A Novel Thinning Algorithm with Fingerprint Minutiae Extraction Capability

Sasan Golabi, Saiid Saadat, Mohammad Sadegh Helfroush, and Ashkan Tashk

***Abstract*—Automatic and reliable extraction of the minutiae from fingerprint images is a critical process in fingerprint matching and a main preprocess for this stage is Thinning. There are a lot of algorithms for fingerprint thinning procedure. All of the previously proposed thinning methods try to thin every ridge due to the content of its central pixel and then extracting minutiae based on some other algorithms for denoising and preventing false minutiae detections at islands or spurities. If an algorithm could thin fingerprint ridges except unrecoverable corrupted regions and also could eliminate noise, it will be considered as a good thinning one and no additional processes are needed before minutiae extraction. The proposed method of this paper has such abilities. The proposed algorithm is implemented by applying four boxes of matrices; each of them thins ridges due to a specific direction; i.e., diagonal, horizontal and vertical directions. The proposed algorithm also is able to thin discrete Latin Characters or symbols. For evaluating the proposed method, several robust and reliable experiments have been employed and the results confirm the higher ability of the proposed method in comparison with the other competing one.**

***Index Terms*—Fingerprint identification, minutiae recognition, thinning, spurities.**

I. Introduction

The fingerprints are probably the most widely used identification mechanisms in biometric systems. there are two factors affecting the high ratio of identification named as speed and reliably of minutiae extraction from the input fingerprint image. Minutiae points are local ridge characteristics that occur either at a ridge ending or bifurcation. A ridge ending is defined as the point where the ridge ends abruptly and the ridge bifurcation is the point where the ridge splits into two or more branches. Both kind of the named minutiae are shown in a sample fingerprint image as Fig. 1.

In AFIS (Automatic fingerprint identification systems), the fingerprint images are subjected to a series of operations. These operations consist of the following steps: [1]

- Fingerprint image enhancement.
- Binarization
- Obtaining of directional fingerprint image.

Manuscript received May 26, 2012; revised June 27, 2012. (Write the date on which you submitted your paper for review.) This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here).

Sasan Golabi, Mohammad Sadegh Helfroush, and Ashcan Tashk are with the Electrical and Electronic Engineering Department, Shiraz University of Technology, Shiraz, Iran (e-mail: s.golabi@ sutech.ac.ir, ms_helfroush@sutech.ac.ir, tash_ashkan@ieee.org).

Saiid saadat is with the Electrical and Electronic Engineering Department, Boushehr Islamaic Azad university, Boushehr, Iran (e-mail: sd_saadat@yahoo.com).

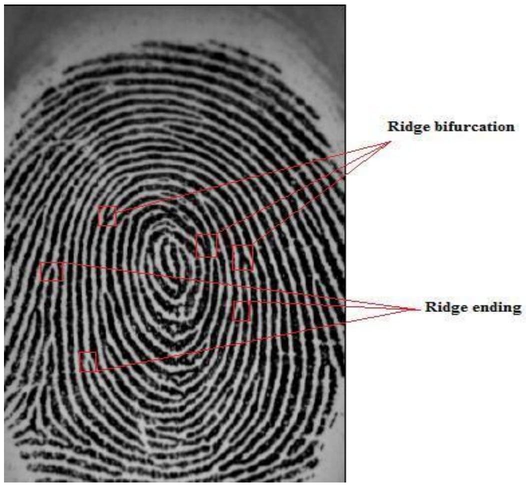

Fig. 1. Ridge ending and bifurcations for sample fingerprint image[2].

- Thinning.
- Minutiae extraction.

In this paper, the concentration is on the thinning process of any type of images and minutiae points extraction.

Thinning is a thickness reduction process applied to each line of a pattern till the line width becomes a single pixel. The requirements of a good thinning algorithm with respect to a fingerprint are as follows: [3]

- The image should be of single pixel width with no discontinuity after thinning.
- Each ridge should be thinned to its center pixel.
- Noise and singular pixel should be eliminated.
- Further removal of pixels is not possible after completion of thinning process.

