Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 40 (2007) 1270–1281

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction

Feng Zhao*, Xiaoou Tang

Department of Information Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong

Received 4 June 2005; received in revised form 17 June 2006; accepted 6 September 2006

Abstract

In this paper, we propose to use the fingerprint valley instead of ridge for the binarization-thinning process to extract fingerprint minutiae. We first use several preprocessing steps on the binary image in order to eliminate the spurious lakes and dots, and to reduce the spurious islands, bridges, and spurs in the skeleton image. By removing all the bug pixels introduced at the thinning stage, our algorithm can detect a maximum number of minutiae from the fingerprint skeleton using the Rutovitz Crossing Number. This allows the true minutiae preserved and false minutiae removed in later postprocessing stages. Finally, using the intrinsic duality property of fingerprint image we develop several postprocessing techniques to efficiently remove spurious minutiae. Especially, we define an H-point structure to remove several types of spurious minutiae including bridge, triangle, ladder, and wrinkle all together. Experimental results clearly demonstrate the effectiveness of the new algorithms.

Keywords: Biometrics; Bug pixel; Duality; Fingerprint minutiae extraction; H-point

1. Introduction

The recent advances of information technologies and the increasing requirements for security have led to a rapid development of automatic personal identification systems based on biometrics. Biometrics [1,2] refers to accurately identifying an individual based on his or her distinctive physiological (e.g., fingerprints, face, retina, iris) or behavioral (e.g., gait, signature) characteristics. It is inherently more reliable and more capable in distinguishing between an authorized person and a fraudulent imposter than traditional token-based or knowledge-based methods. Among all the biometrics, fingerprint recognition is one of the most reliable and promising personal identification technologies.

Fingerprints are graphical flow-like ridges and valleys present on the surface of human fingers [1]. They are widely used for personal identification [3] in many commercial, civilian, and financial applications. Various approaches to automatic minutiae extraction have been proposed in the literature. Most of the techniques [4–7] extract the minutiae from the skeleton of the fingerprint image. The skeleton is computed by thinning the binary image, which is obtained by adaptive thresholding of the input gray scale fingerprint image.

There are two types of minutiae, ridge endings and ridge bifurcations that constitute a fingerprint pattern. Ridges are generally used for minutiae extraction, since most previous researches assume that the ridges and valleys in the fingerprint have a similar width and are equally spaced. In fact, this may not always be true for various fingerprints collected by different sensors. For example, the fingerprint images we collected using an optical sensor show that the average ridge width (typically six pixels) is thicker than the average valley width (typically three to four pixels), as illustrated in Fig. 1. Since a thinner binary image is easier for skeleton computation, we propose to use the valley instead of the ridge for minutiae extraction. We use valley endings and valley bifurcations as fingerprint minutiae.

After the valley skeleton is extracted from the binary image, ideally, the width of the skeleton should be strictly one pixel. However, this is not always true, especially at the

* Corresponding author. Tel.: +852 2609 8206; fax: +852 2603 5032.
E-mail address: fzhao0@ie.cuhk.edu.hk (F. Zhao).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.09.008

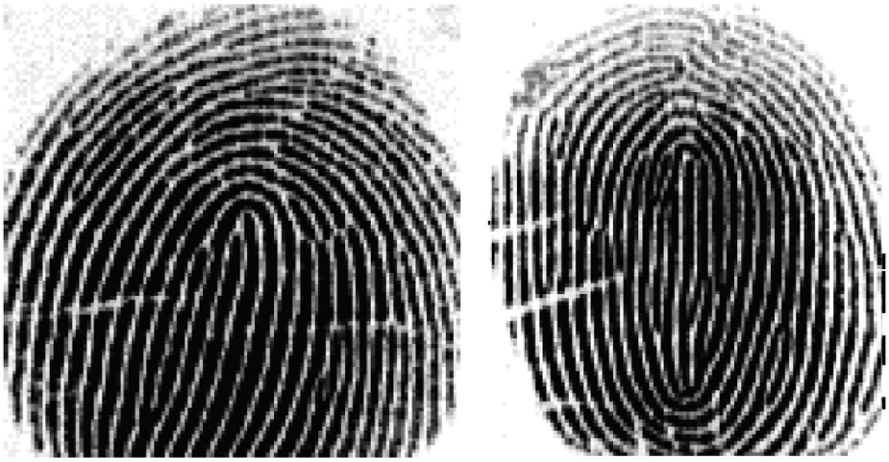

Fig. 1. Fingerprint images acquired using an optical fingerprint sensor (black areas: ridges; white areas: valleys).

intersection points, thus producing spurious minutiae points. We develop a new algorithm to remove such pixels to improve minutiae extraction.

Due to different properties of fingerprint sensors and different conditions under which a fingerprint is scanned, the quality of a fingerprint image can vary greatly. For a fingerprint image of low quality, a large number of false minutiae may be extracted. Postprocessing algorithms are generally needed to reduce the high false alarm rate.

Most of current postprocessing algorithms [8,9] eliminate the false minutiae by evaluating the statistical characteristics within an $M \times M$ matrix moving along the image pixel by pixel. Xiao and Raafat [10] develop a new postprocessing algorithm using both the statistical and structural information to eliminate the false minutiae. However, the method relies heavily on pixel connectivity computation, which is very time-consuming. A neural network-based minutiae filtering technique, which operates directly on the gray scale images is proposed by Maio and Maltoni [11]. The method relies greatly on the type and quality of training data.

Farina et al. [4] propose to clean bridge based on ridge positions instead of directional maps used by conventional methods. They argue that evaluation of the directional maps is very time-consuming. In [19], Hung presents a structural approach to connect the ridge breaks using both ridge and valley spaces. Xiao and Raafat [10] remove bridge, triangle, and ladder by calculating the number of "connected" minutiae and their structural relations. Stosz and Alyea [12] propose to eliminate wrinkle by analyzing the spatial relationship of the consecutive minutiae on the wrinkle. All these approaches rely extensively on pixel connectivity analysis one way or the other.

In this paper, we take full advantage of the duality property of fingerprint image to remove spurious minutiae. Especially, we develop an efficient bridge structure elimination method to remove several types of spurious minutiae including bridge, triangle, ladder, and wrinkle all at once.

The remainder of this paper is organized as follows. Section 2 presents the preprocessing steps and the skeleton-based minutiae extraction method. A detailed description of the proposed postprocessing algorithms is given in Section 3. Experimental results and discussions are reported in Section 4. Section 5 concludes this paper.

