Effective and Efficient Fingerprint Image Postprocessing

Haiping Lu, Xudong Jiang and Wei-Yun Yau

Laboratories for Information Technology
21 Heng Mui Keng Terrace, Singapore 119613
Email: hplu@lit.org.sg xdjiang@lit.org.sg wyyau@lit.org.sg

Abstract

Minutiae extraction is a crucial step in an automatic fingerprint identification system. However, the presence of noise in poor-quality images causes a large number of extraction errors, including the dropping of true minutiae and production of false minutiae. A study on these errors reveals that postprocessing is effective in removing false minutiae while keeping true ones. Furthermore, the overall processing efficiency could be improved because of the reduction in total minutia number. In this paper, we present a novel fingerprint image postprocessing algorithm. It is developed based on several rules, which are generalized through a study on the errors that commonly occur in minutiae extraction and their effects on the overall verification performance. Thorough experimental tests demonstrate the proposed postprocessing algorithm to be both effective and efficient.

1 Introduction

Most fingerprint identification systems are based on minutiae matching [1], and there are two minutia structures that are most prominent: ridge endings and ridge bifurcations [2]. For fingerprint images of poor quality, a large number of spurious minutiae are often extracted due to noise. This problem could be tackled either in the preprocessing stage such as fingerprint image enhancement [2,3], or in the postprocessing stage [4,5]. Fingerprint image postprocessing has been addressed in the literature by several authors. Xiao and Raafat propose a combined statistical and structural approach [4]. Hung exploits the duality between ridge images and valley images to detect and remove false minutiae [5]. A neural network based minutiae filtering method is proposed by Maio and Maltoni [6]. Ratha et al. use three heuristic rules to eliminate false minutiae [7]. In [8], Farina et al. present a set of postprocessing algorithms, including some classical methodologies and some new approaches such as ending and bifurcation validation algorithms.

Spurious minutiae are produced in minutiae extraction due to noisy ridge structures in a fingerprint image. To some extent, they are also related to the specific minutiae extraction algorithm used, as discussed in [1]. Consequently, a postprocessing algorithm, aiming at correcting extraction errors, is related to the extraction algorithm, and optimal postprocessing algorithms could be different for different minutiae extraction algorithms.

In this paper, we propose a new postprocessing algorithm that is developed for the minutiae extraction algorithm in [1]. Efforts are made to preserve true minutiae while eliminating false minutiae. Several rules for postprocessing have been generalized and used as guide in the development. These rules and the approaches taken are applicable to postprocessing for other minutiae extraction algorithms as well.

2 Postprocessing algorithm

Under noisy conditions, a minutiae extraction algorithm could erroneously detect spurious minutiae, or make errors in deciding minutia types and localizing minutiae. Postprocessing can minimize these errors by removing false minutiae, correcting wrongly classified minutia types and increasing precision in minutia localization. At the same time, efforts should be taken to avoid removal of true minutiae during postprocessing.

The new postprocessing algorithm is developed for the minutiae extraction algorithm presented in [1], where the gray-level fingerprint image is traced adaptively, and each valid ridge traced is associated with a ridge number m. If a tracing ridge m intersects another traced ridge l, a bifurcation is detected. An ending is detected when the tracing of a ridge m stops with no other ridge intersection (in this case, l is denoted as zero). A detected minutia is described by its position (i, j), its direction φ and the two associated ridge numbers m, l.

The postprocessing algorithm proposed deals with the minutia list after extraction, using minutia information and the skeleton image $\mathbf{S}$. The original gray-level fingerprint image is referred to when needed. Another useful parameter is the average ridge distance d in pixels.

2.1 Bifurcation correction

Noises in a fingerprint image could cause an ending to be detected as a bifurcation, or result in a false ridge break with one bifurcation and one ending, as shown in Fig. 1.

This kind of spurious bifurcations usually have one highly curved branch and could be identified by checking two direction conditions: $\alpha < 2\pi/3$ and $\beta > \pi/3$, where α and β are as shown in Fig. 2a. If a bifurca-

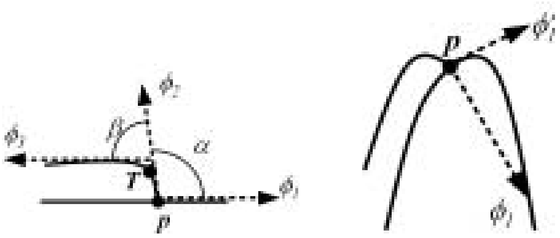
(a) (b)

Figure 1: Bifurcations to be corrected

tion satisfies these two conditions, its minutia type is changed to ending and its minutia position is changed to the position of T, the turning point found by tracking the direction change in tracing. In Fig. 2a, directions ϕ_1, ϕ_2 and ϕ_3 are calculated as the directions of corresponding lines. Each of these lines connects a start point of tracing and an end point of tracing, pointing to the end point of tracing. The start point of tracing for both ϕ_1 and ϕ_2 is the bifurcation point p, and that for ϕ_3 is T. T is also the end point of tracing for ϕ_2.