A comprehensive survey of thinning algorithm is described in [4], [5] Proposed an algorithm for thinning any image, by applying some 3×3 matrices. In this paper, a modified thinning method is proposed that can be used to thin any symbol, Characters and also fingerprint images regardless of their shape and orientation. Usually in fingerprint images, Thinning seen as a preprocess for minutiae extraction. The Proposed algorithm identifies the unrecoverable corrupted areas in the fingerprint and does not thin them; this is an important advantage of the proposed method because such corrupted areas are extremely harmful to the extraction of minutiae points. Moreover, this advantage helps remove the spurious minutiae points which are harmful to fingerprint matching. The rest of this paper is organized as follows: Section II introduce the Thinning algorithm. Section III shows the algorithm applied to various symbols and its application on fingerprints. In Section IV, 32-rule based minutiae detection algorithm is introduced which is the modified form of the previously proposed ones. Section V includes the experimental and evaluation results. Finally the conclusion is presented in Section VI.

II. Proposed Thinning Algorithm

A. Summary of Proposed Algorithm

Thinning process is done with applying 4 box of matrices, Each box include three 3×3 matrices and one 4×4 matrix. First each box will be introduced as follow:

1) Diagonal matrices

This box consists of four 3×3 matrices, which are built for thinning diagonal lines. If any two pixels lie adjacent to each other at an angle of 45° and a third pixel connects both these pixels at 90°, then the third pixel should be eliminated and continuity of image will be preserved. This point leads to the formation of four diagonal matrices as illustrated in Fig.2.

2) Horizontal matrices

For thinning horizontal lines, we use horizontal matrices. Lines should be thin from down to up. This leads to the formation of four matrices as illustrated in Fig. 3.

3) Vertical matrices

Thinning of vertical lines could be done with applying another four matrices which thin lines from right to left.Fig.4.

4) Final matrices

The thinned image should not have noise and spurious, so after applying all the above matrices, thinning should be continued. This leads to the formation of three matrices as illustrated in Fig. 5.

B. Implementation of Proposed Algorithm

First apply the horizontal box to image .Do this for the first matrix in the box and repeat until no changes occur from one iteration to the next .Repeat this procedure for second matrix in the box. Then do the same for the third matrix and finally for the fourth matrix in the box. Now vertical matrices are applied to previous result, as explained for horizontal ones. Thinning will be continued by applying diagonal matrices and this procedure is just like horizontal ones. Now, noise and spurious will be eliminated by applying the final matrices to result of previous steps, and thinned image will be achieved.

III. Experimental Results

As explained above, proposed algorithm do not thin closed lines with small sizes. This application of algorithm is useful in fingerprint identification. In addition, proposed algorithm could thin any images, characters and symbols. If we don't have small size closed lines in the image, this thinning will satisfy the conditions for a thinning which mentioned in the introduction.

To have a well-form image, we do an additional step and apply the matrix which is introduced in Fig. 7, to result of proposed algorithm. Afterwards, noise (single pixels) could be eliminated by applying final matrices or with MATLAB internal functions. In the Fig. 8 and Fig. 9, advantages of proposed method are shown with more details.

As seen, in regions with good quality or no islands, proposed algorithm has better form with less noise. In regions with islands, our algorithms do not thin the image, and as discussed before, this is important for minutiae extraction procedure.

Original matrix			Modified matrix			Original matrix			Modified matrix		
x	1	x	x	1	x	0	0	x	0	0	x
1	1	0	1	0	0	0	1	1	0	0	1
x	0	0	x	0	0	x	1	x	x	1	x
x	1	x	x	1	x	x	0	0	x	0	0
0	1	1	0	0	1	1	1	0	1	0	0
0	0	x	0	0	x	x	1	x	x	1	x

Fig. 2. Diagonal matrices (these matrices are proposed in [5]).

Original matrix			Modified matrix			Original matrix			Modified matrix		
1	1	1	1	1	1	1	1	x	1	1	x
x	1	x	x	0	x	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Original matrix			Modified matrix		
x	1	1	x	1	1
0	1	0	0	0	0
0	0	0	0	0	0

Original matrix				Modified matrix			
1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
x	x	x	x	x	x	x	x

Fig. 3. Horizontal matrices (they thin horizontal lines).