2. Preprocessing and minutiae extraction

A critical step in an automatic fingerprint identification system (AFIS) is reliably extracting minutiae from the input fingerprint images. The skeleton-based method generally consists of the following main steps:

(1) use an adaptive thresholding algorithm [20] to compute the binary image from the input gray scale fingerprint image;
(2) use a thinning algorithm [21,22] to compute the fingerprint skeleton from the binary image;
(3) use Rutovitz Crossing Number to extract minutiae from the skeleton of fingerprint image;
(4) postprocessing the minutiae set according to some heuristic rules.

2.1. Preprocessing

By observation of the skeleton images and their corresponding binary images, it can be seen that the misconnections and the isolated regions (hole, dot, and island) in the binary images introduce a number of spurious minutiae in the skeleton images. Fig. 2 shows an example.

In this work, we propose several preprocessing techniques before thinning of the binary image:

(1) use a morphological operator to separate some linked parallel valleys, to eliminate some spurious bridges and spurs in the skeleton image;

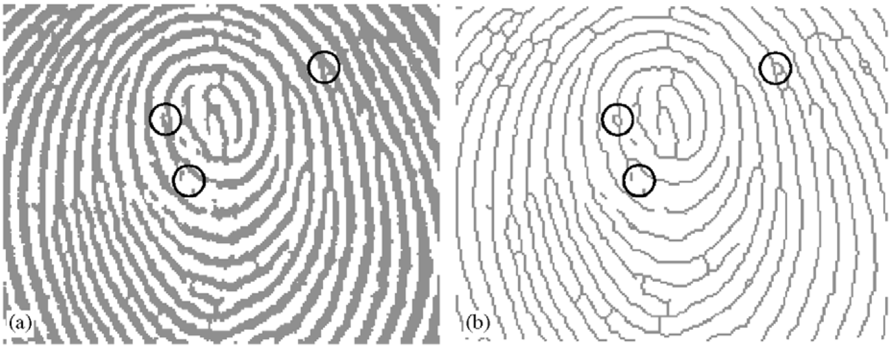

Fig. 2. An example to show the misconnections and isolated regions in (a) the binary image that introduce false minutiae in (b) the corresponding skeleton image.

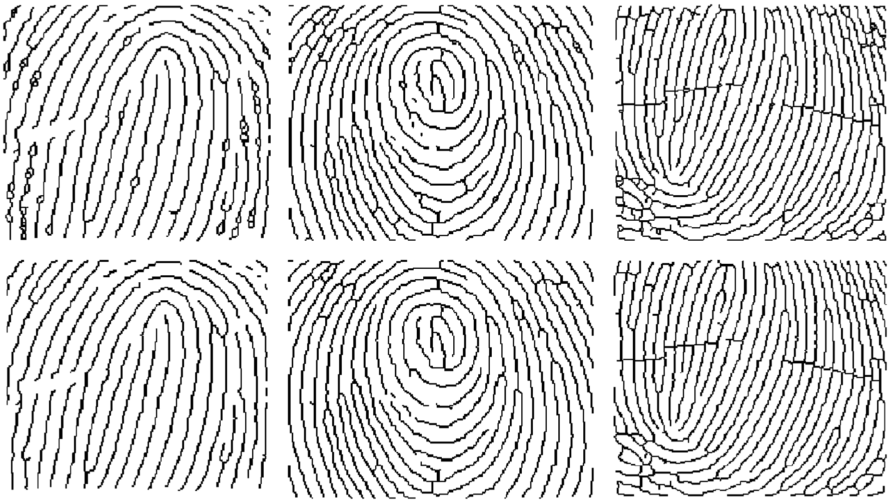

Fig. 3. Examples to show the effect of the preprocessing steps (upper row: original skeleton images; lower row: skeleton images after preprocessing).

(2) fill in the small holes with an area (number of pixels) below a threshold T_{a1}, to eliminate the spurious lakes in the skeleton image;

(3) remove the dots (isolated pixels) and the islands (short lines) with an area below a threshold T_{a2}, to eliminate the spurious lakes, dots, and some islands in the skeleton image.

The thresholds should be selected appropriately. If T_{a1} and T_{a2} are too small, the above spurious minutiae in the skeleton will not be eliminated completely. If they are too large, the skeleton will be distorted. In our experiments, we empirically set $T_{a1}=11$ and $T_{a2}=9$. Fig. 3 shows the effects of the preprocessing steps.

2.2. Minutiae extraction

The concept of Crossing Number (*CN*) is widely used for extracting the minutiae [4–7]. Rutovitz's definition [13] of crossing number for a pixel *P* is

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

$$CN = \frac{1}{2}\sum_{i=1}^{8} |P_i - P_{i+1}| ,$$

where P_i is the binary pixel value in the neighborhood of *P* with $P_i = (0 \text{ or } 1)$ and $P_1 = P_9$.

The skeleton image of fingerprint is scanned and all the minutiae are detected using the properties of *CN*, as illustrated in Fig. 4.

Ideally, the width of the skeleton should be strictly one pixel. However, this is not always true. Fig. 5 shows some examples, where the skeleton has a two-pixel width at some bug pixel locations.

We define a bug pixel as the one with more than two 4-connected neighbors (marked by bold italic ***1*** and ***0***). These bug pixels exist in the fork regions where bifurcations should be detected, but they have $CN=2$ instead of $CN>2$. The

CN	*Property*
0	Isolated point
1	Ending point
2	Connective point
3	Bifurcation point
4	Crossing point

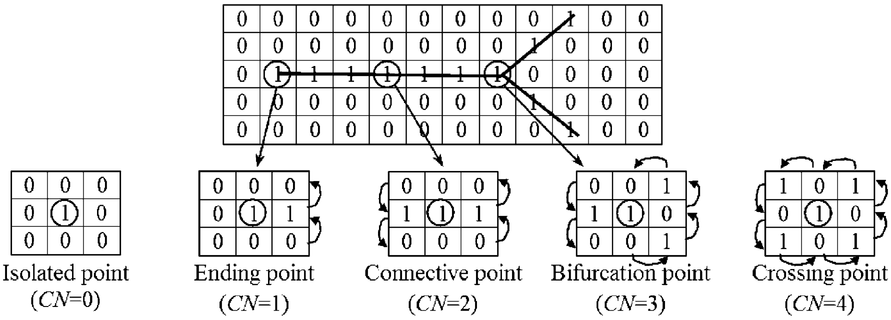

Fig. 4. Illustration of Crossing Number properties (“1”: black pixels in the skeleton image).

