PERGAMON

Pattern Recognition 34 (2001) 999–1013

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge

Xudong Jiang*, Wei-Yun Yau, Wee Ser

Centre for Signal Processing, EEE, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

Received 2 August 1999; received in revised form 1 March 2000; accepted 1 March 2000

Abstract

This paper presents a minutiae detection procedure based on adaptive tracing the gray-level ridge of the fingerprint image with piecewise linear lines of different length. The original fingerprint image is smoothed with an adaptive-oriented smoothing filter only at some selected points. This will greatly reduce the computational time. Each ridge in the skeleton is labeled with a number so that each detected minutia is associated with one or two ridge numbers, which is useful for post processing. We objectively assess the performance of this approach by using two large fingerprint databases.

Keywords: Fingerprint image filtering; Adaptive tracing ridges; Minutiae detection; Post-processing

1. Introduction

Highly reliable automatic personal authentication is fast becoming important in our electronically interconnected society, especially for electronic commerce. Biometrics is seen as an important research area that can satisfy the high security requirement and yet easy to use. The most widely used biometrics is fingerprint because of its uniqueness and immutability. In order to compare two fingerprints, a set of invariant and discriminating features have to be extracted from them. It is well known for fingerprint research that minutiae are discriminating and reliable features. Most fingerprint verification systems required to provide a high degree of security are so far based on minutiae matching. Unfortunately, noise, contrast deficiency, improper image acquisition, geometrical transformation, deformation and skin elasticity often make reliable minutiae detection very difficult. Spurious minutiae can be produced and valid minutiae can be hidden due to the low image quality of the fingerprint. Therefore, minutiae detection is a very crucial process in fingerprint matching.

Various approaches for minutiae detection have been investigated. Most of these approaches consist of a series of processing operation: preprocessing, binarization, thinning, minutiae extraction and post-processing. A brief survey of the published literature in minutiae detection can be found in Refs. [1–3]. In the preprocessing, various approaches have been suggested, such as using bandpass filter [4], contextual/directional filter [3,5], directional Fourier transform filter [6], Gabor filter [7] and fuzzy approach [8] to enhance or smooth each pixel of the raw gray-level fingerprint image. To binarize the fingerprint image, several thresholding methods such as the local [4,9], dynamic [10] and adaptive [8] thresholding have been proposed. A number of thinning algorithms [11–15] have been investigated to obtain the skeleton of the ridge. A few post processing approaches can be found in Refs. [3,5,10,16–18]. In addition, neural network based approaches were introduced in Refs. [19,20].

Maio and Maltoni proposed a rather different approach to minutiae detection [2]. They proposed an approach that follows the ridge lines on the gray-scale image and detects the minutiae directly from the gray-scale image during the ridge following. Their results

* Corresponding author. Tel.: + 65-790-5943; fax: + 65-791-0128.

E-mail address: exdjiang@ntu.edu.sg (X. Jiang).

0031-3203/01/$20.00
PII: S0031-3203(00)00050-9

showed the superiority of this technique in terms of efficiency and robustness [2] as compared to the conventional thresholding and thinning approaches. In our investigation of this technique we found that the efficiency and robustness can be further improved.

In this work, we present an improved minutiae detection approach which uses the basic idea in Ref. [2] but different from it in many details. Our approach adaptively traces the gray-level ridge of the original fingerprint image and applies adaptive oriented filters to the image only at some regions that need to be smoothed. One of the noticeable advantages of ridge detection by tracing is that it approximates the ridge with piecewise linear lines. Ridge detection is therefore not pixelwise but stepwise. Because of the large variance of the bending level of the ridge at different points, the tracing step should be adaptive to the bending level of the ridge. This will speed up the tracing and at the same time maintaining the tracing precision. Another major contribution of our algorithm is the post-processing. Unlike most approaches, our approach is based not only on the location relationship of the minutiae, but also the associate ridge relationship and the certainty level of the minutiae. This is very important for noisy fingerprint image where a large number of spurious minutiae will be initially detected. Reliably differentiating spurious minutiae from true minutiae is therefore crucial for accurate fingerprint recognition. In addition, some fingerprint image smoothing and enhancement methods are discussed because they are essential and important for the minutiae detection to be robust with respect to the quality of fingerprint images.

In the following sections, we will describe in detail our minutiae detection approach. Section 2 discusses our proposed adaptive ridge tracing algorithm. Section 3 describes the minutiae detection method based on this adaptive ridge tracing algorithm. In Section 4 we give an approach of post-processing to remove spurious minutiae. Section 5 describes the implementation of the proposed minutiae detection algorithm and the performance of the proposed approach in a verification test. Section 6 then concludes the paper.

2. Adaptive tracing the ridge of fingerprint

A gray-level fingerprint image $\mathbf{I}$ of size $H \times W$ can be considered as an oriented texture pattern that contains ridges separated by valleys. Let $g(i, j)$ be the gray value of a fingerprint image at pixel (i, j) and $\varphi(i, j)$ be the orientation at this pixel with $0 \leqslant g(i, j) \leqslant 255$, $-\pi/2 < \varphi(i, j) \leqslant \pi/2$ and $(i, j) \in I$. The orientation $\varphi(i, j)$ represents the ridge line local orientation at pixel (i, j) and can be computed using one of the several methods proposed in the literature [21–24]. We incorporate the method in Ref. [23], which uses gradient and least-square methods to estimate the orientation. At the same time, a certainty level $c(i, j)$ of the orientation $\varphi(i, j)$ is calculated by using the method introduced by Jain et al. [25].