Original matrix			Modified matrix			Original matrix			Modified matrix		
0	x	1	0	x	1	0	0	1	0	0	1
0	1	1	0	0	1	0	1	1	0	0	1
0	x	1	0	x	0	0	0	x	0	0	x

Original matrix			Modified matrix		
0	0	x	0	0	x
0	1	1	0	0	1
0	0	1	0	0	1

Original matrix				Modified matrix			
x	0	0	1	x	0	0	1
x	0	1	1	x	0	0	1
x	0	1	1	x	0	0	1
x	0	0	1	x	0	0	1

Fig. 4. Vertical matrices (they thin vertical lines).

Original matrix			Modified matrix			Original matrix			Modified matrix		
x	0	0	x	0	0	0	0	1	0	0	1
1	1	0	1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	1	1	x	1	1	x
0	0	0	0	0	0						
0	1	0	0	0	0						
0	0	0	0	0	0						

Fig. 5. Final matrices for noise elimination.

Fig. 6. An image with a 0.01% Salt and Pepper Noise: (a) original image and the thinned images done by: (b) MATLAB internal function, (c) proposed algorithm by P. M. Pradeep et al. and (d) the proposed algorithm in this paper. (As proposed algorithm has more performance for elimination of noise).

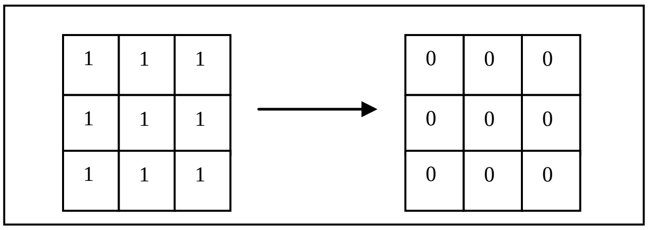

Fig. 7. Final matrix for in the thinning process.

If we deal with fingerprint images with no islands proposed algorithm will thin image with less noise and spurious than MATLAB function. So, this algorithm can be applied to any type of fingerprint images and also for low quality and high quality ones, have better results than thinning with MATLAB internal function.

IV. Minutiae Detection

The next step after image thinning is the extraction of minutiae. 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. If we consider all of this conditions, a minutiae and it's neighborhood form one of the matrices are shown in Fig.10,11. Note here ridges are shown with zeros(black) and valleys are shown with ones (white).

V. Comparison of Results

To evaluate our method, we compare minutiae extraction in fingerprint images thinned by proposed algorithm with fingerprint images thinned by MATLAB internal function. A random point is selected in these images and in 11 neighborhood of it, minutiae points are extracted. Equations used for comparing are:

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad (1)$$

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \quad (2)$$

$$F - Measure = \frac{2 \times \mathrm{Re}call \times \mathrm{Pr}ecision}{\mathrm{Re}call + \mathrm{Pr}ecision} \quad (3)$$

where TP, FP and FN stand for True Positive, False Positive and False negative, respectively. On the other hand, TP shows the number of minutiae extracted properly, FN is the number of minutiae which are missed and FP is the number of unreal detected minutiae.

To use these definitions, in masks that (TP) was zero, we set it equal to (0.5). Fingerprint images are randomly selected from FVC2004. experimental results show the best advantages of proposed algorithm for low quality images.