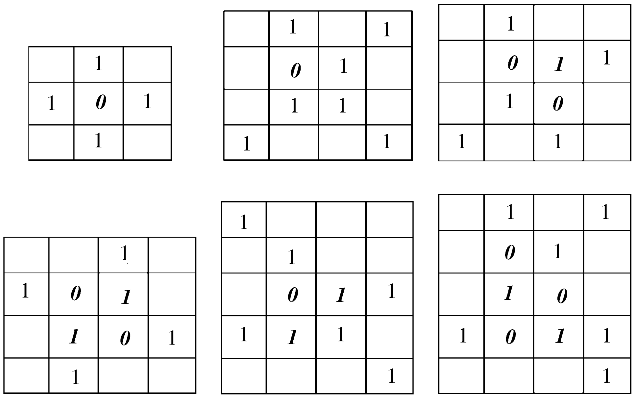

Fig. 5. Examples of bug pixels and their validation (bold-italic ***0***: deleted bug pixels; bold-italic ***1***: preserved bug pixels that are changed to normal pixels).

existence of bug pixels may (i) destroy the integrity of spurious bridges and spurs, (ii) exchange the type of minutiae points, and (iii) miss detecting true bifurcations, as illustrated in Fig. 6. Therefore, before minutiae extraction, we develop a validation algorithm to eliminate the bug pixels while preserving the skeleton connectivity at the fork regions. By scanning the skeleton of fingerprint image row by row from top–left to bottom–right, we delete the first bug pixel encountered and then check the next bug pixel again for the number of 4-connected neighbors. If the number of 4-connected neighbors after the deletion of the previous bug pixel is still larger than two, it will also be deleted; otherwise it will be preserved and treated as a normal pixel. Some examples are shown in Fig. 5. After this validation process, all the pixels in the skeleton satisfy the *CN* properties. Thus we can extract all the minutiae including true minutiae and false minutiae. The false minutiae can be eliminated in the postprocessing stage.

3. Postprocessing algorithms

After preprocessing on the binary and skeleton images, we extract all the minutiae from the fingerprint skeleton using the Rutovitz Crossing Number. However, due to various noises in the fingerprint image, the extraction algorithm produces a large number of spurious minutiae such as break, spur, bridge, merge, triangle, ladder, lake, island, and wrinkle, as shown in Fig. 7. Therefore, reliably differentiating spurious minutiae from genuine minutiae in the postprocessing stage is crucial for accurate fingerprint recognition. The more spurious minutiae are eliminated, the better the matching performance will be. In addition, matching time will be significantly reduced because of the reduced minutiae number. This is very important since the execution time is a critical parameter in an AFIS.

For the various types of false minutiae illustrated in Fig. 7, we use preprocessing algorithms described in Section 2 to eliminate dots, lakes, and a number of islands and spurs. For the rest of the false minutiae types, we observe that there is at least one bridge structure in such spurious minutiae as bridge, triangle, ladder and wrinkle. For the various types of breaks, there are also corresponding bridge structures in the duality image. We define a bridge structure and its corresponding dual break in the duality image collectively as an *H*-point. If we can successfully remove the *H*-points in the image, we can eliminate most of the spurious minutiae.

To further explain the *H*-point definition, we need to first understand the duality definition. In a fingerprint image, for each ridge ending, there is generally a corresponding valley bifurcation and vice versa [2], with the only exception at the singularity points (cores and deltas) [14]. This is called the termination/bifurcation duality, as illustrated in Fig. 8(a). Around a bridge structure, such a duality takes on the form of a bridge in the ridge (or valley) skeleton image and its corresponding dual break in the valley (or ridge) skeleton

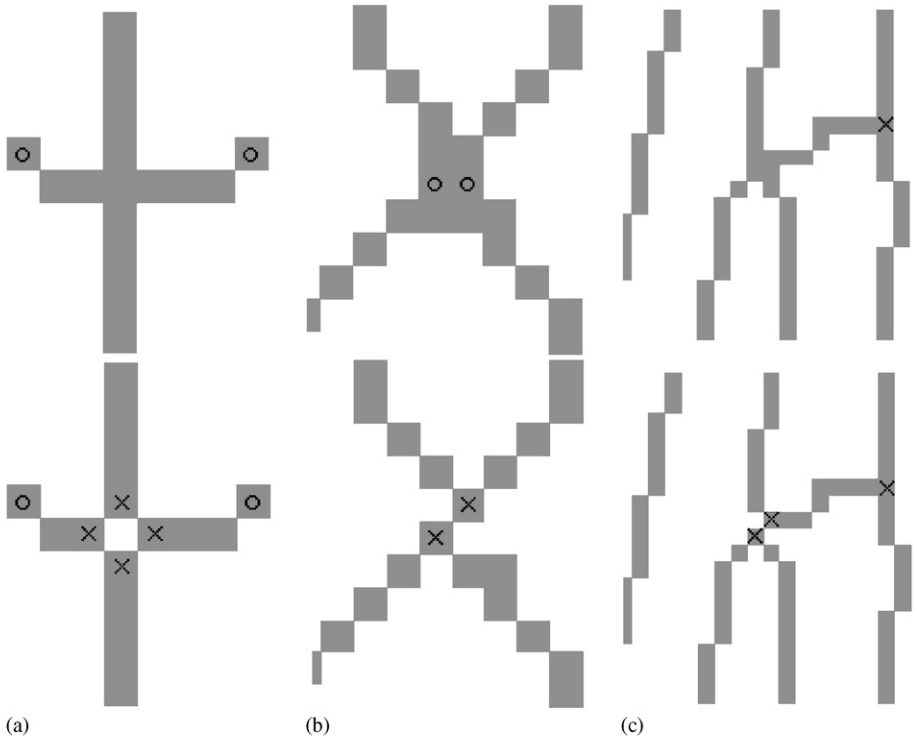

Fig. 6. Without validating the bug pixels, we may have: (a) four bifurcations (“x”) are missed; (b) two bifurcations are misdetected as two endings (“o”); (c) two bifurcations are missed including one true bifurcation.

