is false or not. The experimental results reveal that the proposed algorithm combined with a simple boundary effect treatment succeeds to reduce an important number of false minutiae encountered in the thinned ridge map image.

0-7803-6514-3/00/$10.00©2000 IEEE

The organization of the paper is as follows. This introduction serves as the first section. The following section will briefly review the most important types of false minutia structures as well as some of the main reasons these structures occur in the thinned ridge map image. The proposed algorithm is presented in Section 3. The experimental results are shown in Section 4 and finally some concluding remarks are presented in Section 5.

2. False minutia structures

The most common types of false minutia structures which may be encountered into a thinned fingerprint image are shown in Fig. 1. Each such structure generates two or more false minutiae.

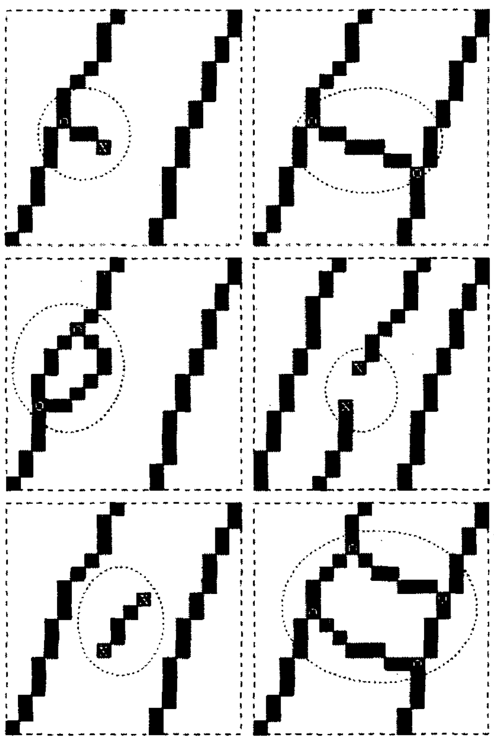

Figure 1. Types of false minutia structures. From left to right and up to bottom we have: spike, bridge, hole, break, spur, and ladder structure. The false minutiae generated by each structure are marked as (x) false ridge ending, and (o) false ridge bifurcation.

The spike structure generates two false minutiae and may occur when thinning a non-smooth ridge.

The bridge and ladder structures usually occur between close ridges.

Very wide ridges may generate hole structures and very wide valleys may generate spurs.

The presence of scars in the fingerprint may determine ridge breaks in the thinned ridge map image.

In addition a large number of false minutiae are always detected close to the boundary of the region of interest (boundary effect). The boundary effect is treated by cancelling all minutiae which are bellow a certain distance to the boundary of the fingerprint pattern.

It is of importance to mention that false minutia structures may also be caused by different image processing operations used to obtain the thinned ridge map image from the original gray-scale fingerprint image.

3. The proposed algorithm

The proposed algorithm operates directly onto the thinned ridge map image testing the validity of each candidate minutia.

The preprocessing operations used in order to transform the gray-scale fingerprint image into a thinned ridge map image are briefly explain in the following. The enhancement and segmentation of the gray-scale fingerprint image are carried out using the method proposed in our previous work [7]. The directional filters proposed there perform as low pass filters along the ridge direction and hence they succeed to remove most of the ridge breaks. As a result we obtain a binary ridge map image, where the ridge and valley pixels have value 1 and 0 respectively.

The ridge-map is further simplified by submitting it to a thinning process in order to reduce the width of each ridge to one pixel. The thinning algorithm presented in [5] (pp.333-336) has been used in our work.

The candidate minutiae are detected in a single scan of the thinned ridge map image using a 3×3 window. Each ridge pixel (value 1) is classified based on the number of 0 to 1 transitions met when making a full clockwise trip along its 8 neighborhood pixels. If there is only 1 such transition then the pixel is classified as a candidate ridge ending. If the number of transitions is 3 then the pixel is classified as a candidate ridge bifurcation. In all other cases the pixel is not included into the candidate minutiae set.

The number of candidate minutiae detected as above is much larger than the number of genuine minutiae in the fingerprint image. Most of the candidate minutiae have no meaningful correspondence into the fingerprint image, they being associated with different false minutia structures. The postprocessing algorithm presented in the following tests the validity of each candidate minutia by analyzing the thinned ridge map image in a $W \times W$ neighborhood of that minutia.