A ridge can be defined as a set of points along the local orientation which are local maxima of $g(i, j)$ relative to sections orthogonal to the local orientation $\varphi(i, j)$. Detecting the skeleton (one pixel width ridge) is therefore locating these local maxima.

2.1. Locating the section maximum

Given a pixel (i^k, j^k), $(i^k, j^k) \in \mathbf{I}$, a section set Ω^k orthogonal to the local orientation $\varphi(i^k, j^k)$ can be defined as follows:

$$\Omega^k = \{(i, j) | (i, j) \in \mathbf{I}, i = i^k + round((j - j^k) \tan(\varphi_s^k)), j \in \Omega_j\}, \tag{1}$$

$$\Omega_j = [\, j^k - round(\sigma \cos(\varphi_s^k)), j^k + round(\sigma \cos(\varphi_s^k))], \tag{2}$$

$$\varphi_s^k = \begin{cases} \varphi(i^k, j^k) - \pi/2 & \text{if } \varphi(i^k, j^k) > 0, \\ \varphi(i^k, j^k) + \pi/2 & \text{otherwise,} \end{cases} \tag{3}$$

where $round(\cdot)$ rounds its argument to the nearest integer and σ is a predetermined constant. With Eq. (3), we keep $-\pi/2 < \varphi_s^k \leqslant \pi/2$.

Determining a local maximum of the section set Ω^k is important for accurate minutiae detection as it will affect the skeleton of the ridge produced in the tracing process. This skeleton image will then be used to detect and locate the minutiae of the fingerprint. Noise and contrast deficiency may cause breaks in the ridges, bridges between ridges, and overall gray-level intensity variation. This will make a simple maximum not be located at the ridge center or worse, be located in another ridge. A large number of spurious minutiae will therefore be produced if proper care is not taken to minimize the location error of the local maximum of the section set. To smooth and enhance the fingerprint image, various filtering techniques were proposed in the literature.

A band-pass filter along the orientation orthogonal to the ridge orientation can be used to separate cluttered or linked parallel ridges. O'Gorman and Nickerson [5] and Mehtre [3] performed image enhancement by using band pass filters. Jain et al. [25,26] detected the ridge with two masks, which in principle perform some kinds of band pass filtering. Hong et al. [7] introduced a fingerprint image enhancement method that employed a Gabor filter together with ridge frequency estimation. They reported a good performance enhancement. Gabor filter actually also performs a band pass filtering along the orientation orthogonal to the ridge orientation. The oriented band-pass filter undoubtedly increases the clarity of ridge structure or the contrast between the ridges and valleys and therefore gives a good visual appearance of the image. Moreover, it is also able to separate some linked

ridges in cases where the ridge frequency is similar to the value estimated during the design of the band-pass filter. However, a fingerprint enhancement algorithm based on the orientated band-pass filter may result in spurious ridge structures. Although this can be avoided by using the correct ridge frequency information, the exact estimation of the ridge frequency is a difficult task, especially in the noisy regions and in the regions in which the minutiae exist. The performance of the band-pass filter strongly relies on the accurate estimation of the ridge frequency. In our experiments, the ridge frequency estimation error may cause ridge location error, introduce false ridge structure and consequently result in a large number of spurious minutiae.

The employment of the band-pass filter is aimed at separating the cluttered or linked parallel ridges that will produce spurious minutiae. Such spurious minutiae can be effectively removed by using some features of these minutiae in the post-processing. This will be discussed later. Our ridge detection algorithm determines the skeleton ridge by searching the maximum and minimum points of the local ridge section set while tracing the gray-level ridge. The low contrast between the ridges and valleys and the contrast inconsistency throughout the fingerprint image will not cause significant inconsistency in the correct detection of the skeleton ridge. Therefore, sharpening of the image is not necessary for our skeleton ridge detection although it will give a better visual appearance of the image. In contrast, image smoothing is very important for reducing the noise, linking the ridge breaks and having the maximal gray values of the image located at the ridge center. In order to avoid the minutia location error and spurious minutiae which may be caused by band-pass filtering due to estimation error of the ridge frequency, we use oriented low-pass filter to reduce the noise of fingerprint image and to link the small breaks in a ridge.

Maio and Maltoni [2] smoothed the fingerprint image with a low-pass filter mask typically of size 3 along the ridge orientation and 7 along the orientation orthogonal to the ridge orientation. However, a filter with a short length along the ridge orientation cannot effectively link the ridge break. A filter with a long length along the orientation orthogonal to the ridge orientation may link two parallel ridges that are not well separated. Therefore, the length of the filter along the ridge orientation should be distinctly greater than along the orientation orthogonal to the ridge orientation.

A low-pass filter can reduce the noise, smooth the small hole and even link the small break in a ridge if the length of the filter mask along the ridge orientation is long enough or the cutoff frequency is low enough. A one-dimensional low-pass filter with very low cutoff frequency along the ridge orientation can be employed because the desired one-dimensional signal along the ridge orientation is a constant in the local window. We choose a Hanning window [27] of length N as our low-pass filter along the ridge orientation:

$$hl(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & 0 \leqslant n \leqslant N-1, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Each pixel of the section set Ω^k is filtered along the ridge orientation by using such a one-dimensional filter of a long fixed length N (typically $N = 11$).