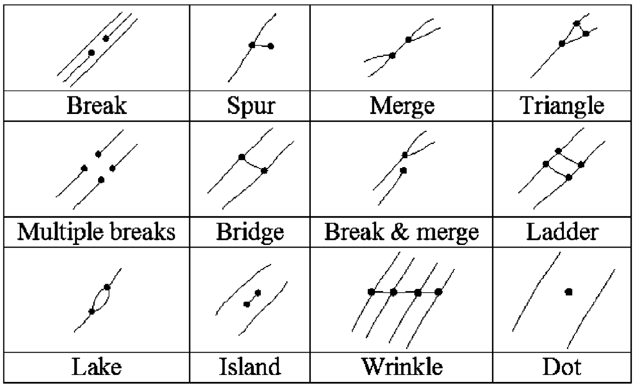

Fig. 7. Examples of false minutiae (black dots).

image, as shown in Fig. 8(b). We define such a structure with two bifurcations and two corresponding endpoints (endings) as an *H*-point. In this work, we design a simple procedure utilizing the duality property to eliminate the *H*-point.

Since any particular processing step will affect the performance of later steps, we have to be very careful of the processing order. We design several algorithms to remove spurious minutiae in the following order, as shown in Fig. 9.

In the first stage, we remove some short breaks based on the conventional definition of a break. If the endpoints of a break satisfy all the following conditions, they will be removed:

(a) the distance between two endpoints is below a threshold T_1;
(b) the difference between the orientation angles of two endpoints (Ang_1, Ang_2) is within an interval of [θ_1, θ_2];
(c) the difference between the orientation angle of the line connecting the two endpoints (Ang_3) and either angle of Ang_1 or Ang_2 is within an interval of [θ_3, θ_4].

In order to calculate the orientation angle of an endpoint, we look for the 8-connected neighbors around the endpoint. During the tracing procedure, our algorithm keeps going forward even after encountering a bifurcation point so that the orientation of endpoint 2 is estimated as the “dashed” line instead of the “solid” line, as illustrated in Fig. 10. Thus the endpoints’ orientations satisfy the above orientation angle rules (b) and (c). Otherwise, endpoints 1 and 2 cannot be removed. If that is the case, the bifurcation and endpoint 2 will be removed in the following spur elimination stage. As a result, the true bifurcation will be detected as an endpoint which results in the type-exchanged error. Therefore, this strategy mainly helps to decrease the type-exchanged error due to the poor valley connectivity.

In the second stage, we first label the connected pixels in the skeleton image. If the distance (d) between a bifurcation point and an endpoint is below the threshold T_2 and their

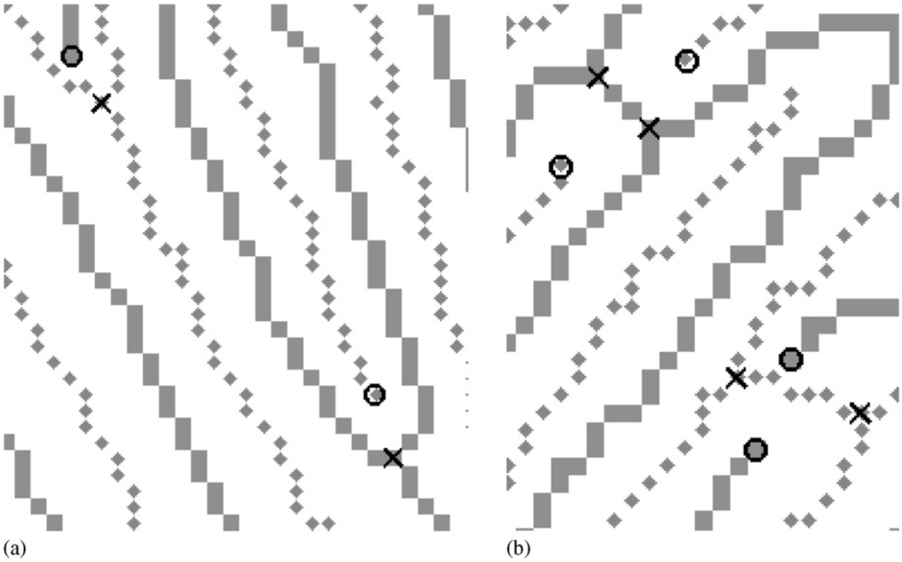

Fig. 8. Illustration of ridge and valley duality. The solid gray squares represent the valley skeleton, and the solid gray diamonds represent the ridge skeleton. (a) The termination/bifurcation duality; (b) H-point examples: a valley bridge corresponds to a ridge break in the dual skeleton image and vice versa ("o": endings; "x": bifurcations).

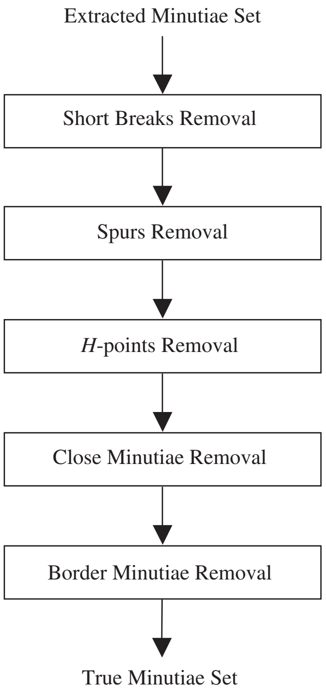

Fig. 9. Elimination process of false minutiae.

labels are the same, we again label the connected pixels within a small window $(2d+1\times 2d+1)$ centered around the endpoint or the bifurcation point. If their labels are still the same, we remove both of them; otherwise they are preserved. Fig. 11 demonstrates the procedure.

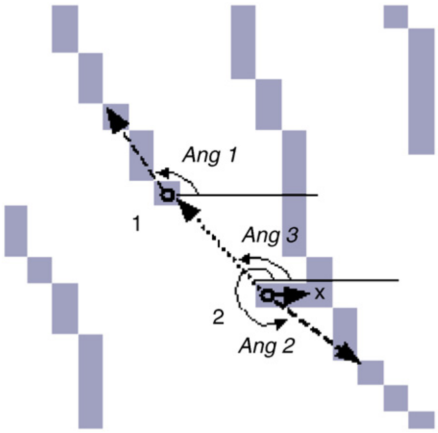

Fig. 10. Keep tracing when a bifurcation point is met ("o": endings; "x": bifurcation).