Algorithm

For each candidate minutia (ridge ending or ridge bifurcation):

1. Create and initialize with 0 an image L of size $W \times W$. Each pixel of L corresponds to a pixel of the thinned image which is located in a $W \times W$ neighborhood centered in the candidate minutia.
2. Label with -1 the central pixel of L (Fig.2a, Fig.3a). This is the pixel corresponding to the candidate minutia point in the thinned ridge map image.
3. If the candidate minutia is a ridge ending then:
   (a) Label with 1 all the pixels in L which correspond to pixels connected with the candidate ridge ending in the thinned ridge map image (Fig.2b).
   (b) Count the number of 0 to 1 transitions (T_{01}) met when making a full clockwise trip along the border of the L image (Fig.2c).
   (c) If $T_{01} = 1$, then validate the candidate minutia as a true ridge ending.
4. If the candidate minutia is a ridge bifurcation then:
   (a) Make a full clockwise trip along the 8 neighborhood pixels of the candidate ridge bifurcation, and label in L with 1, 2 and 3 respectively the three connected components met during this trip (Fig. 3b).
   (b) For each $\ell = 1, 2, 3$ (Fig.3 c,d,e), label with ℓ all pixels in L which:
      i. have the label 0;
      ii. are connected with an ℓ labeled pixel;
      iii. correspond to 1 valued pixels in the thinned ridge map;
   (c) Count the number of 0 to 1, 0 to 2 and 0 to 3 transitions met when making a full clockwise trip along the border of the L image. The above three numbers are denoted by T_{01}, T_{02} and T_{03} respectively as shown in Fig.3f.
   (d) If $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 1$, then validate the candidate minutia as a true ridge bifurcation.

The dimension W of the neighborhood analyzed around each candidate minutia is chosen larger then two times the average distance between two neighborhood ridges. In this way the algorithm succeeds to cancel close minutiae belonging to the same ridge.

Two examples of minutiae which are canceled by the proposed algorithm are shown in Fig.4.

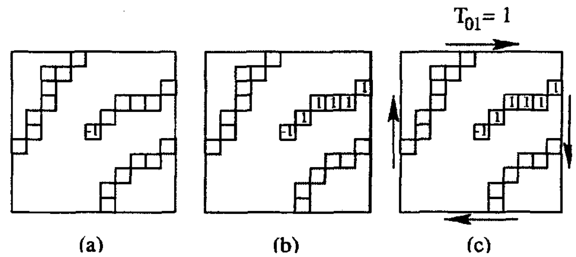

Figure 2. Example of minutia validation. The figure shows the changes in the L image after different steps of the algorithm.

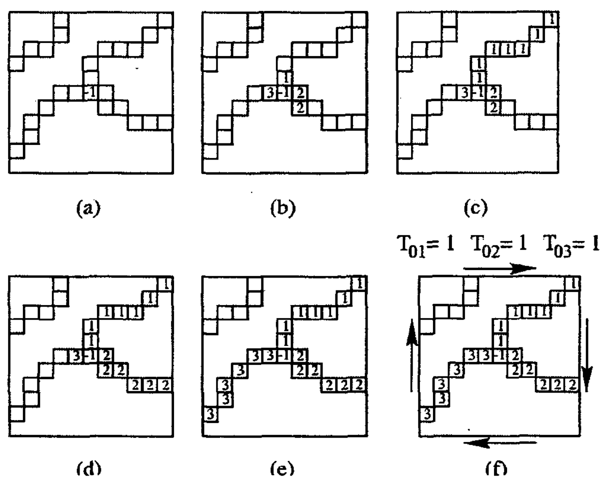

Figure 3. Example of minutia validation. The figure shows the changes in the L image after different steps of the algorithm.

4. Experimental results

A statistical analysis of the performances achieved by the proposed algorithm has been carried out using a number of 100 fingerprint images[1]. All images have the dimension 256×256 and are stored as 8-bit gray scale images.

The main performance criteria is given by the number of false minutiae cancelled by the proposed algorithm. The evaluation of the algorithm has been performed in two different preprocessing conditions: a) the fingerprint segmentation has been carried out using a local threshold operation

[1] The fingerprint images are from the database of Biometric Systems Lab., University of Bologna, Italy, (www.csr.unibo.it/research/biolab)

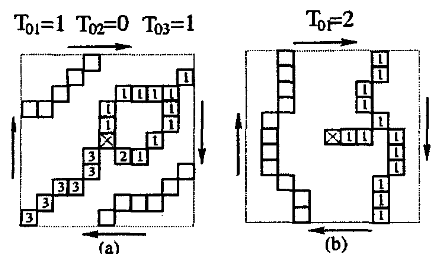

Figure 4. Example of false minutiae cancelation: hole configuration (a) and spike configuration (b).

with no prior filtering, and b) the method of fingerprint segmentation proposed in [7] has been used to obtain the binary ridge map image from the fingerprint gray scale image.