In the orientation orthogonal to the ridge orientation, the low-pass filter should on the one hand, reduce the noise and smooth the small hole while on other hand, not link two not well-separated parallel ridges. The cutoff frequency of the filter along the orientation orthogonal to the ridge orientation should be adaptive to the ridge frequency because the desired one-dimensional signal along the orientation orthogonal to the ridge orientation is a sinusoidal-shape wave. This sinusoidal-shape wave, which has the same frequency as that of the ridges and valleys in the local window, should be allowed to pass through the filter.

The ridge frequency can be estimated based on the calculation of the DFT of the one-dimensional section set Ω^k that is already filtered with $hl(n)$ along the ridge orientation. The filtering along the ridge orientation before the ridge frequency estimation increase the reliability of the ridge frequency estimation. This is possible because the filter parameter along the ridge orientation is independent to the ridge frequency. After the ridge frequency estimation, the filter parameter along the orientation normal to the ridge orientation is adjusted adaptive to the estimated local ridge frequency rf^k or the local ridge distance $rd^k = 1/rf^k$.

We choose a Blackman window [27] of length $M^k + 2$ as our low-pass filter of length M^k along the orientation orthogonal to the ridge orientation.

$$hb^k(m) = \begin{cases} 0.42 - 0.5\cos\left(\dfrac{2\pi m}{M^k + 2 - 1}\right) + 0.08\cos\left(\dfrac{4\pi m}{M^k + 2 - 1}\right), \\ \quad 0 \leqslant m \leqslant M^k + 2 - 1, \\ 0 \quad \text{otherwise.} \end{cases} \tag{5}$$

For a Blackman window of length $M^k + 2$, the length of the filter is actually M^k because $hb^k(0) = hb^k(M + 2 - 1) = 0$. The filter length M^k is chosen to be equal to $round(2/3rd^k)$. Although the filter parameter is also dependent on the ridge frequency, the filtering output is much less sensitive to the accuracy of the local ridge frequency estimation as compared to that of the oriented band-pass filter.

The section set Ω^k is then further filtered along the orientation normal to the ridge orientation. The de-

composition of the filtering into two step, one before the ridge frequency estimation and another after that, not only increases the reliability of the ridge frequency estimation, but also reduces the computation time.

The equivalent two-dimensional oriented filter mask of size $M^k \times N$, which is oriented horizontally to match the horizontally oriented ridges, should be

$$h^k(m, n, 0) = hl\left(n - \frac{N+1}{2}\right) hb^k\left(m - \frac{M^k+1}{2}\right). \tag{6}$$

The equivalent two-dimensional filters for the other orientations can be calculated by rotating this horizontally oriented filter mask. The coefficient of oriented filter with orientation φ at location (m', n') is calculated by rotating by angle φ back to the location (m, n) on the horizontally oriented filter mask as follows:

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} n' \\ m' \end{bmatrix}. \tag{7}$$

Hence, each pixel $g(i, j)$ of section set Ω^k is equivalently convolved with such a oriented filter $h^k(m, n, \varphi)$ as follows:

$$g'(i,j) = \sum_{m=-(M-1)/2}^{(M-1)/2} \sum_{n=-(N-1)/2}^{(N-1)/2} h^k(m, n, n) \cdot g(i-m, j-n). \tag{8}$$

A 5×11 horizontally oriented filter, $h^k(m, n, 0)$, is shown in space domain in Fig. 1 and in frequency domain in Fig. 2.

This filter smoothes the ridge near the point (i^k, j^k). The small gaps caused by noise are smoothed and the small breaks of the ridge caused by scar are linked. Therefore, the filtering accentuates the local maximal gray-level values of the ridge and minimizes the location error of the local maximum of the section set.

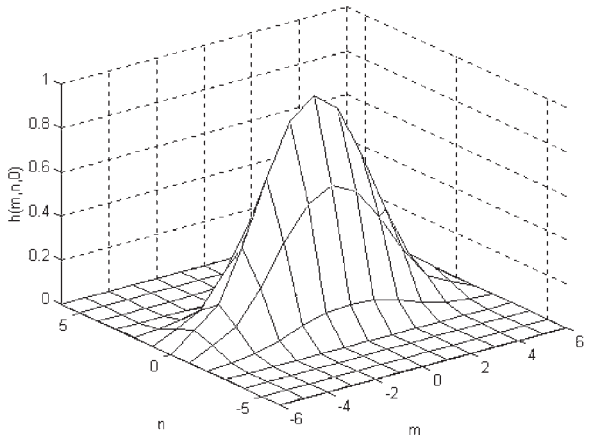

Fig. 1. A horizontal oriented filter ($\varphi = 0$).

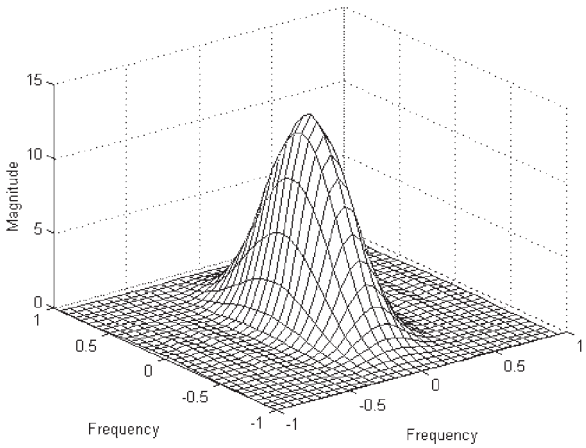

Fig. 2. Frequency response of the filter ($\varphi = 0$).