In the third stage, the H-points are detected and eliminated. If a bridge in the ridge (or valley) skeleton image and a break in the valley (or ridge) skeleton image satisfy the following conditions, they form an H-point (see Fig. 12):

(1) the intersecting point lies between the two endpoints and the two bifurcation points;
(2) the distance between the bridge midpoint M_2 and the break midpoint M_1 is within a threshold T_3;
(3) the intersecting angle θ between the bridge and the break is within an interval of $[\theta_5, \theta_6]$.

Finally, we eliminate those minutiae that are too close to each other and all the minutiae within a certain distance threshold T_4 from the image border. After postprocessing, a large percentage of the spurious minutiae are eliminated,

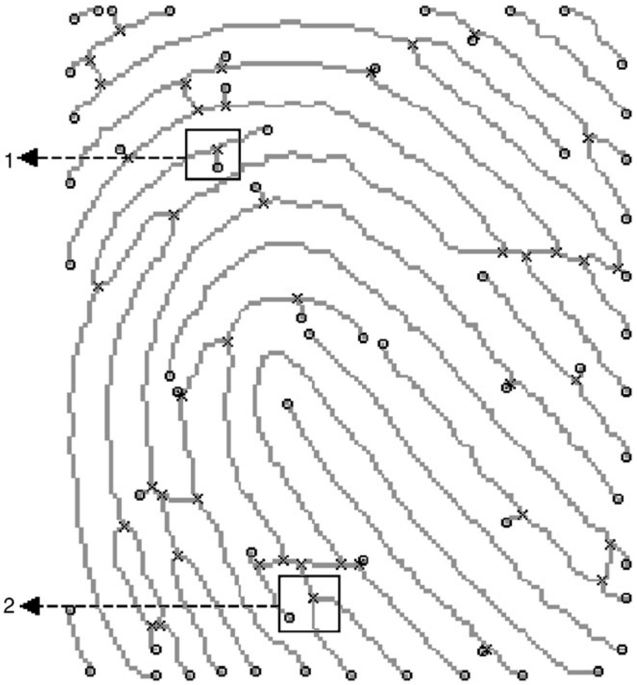

Fig. 11. Examples showing the elimination of spurs. After relabeling the connected pixels within the square, the bifurcation and the endpoint inside square 1 still have the same labels, whereas those within square 2 have different labels ("o": endings; "x": bifurcations).

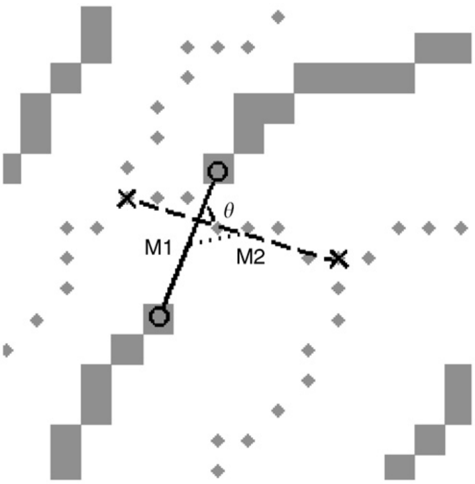

Fig. 12. An H-point example ("o": endings; "x": bifurcations).

the remaining minutiae are treated as true minutiae which are used for later fingerprint matching.

4. Experiments and discussions

To evaluate the performance of the proposed algorithms, we perform a series of experiments on a portion of a live-scan fingerprint database captured with an optical sensor (*StarTek FM100*). It contains a total of 6780 fingerprint images from 113 subjects. Each person was asked to provide fingerprint images for all the 10 fingers. In the first stage, three fingerprint images per finger were acquired. Three more fingerprint images per finger were obtained a month later. The subjects mainly consist of undergraduate and postgraduate students and their friends. The profile of the database is shown in Table 1. The captured fingerprint images vary in quality. In the experiments, we select one fingerprint image of medium quality from the six impressions of each finger. These fingerprint images (256×256) are then cropped into 170×180 in size in order to remove the very noisy border area.

Table 1
Live-scan fingerprint database

Subjects (%)	Students	90.3
	Others	9.7
Gender (%)	Female	38
	Male	62
Age (%)	Under 25	80
	25–50	20
Time interval	One month between two capture stages	
Environment	Normal office conditions (indoor)	

The valley skeleton and ridge skeleton are first obtained from the valley image and its dual ridge image, respectively. The valley skeleton agrees rather well with the original valley image, while the ridge skeleton introduces a large number of spurious lakes and bridges. Consequently, the ridge skeleton will produce more spurious minutiae. Fig. 13 shows a typical example.

The accuracy rates of applying the minutiae extraction algorithm on ridge skeleton and valley skeleton before and after preprocessing are reported in Tables 2 and 3, respectively. In the tables, the accuracy rates of ending and bifurcation are computed by E_t/E_e and B_t/B_e, respectively. The total rate is calculated using the following formula:

$$\textit{Total rate} = \frac{E_t + B_t}{E_e + B_e}, \tag{1}$$

where E_t and B_t are the number of true endings and true bifurcations in the extracted endings (E_e) and bifurcations (B_e), respectively.

From the results, we can see that after preprocessing the accuracy rate of bifurcation is improved significantly, especially for the ridge bifurcation. It demonstrates that the preprocessing algorithm does eliminate a large number of spurious lakes, bridges, and spurs, which introduce false bifurcations. However, the accuracy rate of endings is only increased slightly since the preprocessing algorithm only eliminates some spurious islands that introduce false endings. In fact, the spurious dots also introduce false endings and are eliminated efficiently in the preprocessing stage. However, there are only a small number of dots in the skeleton image. The improvement of the accuracy rate of ridge bifurcation is greater than that of valley bifurcation.

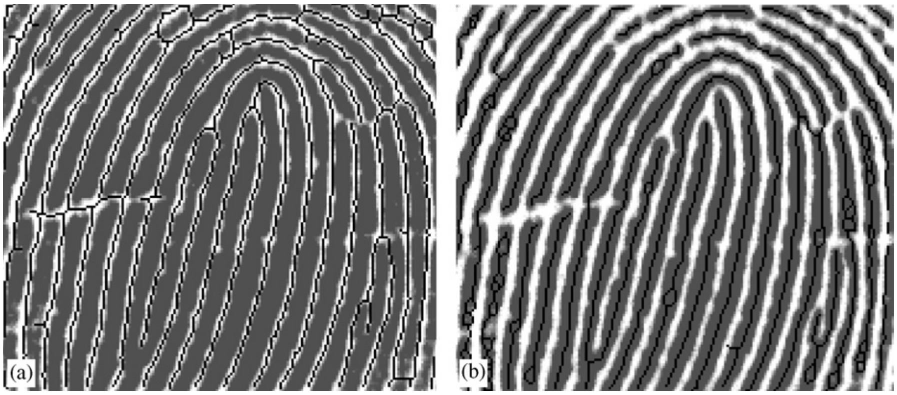

Fig. 13. (a) Valley skeleton; (b) ridge skeleton (the skeleton is overlaid on the original gray scale fingerprint image).