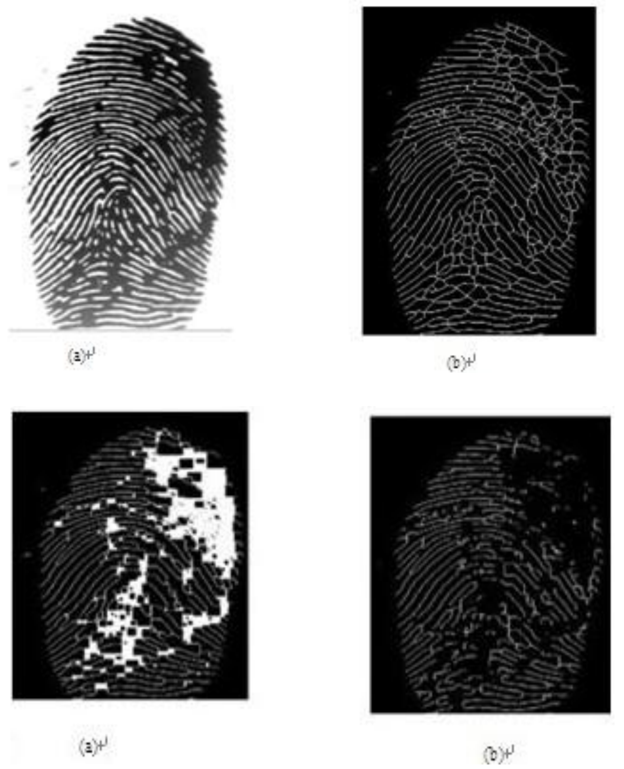

Fig. 8. Thinning a poor quality fingerprint image: (a) the original image, (b) thinned one with the MATLAB internal function (there's a lot of islands), (c) the thinned one with the proposed algorithm (unrecoverable regions are detected and are not thinned. So there would not be any island areas in the thinned image), (d) fingerprint image after applying a ones matrix and noise elimination.

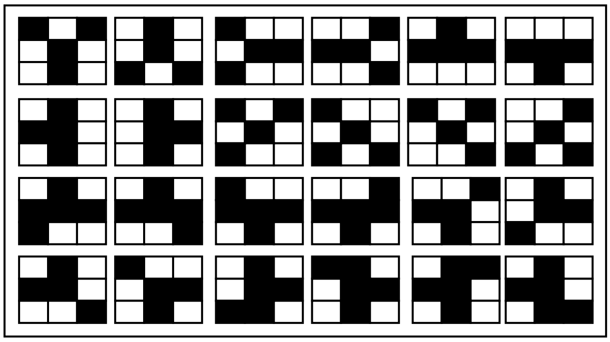

Fig. 9. Masks for all bifurcation possible, [5].

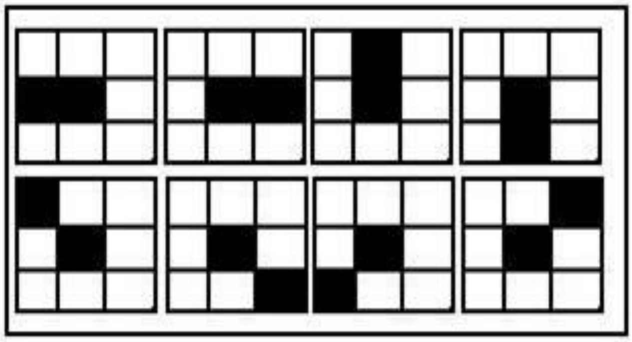

Fig. 10. Masks for all ridges ending possible.