The two postprocessing methods are briefly explain in the following. In the method (a) a local threshold is computed as the average gray level in the neighborhood of each image pixel. The pixel is then classified as ridge (valley) pixel if its gray level is smaller (larger) than the local threshold. No other image enhancement (e.g. directional filtering) is used and hence the thinned ridge map image looks rather noisy containing a large number of false minutiae. An example of thinned ridge map image obtained as above is shown in Fig. 5. The second method (b) includes some directional enhancement. The image pixels are classified as ridge or valley pixels based on the sign of the principal curvature of the image surface, which is approximated using a set of precomputed directional filters. The method results in a quite accurate binary ridge map image and hence most of the false minutiae are already eliminated during this preprocessing stage.

The postprocessing algorithm has been successively applied onto each one of the 100 fingerprint images used for experiments. Each time the number of minutiae before and after postprocessing have been recorded. Statistical results obtained over the entire set of fingerprint images are shown in Table 1.

In order to study only the effect of the proposed algorithm the false minutiae due to the boundary effect have not been considered. The boundary effect have been treated by cancelling all minutiae within a distance smaller than 16 pixel from the boundary of the region of interest. Also, in the case the preprocessing method (a) is used, a simple ridge break treatment has been applied prior to the application of the proposed algorithm. This is, two ridge endings which are in a distance smaller than 10 pixels one to another, and have the same orientation are cancelled. The false ridge endings detected as above have not been considered among the minutiae encountered before the application of the algorithm.

The results in Table 1 reveal that the proposed algorithm is able to reduce an important number of false minutiae detected in the fingerprint image. Better performance are obtained in cancelling the false ridge bifurcations. The number of ridge bifurcations encountered after the application of the proposed algorithm is almost the same in both preprocessing cases. This suggest that for this kind of minutiae the postprocessing algorithm is less dependent of the method used for fingerprint segmentation. A less elaborate method with a lower computational complexity (as method (a)) suffices for this point of view. The number of false ridge endings is also decreased by the proposed algorithm. However, the performance are better in the case of using the preprocessing method (b). This is because this method succeeds to remove most of the ridge break configurations which the algorithm is not able to treat. The simple ridge break treatment used in the case of method (a) seems to be insufficient.

Table 1. The number of minutiae (average and standard deviation) before and after the application of the proposed algorithm.

Preprocessing	Minutia	Before	After
Method (a)	bifurcations	132.8(69.6)	10.4(4.8)
	endings	82.6(46.3)	37.3(11.4)
Method (b)	bifurcations	30(16.8)	10.5(4.3)
	endings	50.9(25.9)	28.6(7.6)

Two examples of minutiae validation using the proposed algorithm are shown in Fig. 5 and Fig. 6. The example shown in Fig. 5 uses the preprocessing method (a). For this particular example the proposed algorithm reduces the number of minutiae from 322 to 72. The example shown in Fig. 6 uses the preprocessing method (b). This method is able to cancel an important number of false minutiae. For example the scratch present in the upper part of the fingerprint would generate more than 10 false ridge endings. Nevertheless, because of the directional smoothing perform by the preprocessing method the scratch is deleted. For this example the proposed algorithm validates 21 minutiae from a number of 38 minutiae present before its application.

5. Conclusions

A new algorithm of fingerprint postprocessing has been proposed in this paper. The proposed algorithm is able to detect and cancel false minutiae associated with spikes, holes, bridges, ladder structures and spurs. The experimental results revel that the proposed algorithm cancels a