Table 2
Accuracy rates for ridge minutiae

	Before preprocessing (%)	After preprocessing (%)
Ending	10.6	10.7
Bifurcation	19.3	50.8
Total rate	13.4	16.8

Table 3
Accuracy rates for valley minutiae

	Before preprocessing (%)	After preprocessing (%)
Ending	12.2	12.6
Bifurcation	16.2	26.3
Total rate	13.2	16.2

Table 4
Number of minutiae before and after validating bug pixels

After preprocessing		After validating bug pixels	
Endings	Bifurcations	Endings	Bifurcations
67	47	67	47
87	27	87	27
55	123	55	125
62	110	62	118
77	55	75	57
106	20	93	27

This shows that the ridge skeleton introduces more spurious minutiae. In addition, the computation speed for valley thinning is much faster than ridge thinning.

Table 4 shows some typical results of validating the bug pixels. From the results, we can see that the bug pixels exist in the fork region where bifurcations should be extracted. Some fingerprint skeletons may have more bug pixels and some may have none.

We have compared the performance of our minutiae extraction algorithm with the other methods proposed by Maio [15], Cheng [16], and Kim [17]. Before showing the experimental results, we first describe the following terms used for performance evaluation:

Table 5
Average error rates after postprocessing

Methods	False(%)	Dropped(%)	Type-exchanged(%)	Total error(%)
Proposed	15.3	6.9	5.3	27.5
Maio [15]	11.8	6.5	13.1	31.4
Cheng [16]	9.6	15.9	10.4	35.9
Kim [17]	25.8	13.8	6.3	45.9

True minutiae (M_T): Minutiae marked by a human expert.
Extracted minutiae (M_E): Minutiae remained after postprocessing.
False (non-existent) minutiae (M_F): Minutiae extracted by the algorithm that do not coincide with M_T.
Dropped (missing) minutiae (M_D): Minutiae marked by a human expert that are not extracted by the algorithm.
Type-exchanged minutiae (M_{TE}): Minutiae extracted by the algorithm that coincide with M_T except the minutiae type.

Table 5 reports the results in terms of average error rates of false (M_F/M_E), dropped (M_D/M_T), and type-exchanged (M_{TE}/M_E) minutiae. The total error rate is the sum of them. From the results, we can see that our algorithm is better than the other methods in terms of type-exchanged and total error rates. Compared with the best result by Cheng's method [16], our algorithm produces a slightly higher error rate of false minutiae. For the dropped error rate of our algorithm, it is comparable to the best result by Maio's method [15]. In addition, since the total error rate in the preprocessing stage is about 83.8%, we can conclude that the average error rate drops 56.3% after postprocessing.

To further quantitatively evaluate the performance of our minutiae extraction algorithm we adopt the "goodness index" (*GI*) measurement [7], which compares the extracted

Table 6
GI values for a dataset of 10 fingerprint images

Fingerprint	P	D	I	T	GI
01	11	1	1	12	0.75
02	18	4	1	19	0.68
03	21	3	3	28	0.54
04	13	2	2	17	0.53
05	15	3	3	18	0.50
06	19	3	4	24	0.50
07	15	7	1	16	0.44
08	12	5	1	14	0.43
09	12	2	3	17	0.41
10	8	3	3	11	0.18

Empirically determined parameter values: T_1, T_2, T_3, T_4, θ_1, θ_2, θ_3, θ_4, θ_5, θ_6 are 9, 6, 2.5, 7 pixels, 145°, 225°, −25°, 25°, 65°, and 115°.

minutiae with the true minutiae obtained from the same fingerprint by a human expert (ground truth). The goodness index is defined as follows:

$$GI = \frac{P - D - I}{T}, \tag{2}$$

where P is the total number of paired minutiae, D is the number of deleted spurious minutiae (false and type-exchanged), I is the number of inserted missing minutiae (dropped) and T is the number of true minutiae. An extracted minutia m_1 is said to be paired with the true minutia m_2 marked by the human expert if m_1 lies within an 8x8 tolerance box centered around m_2. The maximum value of GI is 1, which means that all the extracted minutiae are paired with true