The proposed filtering method reduces the noise, links the ridge breaks and ensures that the maximal gray values of the image is located at the ridge center. Furthermore, this method does not result in any spurious ridge structure. Although the proposed method can not separate the linked parallel ridges, the spurious minutiae introduced by such defect can be identified and removed in the post-processing stage. In addition, the proposed algorithm decomposes the filtering into two one-dimensional filtering and integrates the three procedures together: section set building, ridge frequency estimation and filtering. This will avoid repeating some of the common computation, which significantly reduces the computation time.

After the filtering, a local maximum $g'(i_m^k, j_m^k)$, $(i_m^k, j_m^k) \in \Omega^k$, which is closest to (i^k, j^k), can be easily found. In addition, we find the local minimum $g'(i_n^k, j_n^k)$, $(i_n^k, j_n^k) \in \Omega^k$, of section set and define

$$d^k = g'(i_m^k, j_m^k) - g'(i_n^k, j_n^k). \tag{9}$$

d^k then represents the local contrast between the smoothed ridge and valley. $g'(i_m^k, j_m^k)$ is a point of the one pixel width ridge line.

2.2. Adaptive tracing the gray-level ridge

So far, given a point (i^k, j^k) and the orientation at this point $\varphi(i^k, j^k)$, we can find a ridge center point (i_m^k, j_m^k) closest to the point (i^k, j^k), its smoothed gray value $g'(i_m^k, j_m^k)$ and local contrast d^k. Initially, we assume a ridge tracing direction at this point $\varphi_d^k = \varphi(i_m^k, j_m^k)$ and then compute the next point (i^{k+1}, j^{k+1}) at a step size μ^k away using

$$\begin{pmatrix} i^{k+1} \\ j^{k+1} \end{pmatrix} = \begin{pmatrix} i_m^k \\ j_k^m \end{pmatrix} + round\left(\mu^k \begin{pmatrix} \sin(\varphi_d^k) \\ \cos(\varphi_k^d) \end{pmatrix}\right) \tag{10}$$

A linear ridge tracing line T^k from point (i_m^k, j_m^k) to point (i^{k+1}, j^{k+1}) is defined as

$$T^k = \{(i,j) | (i,j) \in \mathbf{I},\ i = i_m^k + round(j - j_m^k) \cdot \tan(\varphi_d^k)),$$
$$j \in T_t^k\}, \tag{11}$$

$$T_t^k = [\, j_m^k, j^{k+1}]. \tag{12}$$

The step size, μ^k, of the tracing line is determined by making the following criterion true:

$$g'(i,j) \geqslant g'(i_m^k, j_m^k) - \alpha d^k \ \text{ if } [(i,j) \in T^k] \cap [(i,j) \neq (i^{k+1}, j^{k+1})],$$
$$g'(i^{k+1}, j^{k+1}) < g'(i_m^k, j_m^k) - \alpha d^k, \tag{13}$$

where α is a constant and $0 < \alpha < 1$. $g'(i,j)$ is the smoothed gray value of pixel (i,j) obtained by using the filter $h^k(m,n)$ through Eq. (8).

Obviously, the length of the tracing line, μ^k, is adaptive to the change of the ridge contrast and the bending level of the ridge. A long tracing line will be obtained if there is little variation in the contrast and the bending level of the ridge is low. High bending level of the ridge (possibly facing a ridge bifurcation) or large contrast variation (possibly facing a ridge ending) will result in a short tracing line. Therefore, the algorithm speeds up the tracing while maintaining the tracing precision.

From the point (i^{k+1}, j^{k+1}) we can as before build a section set Ω^{k+1} and find a ridge center point (i_m^{k+1}, j_m^{k+1}), its smoothed gray value $g'(i_m^{k+1}, j_m^{k+1})$ and the local contrast d^{k+1}. The detected ridge is then approximated by a linear one pixel width ridge line, $R^k = [(i_m^k, j_m^k), (i_m^{k+1}, j_m^{k+1})]$. The direction of this linear line, φ_r^k, is calculated using

$$\varphi_r^k = \tan^{-1}\left(\frac{i_m^{k+1} - i_m^k}{j_m^{k+1} - j_m^k}\right), \quad -\pi < \varphi_r^k \leqslant \pi. \tag{14}$$

From the point (i_m^{k+1}, j_m^{k+1}), the ridge will be further traced. To complete a ridge tracing, the algorithm will start from $k = 0$, trace the ridge in one direction (represented with $n = 0$) until a stop criterion becomes true and then restart to trace the ridge from $k = 0$ but in opposite direction ($n = 1$) until one of the stop criteria becomes true. The stop criteria will be discussed later. A ridge is said to be completely traced if the stop criteria become true twice. To keep the ridge being traced toward the forward direction and not backward to the already traced side, the tracing direction is determined as follows:

$$\varphi_d^0 = \begin{cases} \varphi(i_m^0, j_m^0) & \text{if } n = 0, \\ \varphi(i_m^0, j_m^0) - \pi & \text{if } n = 1 \text{ and } \varphi(i_m^0, j_m^0) > 0 \\ \varphi(i_m^0, j_m^0) + \pi & \text{if } n = 1 \text{ and } \varphi(i_m^0, j_m^0) \leqslant 0 \end{cases} \tag{15}$$