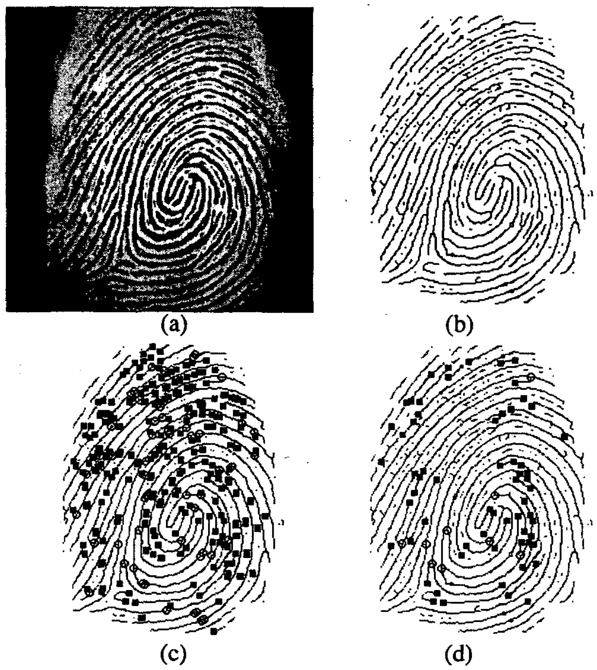

Figure 5. Example of minutiae validation when the image is segmented using a local threshold: the original image (a), the thinned ridge map image (b), the candidate minutiae over the thinned ridge map image (c), and the valid minutiae over the thinned ridge map image. The ridge endings and ridge bifurcations are marked with squares and circles respectively.

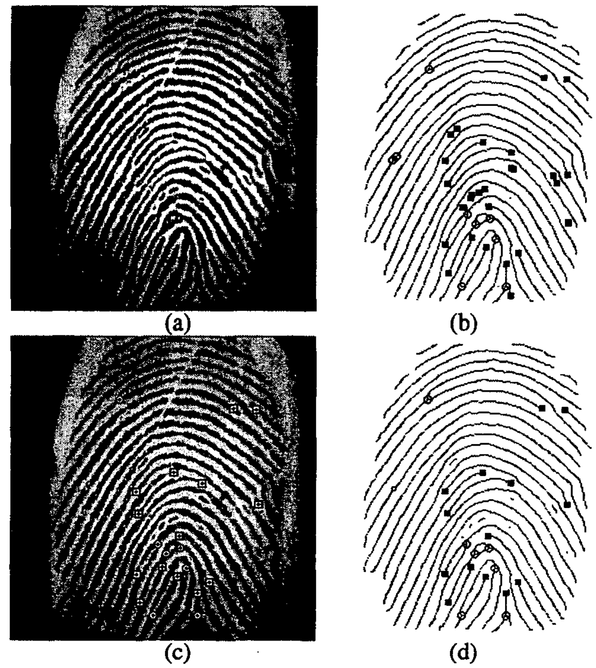

Figure 6. Example of minutiae validation when the image is segmented using the method proposed in [7]: the original image (a), the candidate minutiae over the thinned ridge map image (b), the valid minutiae over the original image (c), and the valid minutiae over the thinned ridge map image. The ridge endings and ridge bifurcations are marked with squares and circles respectively.

large number of false minutiae and in particular better performances are achieved in cancelling false ridge bifurcations. Currently we are working to develop more robust techniques for detecting and cancelling false ridge endings associated with ridge break structures.

References

[1] B.Moayer and K.S.Fu. A tree system approach for fingerprint pattern recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(3):376–387, Mar. 1986.

[2] H.C.Lee and R. Gaenssleh. *Advances in Fingerprint Technology*. Elsevier, New-York, 1991.

[3] D. Hung. Enhancement and feature purification of fingerprint images. *Pattern Recognition*, 26(11):1661–1671, Nov. 1993.

[4] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):302–313, Apr. 1997.

[5] I. Pitas. *Digital Image Processing Algorithms*. Prentice Hall, 1993.

[6] N. K. Ratha, S. Chen, and A. K.Jain. Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, Nov. 1995.

[7] M. Tico and P. Kuosmanen. A topographic method for fingerprint segmentation. *Proc. of The Int. Conf. on Image Processing (ICIP'99)*, pages 36–40, Oct. 1999.

[8] A. Wahab, S.H.Chin, and E.C.Tan. Novel approach to automated fingerprint recognition. *IEE Proc. Vis. Image Signal Processing*, 145(3):160–166, Mar. 1998.

[9] J. Wegstein. *An Automated Fingerprint Identification System*. US Government Publication, Washington, 1982.

[10] Q. Xiao and H. Raafat. Fingerprint image postprocessing: a combined statistical and structural approach. *Pattern Recognition*, 24(10):985–992, Oct. 1991.