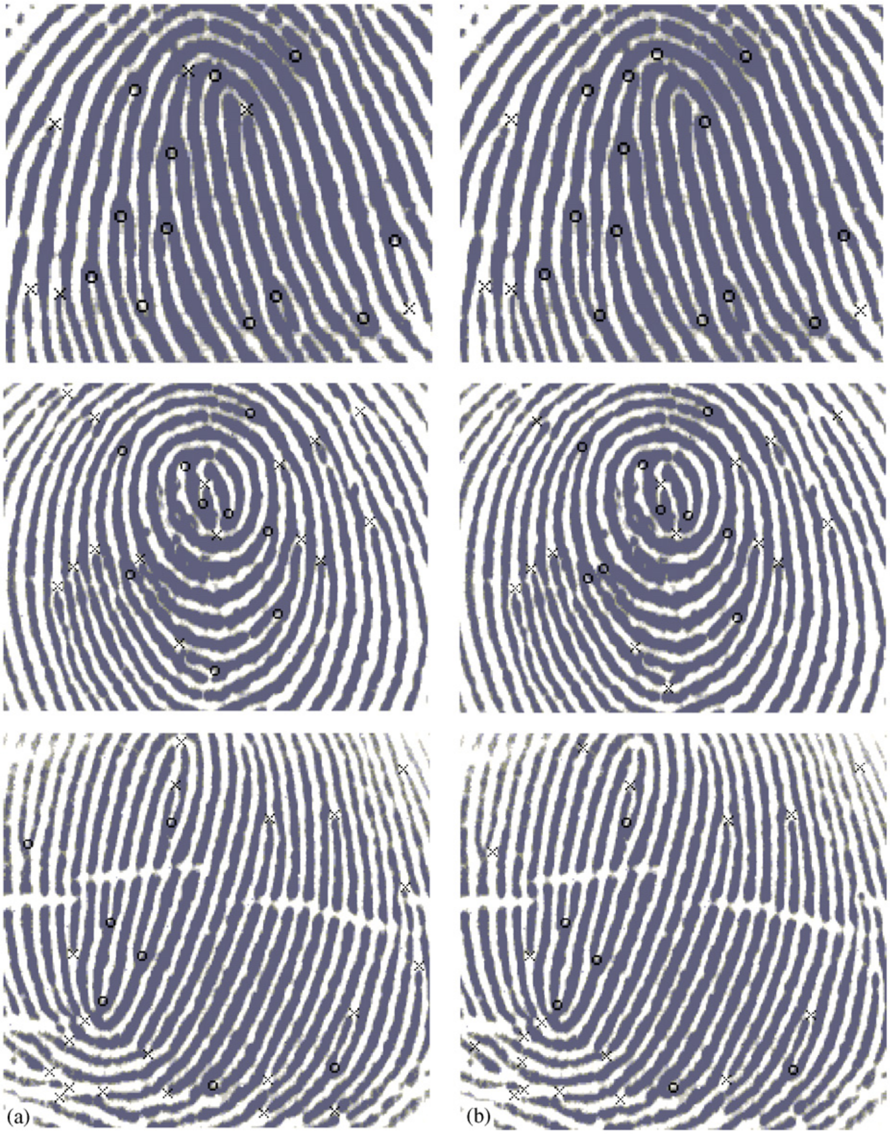

Fig. 14. Minutiae extraction example results: (a) minutiae marked by a human expert; (b) automatically extracted minutiae;

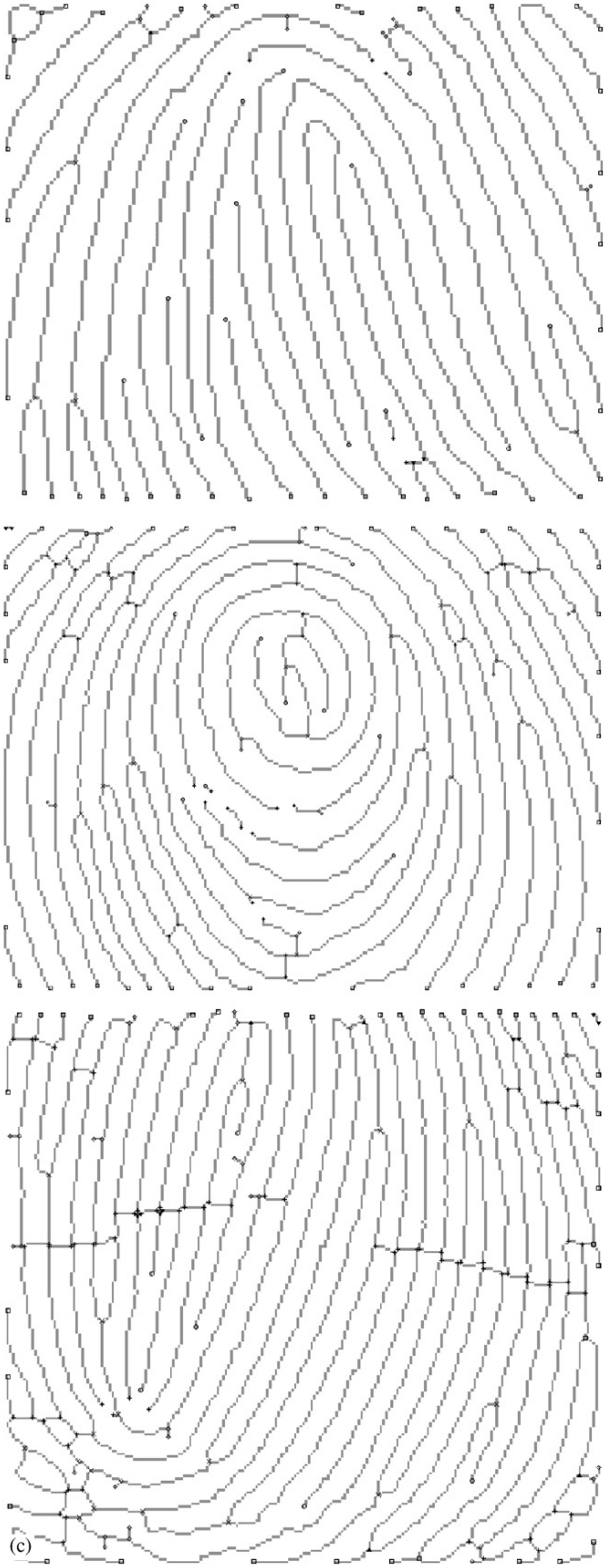

Fig. 14. (c) postprocessing results at different processing stages (diamond (◇): eliminated spurs; plus (+): eliminated H-points; solid down triangle (▼): eliminated close minutiae; square (□): eliminated border minutiae; circle (○): survived endings; X-mark (x): survived bifurcations).

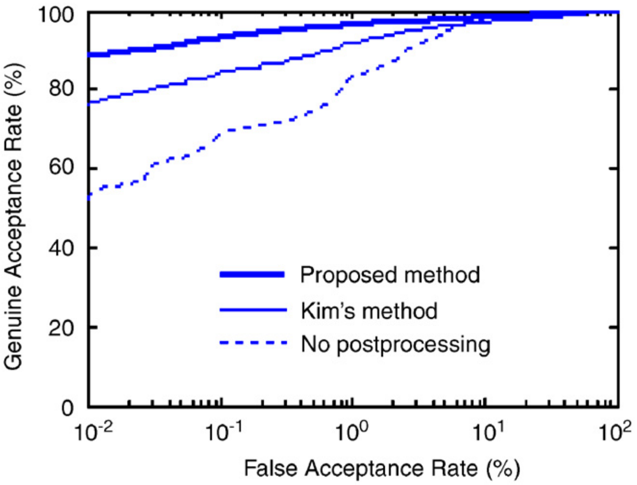

Fig. 15. The ROC curves of the original matching system integrating the proposed method, Kim's method, and without adopting postprocessing.

minutiae and no spurious minutiae and missing minutiae are detected ($P = T$, $D = I = 0$). A high value of *GI* indicates the high quality of the extracted minutiae. Table 6 presents the *GI* values for a representative subset of 10 fingerprint images. The maximum and minimum values of *GI* for this dataset are 0.75 and 0.18, respectively. The average value is 0.50. The results outperform those presented in Ref. [7] and are comparable to those reported in Ref. [18].