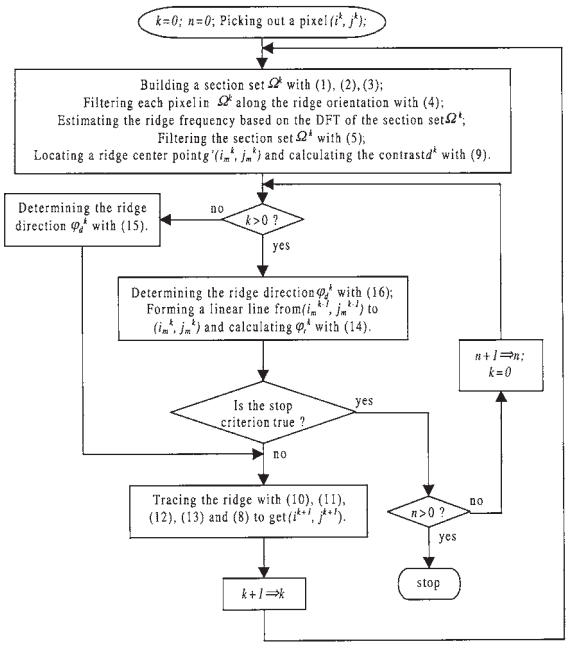

Fig. 3. Flowchart of the ridge tracing algorithm.

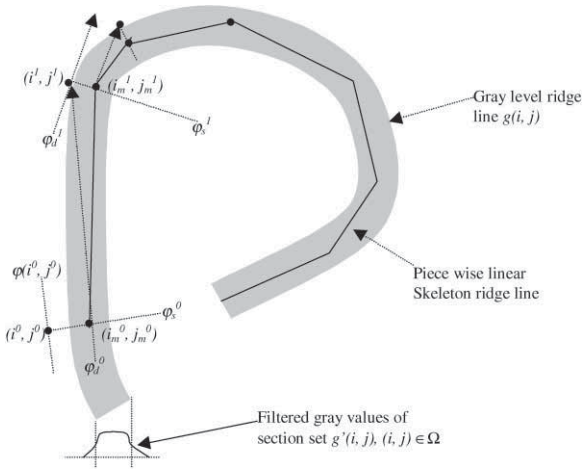

Fig. 4. Forming a skeleton ridge line (piece wise linear one pixel width ridge line) by tracing the gray-level ridge.

and for $k > 0$,

$$\varphi_d^k = \begin{cases} \varphi(i_m^k, j_m^k) & \text{if } |\varphi(i_m^k, j_k^m) - \varphi_d^{k-1}| \leqslant \pi/2, \\ \varphi(i_m^k, j_m^k) - \pi & \text{if } |\varphi(i_m^k, j_k^m) - \varphi_d^{k-1}| > \pi/2,\ \varphi(i_m^k, j_m^k) > 0, \\ \varphi(i_m^k, j_m^k) + \pi & \text{if } |\varphi(i_m^k, j_k^m) - \varphi_d^{k-1}| > \pi/2,\ \varphi(i_m^k, j_m^k) \leqslant 0. \end{cases} \tag{16}$$

Fig. 3 illustrates the flowchart of the adaptive ridge tracing algorithm while Fig. 4 conceptually shows the adaptive tracing process in operation.

3. Minutiae detection

Minutiae will be detected while the ridges are traced and the skeleton image is formed. Tracing should take place only within the region of interest of the fingerprint image. The region of interest is segmented from the image $g(i,j), (i,j)\in\mathbf{I}$, based on the local certainty level $c(i,j)$ of the orientation field $\varphi(i,j)$, which is similar to that used in Ref. [25].

We form the region of interest by initializing the skeleton image $s(i,j), (i,j)\in\mathbf{S}$, which has the same size as $\mathbf{I}$:

$$s(i,j)=\begin{cases}0 & \text{if } (i,j)\in\mathbf{S}\cap g(i,j)\in\text{Interest region},\\ 1 & \text{if } (i,j)\in\mathbf{S}\cap g(i,j)\notin\text{Interest region}.\end{cases} \tag{17}$$

After a ridge of gray-level image $\mathbf{I}$ is traced, the pixels of $\mathbf{S}$ corresponding to the piecewise linear tracing ridge line $R^k(n)\,(n=1,2;k=0,1,2,\ldots)$ are assigned with a number $m\,(m>1)$, called the ridge number, if the length of the ridge is great than a predetermined threshold, μ_r as follows:

$$s(i,j)=m,\ (i,j)\in R_k(n),\ n=0,1;\ k=0,1,2,\ \ldots$$

$$\text{if}\sum_n\sum_k\mu^k(n)>\mu_r. \tag{18}$$

Let (i_t,j_t) denotes a pixel on a tracing ridge line R^k of the skeleton image $\mathbf{S}$ and $N_9(i_t,j_t)$ denotes a set that contain element (i_t,j_t) and its eight neighbors. Let $s_m(i_t,j_t)$ denotes the maximum value of $s(i,j)$ for $(i,j)\in N_9(i_t,j_t)$ where

$$s_m(i_t,j_t)=\max_{i,\,j}(s(i,j)),\quad \text{for } (i,j)\in N_9(i_t,j_t). \tag{19}$$

If a piece of tracing ridge line R^k is within the interest region and do not intersect other already traced ridge line, then $s_m(i_t,j_t)=0$ for all pixel (i_t,j_t) on the tracing ridge line R^k.