In addition, it should be noted that most bifurcations near the core, where ridges are highly curved, are likely to satisfy the two conditions if ϕ_1 is calculated from tracing a very short distance as $\phi_1\prime$, shown in Fig. 2b. Thus, the distance traced to calculate ϕ_1 should be large enough so that this kind of true bifurcations won't be corrected mistakenly. This distance is set to $3d$ in our postprocessing.

This processing contributes to not only the correction of minutia type, but also a more accurate localization. Furthermore, after this processing, the spurious minutiae caused by a break as in Fig.1b are converted to a pair of facing endings, which could be identified and eliminated in the ending pair processing later.



(a) (b)

Figure 2: Bifurcation correction

2.2 Near-edge ending handling

Due to imperfect segmentation, false minutiae could be present near edges. In [7], this is called "boundary effects" and the solution is to delete minutiae detected within a specified border of the foreground boundary, but a number of true minutiae could be deleted this way.

It is observed that most false bifurcations near edges could be corrected to endings by the bifurcation correction introduced above. Therefore, problems near edges could be reduced to handling of endings.

From Fig. 3, line po is orthogonal to φ_p at a small offset from the ending p. It is traced in both directions (ϕ_{o1} and ϕ_{o2}) for a distance of $2d$ in the skeleton image **S**, and the tracing stops if any other ridge is met. The ending is possibly false if an edge is met when tracing, e.g. in ϕ_{o1}. However, this condition alone is not reliable enough to claim that the ending near edge is false because of possible segmentation problems. Hence, the original gray-level image is used to further verify the validity of the ending. The gray-level image is traced in ϕ_{o1}, searching first for a valley by comparing intensities, and then for a ridge in a similar manner. The ending will be classified as false only when no ridge is found while a valley is detected.

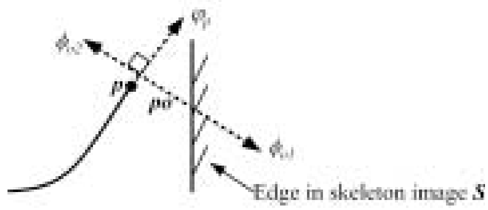
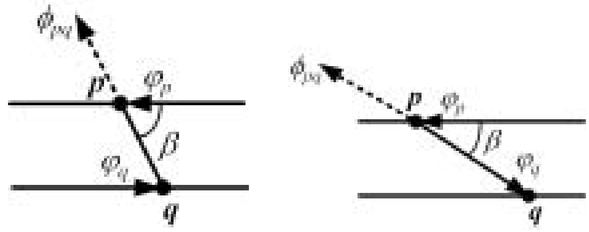


Figure 3: Near-edge ending handling

2.3 False bifurcation pair processing

False bifurcation pairs include conventional bridges, crosses, and islands. False islands could be easily identified and removed. Bridges and crosses are commonly due to noise between nearby ridges with similar orientations and they can be classified as type A or type B, as shown in Fig. 4, where p and q are a bifurcation pair. Such bifurcation pairs are usually facing each other, and they are possibly false when $\Delta\varphi_{pq} > 3\pi/4$, where $\Delta\varphi_{pq} = |\varphi_p - \varphi_q|$ if $|\varphi_p - \varphi_q| \leq \pi$ and $\Delta\varphi_{pq} = 2\pi - |\varphi_p - \varphi_q|$ if $|\varphi_p - \varphi_q| > \pi$.

Next, if these two bifurcation points are quite close with distance $|pq| \leq d$, they are very likely to be false and hence eliminated. Otherwise, if $d < |pq| < 1.5d$, β needs to be checked, which is the angle between φ_p and ϕ_{pq} , the direction of line pq. For type A, this pair are false if $\beta < \pi/8$. For type B, the pair are false only if $\beta < \pi/12$, which is more strict to avoid removal of true minutiae.



(a) Type A: $l_p = l_q$ (b) Type B: $l_p = m_q$

Figure 4: False bifurcation pair processing

2.4 False ending pair processing

False ending pairs are commonly due to ridge breaks and scars. For an ending p, if there are several other endings that could form a pair with it, the nearest one q is chosen. The line pq is traced first. These two endings are true if there is any other ridge met during the tracing. When there is no ridge in between, they are eliminated if they are very close with distance $|pq| \leq 0.5d$, and if $|pq| > 0.5d$, they are possibly false for $\Delta\varphi_{pq} > 2\pi/3$ since such an ending pair are in opposite directions mostly.