Fig. 14 shows several typical results of our processing algorithms. From the results, we can observe that

(1) the survived minutiae after the postprocessing agree rather well with the minutiae marked by the human expert in regions where the valley structures are clear;
(2) false minutiae and dropped minutiae typically occur in the noisy regions near the border or creases;
(3) type-exchanged minutiae occur in the areas where the valley connectivity is poor.

Finally, we test our algorithms on a standard database FVC2002 DB1, which contains a set of 880 fingerprint images (388 × 374) from 110 individuals. The overall matching performance is measured by the receiver operating characteristic (ROC) curve, which plots the genuine acceptance rate (GAR) against the false acceptance rate (FAR) at different operating points (matching score thresholds). As illustrated in Fig. 15, the overall performance of the matching system is greatly improved by integrating the postprocessing algorithms. In addition, the proposed approach outperforms Kim's method [17] over a wide range of FAR values.

5. Conclusions

In this paper, we develop several efficient preprocessing techniques to enhance the skeleton image for minutiae extraction from the valley instead of the ridge of the fingerprint. Our minutiae extraction algorithm can detect all the minutiae, including both true and false minutiae, using the Rutovitz Crossing Number (*CN*) on the skeleton images after validating all the bug pixels introduced at the thinning stage. This allows the true minutiae preserved and false minutiae removed in later postprocessing stages. Taking full advantage of the duality property of the fingerprint image, we design several postprocessing techniques to efficiently remove spurious minutiae. Especially, we develop an efficient *H*-point elimination method to remove several types of spurious minutiae including bridge, triangle, ladder, and wrinkle all at once. The high values of goodness index (*GI*) have illustrated the encouraging performance of our proposed method.

Acknowledgments

The work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region. The work was done while all the authors were with the Chinese University of Hong Kong.

References

[1] A.K. Jain, R. Bolle, S. Pankanti (Eds.), Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, Boston, 1999.
[2] L.C. Jain, U. Halici, I. Hayashi, S.B. Lee, S. Tsutsui (Eds.), Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, Boca Raton, 1999.
[3] H.C. Lee, R.E. Gaensslen (Eds.), Advances in Fingerprint Technology, CRC Press, Boca Raton, 1994.
[4] A. Farina, Z.M. Kovács-Vajna, A. Leone, Fingerprint minutiae extraction from skeletonized binary images, Pattern Recognition 32 (5) (1999) 877–889.
[5] A.K. Jain, L. Hong, R. Bolle, On-line fingerprint verification, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 302–314.
[6] B.M. Mehtre, Fingerprint image analysis for automatic identification, Mach. Vision Appl. 6 (1993) 124–139.
[7] N.K. Ratha, S. Chen, A.K. Jain, Adaptive flow orientation-based feature extraction in fingerprint images, Pattern Recognition 28 (11) (1995) 1657–1672.
[8] B. Moayer, K.S. Fu, A tree system approach for fingerprint recognition, IEEE Trans. Pattern Anal. Mach. Intell. 8 (3) (1986) 376–387.
[9] L. O'Gorman, J.V. Nickerson, An approach to fingerprint filter design, Pattern Recognition 22 (1) (1989) 29–38.
[10] Q. Xiao, H. Raafat, Fingerprint image postprocessing: a combined statistical and structural approach, Pattern Recognition 24 (10) (1991) 985–992.
[11] D. Maio, D. Maltoni, Neural network based minutiae filtering in fingerprints, in: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, 1998, pp. 1654–1658.
[12] J.D. Stosz, L.A. Alyea, Automatic system for fingerprint authentication using pores and ridge structure, in: Proceedings of SPIE Automatic Systems for the Identification and Inspection of Humans, vol. 2277, 1994, pp. 210–223.
[13] D. Rutovitz, Pattern recognition, J. Roy. Stat. Soc. 129 (1966) 504–530.

[14] K. Uchida, T. Kamei, M. Mizoguchi, T. Temma, Fingerprint card classification with statistical feature integration, in: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, 1998, pp. 1833–1839.

[15] D. Maio, D. Maltoni, Direct gray-scale minutiae detection in fingerprints, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 27–40.

[16] J. Cheng, J. Tian, Fingerprint enhancement with dyadic scale-space, Pattern Recognition Lett. 25 (11) (2004) 1273–1284.

[17] S. Kim, D. Lee, J. Kim, Algorithm for detection and elimination of false minutiae in fingerprint images, in: Proceedings of the Third International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'01), Halmstad, Sweden, 2001, pp. 235–240.

[18] D. Simon-Zorita, J. Ortega-Garcia, S. Cruz-Llanas, J. Gonzalez-Rodriguez, Minutiae extraction scheme for fingerprint recognition systems, in: Proceedings of the International Conference on Image Processing, vol. 3, 2001, pp. 254–257.

[19] D.C.D. Hung, Enhancement and feature purification of fingerprint images, Pattern Recognition 26 (11) (1993) 1661–1671.

[20] J.R. Parker, Gray level thresholding in badly illuminated images, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 813–819.

[21] E.S. Deutsch, Thinning algorithms on rectangular, hexagonal, and triangular arrays, Commun. ACM 15 (9) (1972) 827–837.

[22] L. Lam, S.W. Lee, C.Y. Suen, Thinning methodologies—a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 14 (9) (1992) 869–885.

About the Author—FENG ZHAO received his B.E. degree from the University of Science and Technology of China, Hefei, in 2000, and his M.Phil. degree from the Chinese University of Hong Kong, in 2002. He is currently a Ph.D. student in the Department of Information Engineering, the Chinese University of Hong Kong. His research interests include fingerprint recognition, image processing, and pattern recognition.

About the Author—XIAOOU TANG received his B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and his M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received his Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong. He is also the group manager of the Visual Computing Group at the Microsoft Research Asia. He is a local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an area chair of ICCV'07, a program chair of ICCV'09, and a general chair of the ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a guest editor of the Special Issue on Underwater Image and Video Processing for the IEEE Journal of Oceanic Engineering and the Special Issue on Image- and Video-based Biometrics for IEEE Transactions on Circuits and Systems for Video Technology. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). His research interests include computer vision, pattern recognition, and video processing.