While tracing the ridge, minutiae detection is performed by checking for the stop criteria for all pixel (i_t,j_t) on the tracing ridge line R^k. The stop criteria are:

- Tracing exits from interest region. This will take place when $s_m(i_t,j_t)=1$. Tracing will stop without minutiae extraction.
- Tracing ridge line intersects another already traced skeleton ridge line. This will take place when $s_m(i_t,j_t)=l$ with $l>1$. In this case, the intersection of ridge l and current ridge m yields a bifurcation minutiae. Tracing will stop and a bifurcation minutia $[i_t,j_t,\varphi_d^k,c(i_t,j_t),m,l]$ will be recorded if $\sum_n\sum_k\mu^k(n)>\mu_r$. The local certainty level $c(i_t,j_t)$ of the orientation at the point of the minutiae represents a certainty level of this minutiae.
- Tracing ridge line excessively bends. When a ridge ending occurs at (i_m^k,j_m^k), μ^k will be very small and (i_m^{k+1},j_m^{k+1}) will be located in another ridge. Therefore, the difference between the direction φ_r^k and φ_d^k will be greater than the usual threshold value φ_t. In this case, the point (i_m^k,j_m^k) yields an ending minutia. Tracing will stop and an ending minutia $[i_m^k,j_m^k,\varphi_d^k,c(i_m^k,j_m^k),m,0]$ will be recorded if $\sum_n\sum_k\mu^k(n)>\mu_r$. According to this criterion the ridge ending detection is independent of the gray-level of the current region so that the tracing can be performed on both saturated region as well as on contrast deficient region. The ridge ending is detected when the local maximum is not located in a reasonable region of traced ridge but possibly in another ridge. Therefore, the algorithm is robust to the gray value and contrast inconsistency throughout the fingerprint image.

The recorded ridge numbers m and l associated with the minutia and the certainty level of minutia are used for post-processing to remove possible spurious minutiae. The post-processing will be discussed later. Fig. 5 illustrates a flowchart of our minutiae detection algorithm. In Fig. 5, the variable sp is the search step used to examine whether a ridge line should be traced starting from the corresponding point. In order to ensure that all

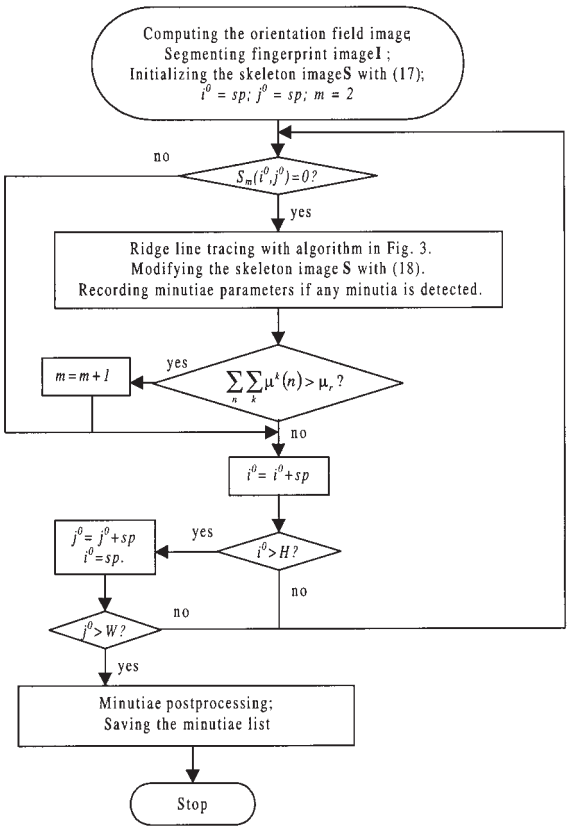

Fig. 5. Flowchart of minutiae detection algorithm.

the ridges of the fingerprint image **I** are traced, the value for *sp* should be less than the smallest possible ridge distance.

4. Post-processing

Minutia was detected in previous section without considering its spatial and structural relationship to other minutiae and ridges in the neighborhood. Inadvertently, some spurious minutiae will be initially detected. In our minutiae detection algorithm that incorporates a thoughtful oriented low-pass filtering and a reasonable limiting of the minimal ridge length have contributed to a lower number of spurious minutiae. Isolated points and blobs do not exist in our skeleton image. Similarly, for extremely short ridges and small islands. However, a wide ridge break caused by a large scar that cannot be linked by oriented smoothing filter, will cause a pair of spurious ridge endings. Moreover ridge cross-connections can arise due to over ink, over press or clutter noise and will result in spurious ridge bifurcation. In addition, a not well smoothed ridge may result in a spur that consists of a spurious ending and a spurious bifurcation. These spurious minutiae can be eliminated based on their spatial, structural and ridge relationships of the minutiae in a neighborhood. Fig. 6(a) shows a spurious minutia pair caused by a wide ridge break due to large scar. Figs. 6(b)–(d) show three spurious minutia pairs caused by ridge cross-connections due to over ink, over press or clutter noise. Fig. 6(e) shows a spurious minutia pair caused by a not well smoothed ridge. The gray-level ridges associated with these spurious minutiae and the possible tracing ridge lines with directions are shown in the figures.