Next, β, the angle between φ_p and ϕ_{pq}, needs to be considered. For $0.5d < |pq| \leq d$, p and q are deleted if $\beta > 2\pi/3$. For $d < |pq| \leq 1.8d$, they are deleted only if $\beta > 5\pi/6$.

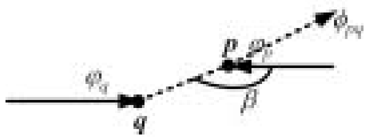


Figure 5: False ending pair processing

2.5 Other techniques employed and processing sequence

We also employed some other simple techniques. False minutiae detection using these techniques is based on distance and direction relations, and ridge information. These techniques include very-close minutia pair deletion, facing-spur and against-spur handling, and a special treatment for fingerprint images of very poor quality. A spur consists of a bifurcation p and an ending q within a specific distance. It is a facing-spur if $\Delta\varphi_{pq} > 3\pi/4$, and it is an against-spur otherwise. From our observation, facing-spurs are more likely to be false. Thus, the distance threshold for facing-spurs is larger than that for against-spurs, which is a looser condition. The special treatment for very poor images is to remove all minutiae within 12 pixels distance from edges in the skeleton image $\mathbf{S}$.

The processing sequence could have a significant impact on the overall performance. As seen in the description of bifurcation correction and near-edge ending handling, various techniques are not totally independent and one could have some effects on another. On the other hand, some techniques are more reliable than others. Therefore, to achieve the best performance, the more reliable a processing technique is, the earlier stage it should be put in the postprocessing sequence. This is also verified in experiments. Hence, various processing techniques are arranged in the following sequence:

(1) Very-close minutia pair deletion

(2) Bifurcation correction

(3) Near-edge ending handling

(4) Facing-spur handling

(5) False bifurcation pair processing

(6) False ending pair processing

(7) Against-spur handling

(8) Special treatment for very poor images

2.6 Guiding rules

In the development of these postprocessing techniques, some guiding rules are followed, as detailed below.

Firstly, although the target is to remove false minutiae, keeping true minutiae is more important for reliable minutiae matching, based on our testing. In practice, a technique designed to remove false minutiae is likely to remove some true minutiae as well. Thus, a postprocessing technique will only be adopted if it removes significantly more false minutiae than true minutiae.

Secondly, the setting of parameters, mostly some thresholds, should be related to the image to be processed if possible. For instance, it is advantageous to relate distance thresholds to the average ridge distance d. Parameters adaptive to the image usually perform better than fixed values.

Thirdly, false minutiae are preferable to be processed either solely or in pairs since most of them could be viewed as occurring solely or in pairs. Even in the case of a scar where many false minutiae present, it is still advantageous to process in pairs to avoid removing true minutiae, especially when there is any true minutia near the scar. A pair of false minutiae will be removed once detected and both will not be considered in following processing.

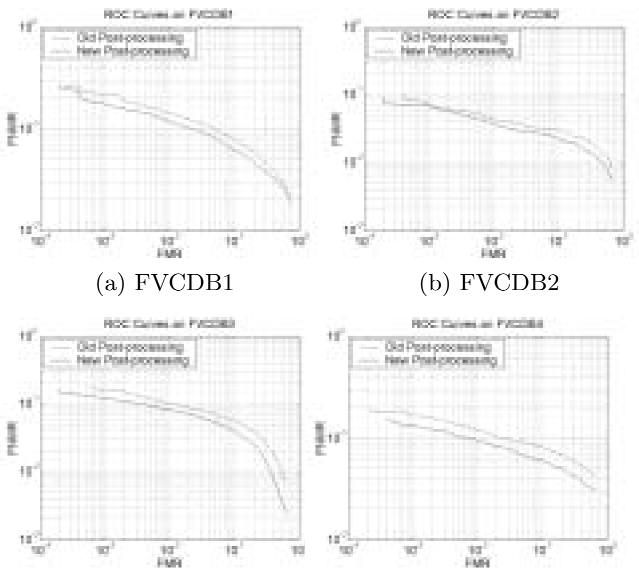
Next, for some complex types of false minutiae, some processing rules could be over-simplified. They are not sufficient to tackle these problems effectively to achieve better performance. Hence, it would be beneficial to introduce several layers of processing, as illustrated in near-edge ending handling, false bifurcation pair processing and false ending pair processing.

Lastly, as described previously, more reliable processing techniques should be placed at an earlier stage.