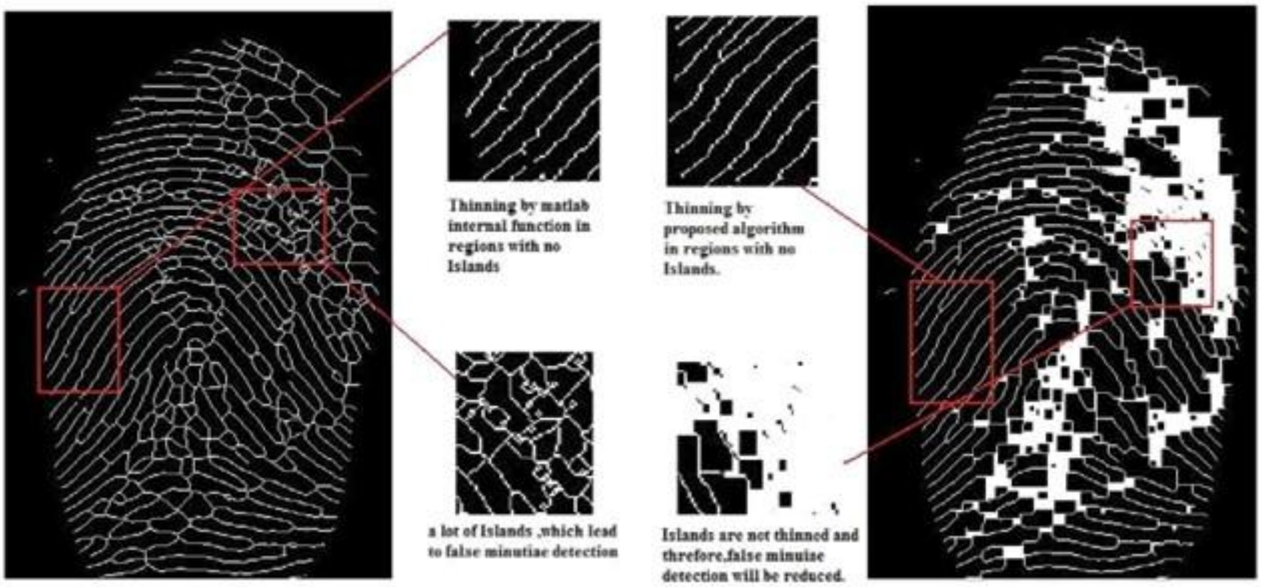

Fig. 11. Comparing two thinning method: (a) thinning by MATLAB internal function. (b) Thinning by proposed algorithm. White regions in this image, are unrecoverable areas of image, so the algorithm don't thin them. As illustrated, this avoid of creation Islands and thus, we will have minutia detection with more accuracy. It's shown that, in other regions, thinning is better than MATLAB function one.

VI. Conclusion

Results show that proposed algorithm detects almost all the minutiae with lowest FP number. In low quality fingerprint images, unrecoverable regions have not been thinned and this leads to more accuracy in minutiae extraction procedure. In addition, for high quality fingerprints, proposed algorithm has better performance in minutiae extraction than MATLAB thinning function. Another advantage of proposed algorithm is its usage in thinning of any image like symbols and characters. Elimination of noise and spurious in proposed algorithm is high and its implementation, is so simple.

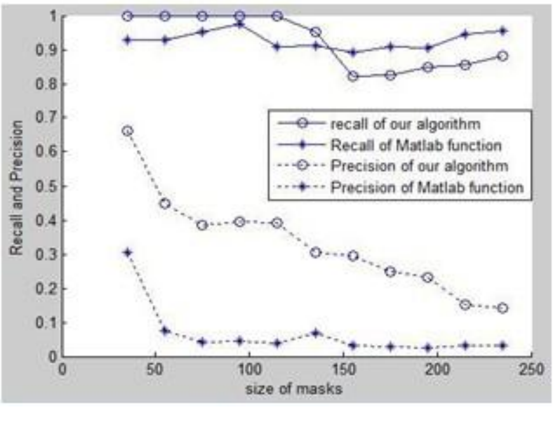

Fig. 12. Compare curves for proposed algorithm and internal MATLAB thinning function: recall and precision curves.

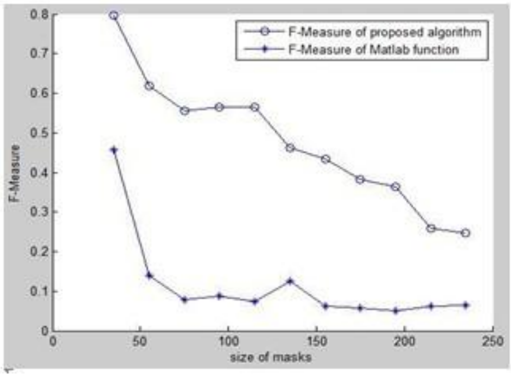

Fig. 13. Compare curves for proposed algorithm and internal MATLAB thinning function: the F-Measure curves for both the proposed algorithm and the MATLAB thinning function.

References

[1] D. Maio and D. Maltoni, "Direct gray –scale minutiae detection in fingerprints," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, pp. 27-40, Jan 1997.

[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint Recognition*

[3] L. Lam, S. Lee, and C. Suen, "Thinning methodologies-A Comprehensive survey," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 369, 387, 2000.

[4] M. Ahmad and R. Ward, " A rotation Invariant Rule-based Thinning Algorithm for character recognition," *IEEE trans on pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp.1672-1680, Dec 2002

[5] P. M. Patil, S. R. Suralkar, and F. B. Sheikh, "Rotation Invariant Thinning Algorithm to Detect Ridge Bifurcations for fingerprint Identification," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, 2005.