Fig. 7 shows some other possible spurious minutia structures that appeared in the literatures [17,18]. The spurious minutia structure in Fig. 7(a) can be caused by noised thick ridges. Our oriented low-pass filter has the cutoff frequency normal to the ridge orientation being

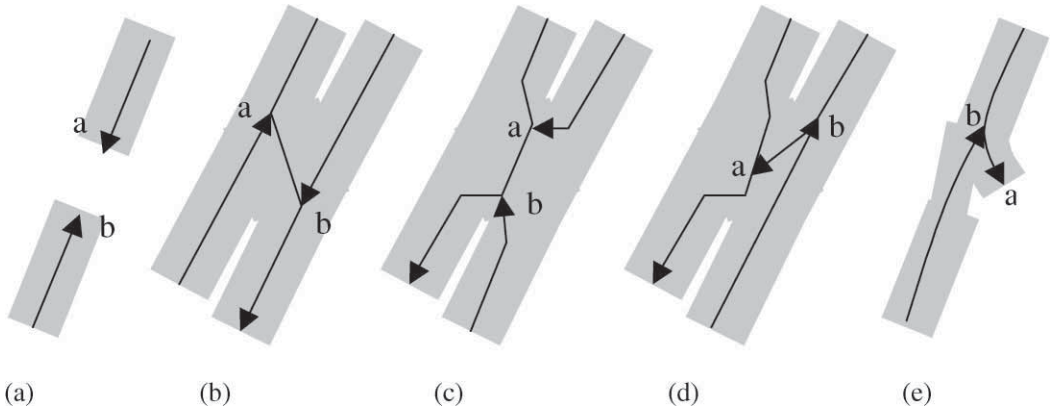

Fig. 6. Some typical spurious minutia structures and the tracing ridge lines.

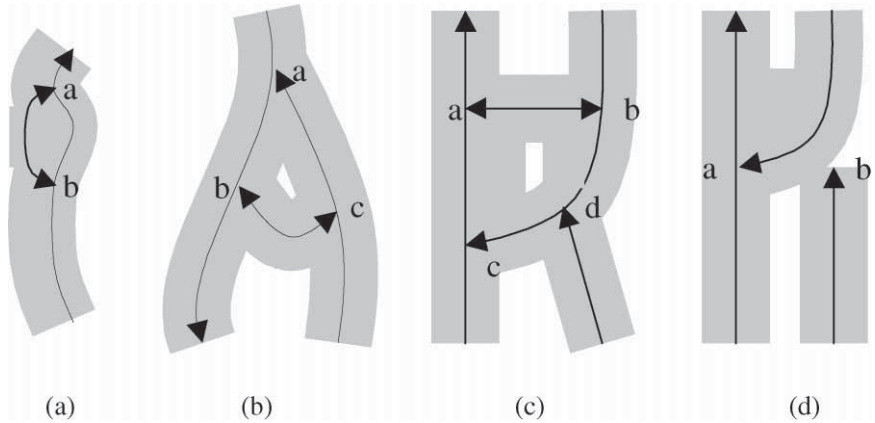

Fig. 7. Some minutia structures not considered in the post processing of this work.

adaptive to the estimated ridge frequency. As a result, the noised thick ridges are well smoothed and usually not split into two skeleton ridges. Both Figs. 7(b) and (c) contains at least one short skeleton ridge, which can be avoided by a reasonable limiting of the minimal skeleton ridge length in our tracing processing. In the absence of the short skeleton ridge, minutiae b and c in Fig. 7(b) will disappear and minutia a should be a genuine minutia. The spurious minutia structure in Fig. 7(c) will convert to the structure in Figs. 6(b)–(d) depending on the tracing processing if the short ridge a–b disappears. The minutia structure as shown as in Fig. 7(d) may occur in our tracing processing. However, such a structure may also be resulted by two genuine minutiae. If we remove the minutiae that have the structure like Fig. 7(d) some genuine minutiae will also be removed. Therefore, there are two reasons for only recognizing the minutia structures in Fig. 6 as the spurious minutia structures that should be removed. One is that the proposed minutiae detection method avoids producing some other minutia structures. Another is that the post-processing should rely on the reliable information because removing spurious minutiae and keeping genuine minutiae are same important.

The crucial step of the post-processing is to differentiate the spurious minutiae from the genuine minutiae. If the spurious minutia structures in Fig. 6 can be correctly and discriminatingly described, removing the spurious minutiae is then an easy task by using these descriptions of the spurious minutia structures. The proposed ridge tracing process provides the ridge tracing direction at the minutia point and the serial number of the skeleton ridge associated with the minutia. These are reliable information that can make the description of the spurious minutia structures more discriminating.

Let the recorded parameters of minutia a in Fig. 6 be $[i_a, j_a, \varphi_{da}, c(i_a, j_a), m_a, l_a]$ while the recorded parameters of minutia b be $[i_b, j_b, \varphi_{db}, c(i_b, j_b), m_b, l_b]$. We define a distance between these two minutiae and the radial angle as follows:

$$d_{ab} = \sqrt{(i_b - i_a)^2 + (j_b - j_a)^2}, \tag{20}$$

$$\varphi_{ab} = \tan^{-1}\left(\frac{i_b - i_a}{j_b - j_a}\right), \quad -\pi < \varphi_{ab} \leqslant \pi. \tag{21}$$