3 Performance evaluation

The performance of this new postprocessing algorithm is evaluated in a similar way as in the Fingerprint Verification Competition (FVC) 2000 [9]. Four databases used in FVC2000 with 800 fingerprint images per database are tested to evaluate the accuracy and efficiency. They are called DB1, DB2, DB3, and DB4 in the following discussions. The performance of the new postprocessing algorithm is compared with that of the old postprocessing algorithm presented in [1].

The effectiveness of the new postprocessing algorithm is shown by the improvement in fingerprint verification accuracy. As in FVC2000 [9], the Receiving Operating Curves (ROC) are plotted and shown in Fig. 6, and the Equal Error Rates (EER) are computed and presented in Table 1.



(a) FVCDB1 (b) FVCDB2

(c) FVCDB3 (d) FVCDB4

Figure 6: ROCs for FVC databases

In Table 1, EER for the old postprocessing algorithm is denoted as $OldEER$ and EER for the new algorithm proposed in this paper is denoted as $NewEER$. The improvement achieved is called $EERImprove$, and $EERImprove = (OldEER - NewEER)/OldEER$.

Table 1: EER improvement

FVCDB	OldEER	NewEER	EERImprove
DB1	7.87%	6.84%	13.15%
DB2	3.38%	2.67%	20.96%
DB3	5.93%	4.87%	17.84%
DB4	7.60%	5.84%	23.19%
Average	6.20%	5.06%	18.42%

It could be seen from both ROCs and EERs that the new postprocessing algorithm has effectively improved the accuracy of the minutiae matching algorithm.

The efficiency is evaluated by the processing time. One is the average minutiae extraction time $TExtract$, which includes postprocessing time, and the other is the average minutiae matching time $TMatch$. The processing time shown in Table 2 is the five times average running under the same PC environment. In the table, $Post\% = TimeSpentOnPostprocessing/TExtract$.

As shown in the table, for the new postprocessing, the extraction time is increased by only a very small amount while the matching time is reduced slightly, on average. It should be noted also that the time spent on the new postprocessing is still negligible on average ($< 0.5\%$), compared with the total extraction time. Hence, the new postprocessing algorithm does not affect the efficiency of our fingerprint verification algorithm much, which ranks top in FVC2000 [9].

Table 2: Processing time

FVCDB	TExtract(ms)		TMatch(ms)		Post%	
	Old	New	Old	New	Old	New
DB1	63.4	63.6	1.03	0.97	0.16	0.64
DB2	68.4	68.5	0.85	0.88	0.09	0.61
DB3	153.7	154.4	1.80	1.76	0.16	0.45
DB4	56.8	57.0	0.40	0.42	0.06	0.31
Average	85.6	85.9	1.02	1.01	0.13	0.49

4 Conclusions

In this paper, an effective and efficient postprocessing algorithm has been proposed for the minutiae extraction algorithm in [1]. To improve the overall performance of an automatic fingerprint identification system, it is very important to preserve true minutiae while removing spurious minutiae in postprocessing. Thus, our new postprocessing algorithm makes efforts to reliably differentiate spurious minutiae from true ones. These efforts include making use of ridge number information, referring to the original gray-level image, designing and arranging various processing techniques properly, and selecting various processing parameters carefully. The experimental results have shown that the postprocessing algorithm proposed has effectively improved the verification accuracy with little effect on efficiency, compared with the previous postprocessing algorithm.

References

[1] X. D. Jiang, W. Y. Yau, and W. Ser, “Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge,” *Pattern Recognition*, vol. 34, no. 5, pp. 999–1013, May 2001.

[2] L. Hong, Y. Wan, and A. K. Jain, “Fingerprint Image Enhancement: Algorithm and Performance Evaluation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, Aug. 1998.

[3] X. Luo, and J. Tian, “”Knowledge Based Fingerprint Image Enhancement,” in *Proceedings of Int. Conference on Pattern Recognition*, pp. 783–786, 2000.

[4] Q. Xiao, and H. Raafat, “Fingerprint image postprocessing: a combined statistical and structural approach,” *Pattern Recognition*, vol. 24, no. 10, pp. 985–992, Oct. 1991.

[5] D. C.D. Hung, "Enhancement and feature purification of fingerprint images," *Pattern Recognition*, vol. 26, no. 11, pp. 1661–1671, Nov. 1993.

[6] D. Maio, and D. Maltoni, "Neural Network Based Minutiae Filtering in Fingerprints," in *Proceedings of Int. Conference on Pattern Recognition*, pp. 1654-1658, 1998.

[7] N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, no. 11, pp. 1657–1672, Nov. 1995.

[8] A. Farina, Z. M. Kovacs-Vajna, and A. Leone, "Fingerprint minutiae extraction from skeletonized binary images," *Pattern Recognition*, vol. 32, no. 5, pp. 877–889, May 1999.

[9] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2000: Fingerprint Verification Competition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 402–412, Mar. 2002.