For a spurious minutia pair caused by a wide ridge break such as that in Fig. 6(a), they will satisfy

$$l_a = l_b = 0,$$

$$d_{ab} < D_1,$$

$$\Delta\varphi_a < \Psi_1,$$

$$\Delta\varphi_b > \pi - \Psi_1, \tag{22}$$

where

$$\Delta\varphi_a = \begin{cases} |\varphi_{ab} - \varphi_{da}|, & \text{if } |\varphi_{ab} - \varphi_{da}| \leqslant \pi, \\ 2\pi - |\varphi_{ab} - \varphi_{da}| & \text{if } |\varphi_{ab} - \varphi_{da}| > \pi, \end{cases} \tag{23}$$

$$\Delta\varphi_b = \begin{cases} |\varphi_{ab} - \varphi_{db}|, & \text{if } |\varphi_{ab} - \varphi_{da}| \leqslant \pi, \\ 2\pi - |\varphi_{ab} - \varphi_{db}| & \text{if } |\varphi_{ab} - \varphi_{db}| > \pi, \end{cases} \tag{24}$$

where D_1 and Ψ_1 are the pre-specified constants.

For a spurious minutia pair caused by ridge cross-connections due to over ink, over press or clutter noise as shown in Figs. 6(b)–(d), they will satisfy

$$l_a = l_b > 0 \quad \text{or } (l_b = m_a \text{ and } l_a > 0),$$

$$d_{ab} < D_2,$$

$$m_a \neq m_b,$$

$$abs(\pi - abs(\varphi_{da} - \varphi_{db})) < \Psi_2,$$

$$c(i_a, j_a) < CL \cap c(i_b, j_b) < CL, \tag{25}$$

where D_2, Ψ_2 and CL are the pre-specified constants. CL can be typically equal to or slightly smaller than the mean value of orientation certainty levels at all the minutiae points detected. We incorporate the certainty level of the orientation at minutiae point in the determination of the spurious bifurcation because the over ink, over press or clutter noise which cause the link of ridges often results in a lower orientation certainty levels in the region.

For a spurious minutia pair caused by a not well smoothed ridge as shown in Fig. 6(e), they will satisfy

$$l_b = m_a,$$

$$d_{ab} < D_3,$$

$$m_a \neq m_b,$$

$$abs(\pi - abs(\varphi_{da} - \varphi_{db})) < \Psi_2,$$

$$l_a = 0, \tag{26}$$

where D_3 is a pre-specified constant.

Eqs. (22), (25) and (26) describe and therefore recognize the spurious minutia structures based on the minutia location relationship (d_{ab}, $\Delta\varphi_a$ and $\Delta\varphi_b$), the direction relationship (φ_{da} and φ_{db}), the associate ridge serial number relationship (m_a, l_a, m_b and l_b) and the minutia certainty level ($c(i_a, j_a)$ and $c(i_b, j_b)$). In the proposed post-processing algorithm, we eliminate all minutia pairs from the minutiae list that make the Eqs. (22) and (25) or (26) hold true.

Algorithm. Suppose K minutiae, $K > 1$, were initially detected from the fingerprint image and the parameter of each minutia was recorded with $[i_k, j_k, \varphi_{dk}, c(i_k, j_k), m_k, l_k]$. Initially, all minutiae k, $1 \leqslant k \leqslant K$, are marked with $f_k = 0$. The algorithm starts with $k = 1$.

(1) **If** ($f_k = 1$) go to Step (6); else let $n = k + 1$;
(2) **If** ($f_n = 1$) go to Step (5);
(3) **If** (Eq. (22), (25) or (26) with $a = k$ and $b = n$ holds true) let $f_k = f_n = 1$ and go to Step (6);
(4) **If** (Eq. (25) or (26) with $a = n$ and $b = k$ holds true) let $f_k = f_n = 1$ and go to Step (6);
(5) $n + 1 \Rightarrow n$; **If** ($n \leqslant K$) go back to Step (2);
(6) $k + 1 \Rightarrow k$; **If** ($k < K$) go back to Step (1);
(7) Remove all minutiae k, $1 \leqslant k \leqslant K$, which have $f_k = 1$, from the minutia list.

After the post-processing procedure is performed, a large percentage of the spurious minutiae will be deleted. The surviving minutiae are treated as true minutiae. Our post-processing procedure is not only based on the spatial and structural relationship between the minutiae but also the associated ridge relationship and the certainty level of the minutiae. This makes the task of differentiating spurious minutiae from true minutiae more reliable.

In the proposed minutiae detection algorithm, the procedures of the ridge frequency estimation, image filtering, section set building, ridge tracing and detecting as well as minutiae extraction are not sequentially employed throughout the image one by one, but incorporated and alternately applied to the original raw gray-level fingerprint image. This will avoid repeating some of the common computation, which significantly reduces the computation time. Furthermore, since the tracing steps, μ^k, is adaptive to the degree of ridge bending and ridge contrast variation, our algorithm will be faster than a fixed tracing step while maintaining a good tracing precision. A larger tracing step implies that less section set Ω^k need to be processed. Consequently, only a small portion of the image needs to be smoothed with an oriented low-pass filter. The proposed adaptive oriented low-pass filter will smooth the noise and contrast deficiency and to link small ridge break. The problems of wide ridge breaks due to wide scar and ridge cross-connections due to over ink, over press or clutter noise are solved in the post-processing stage.

Fig. 8(a) shows a fingerprint image and Fig. 8(b) illustrates the detected skeleton ridges and minutiae before post-processing, where the black dots represent bifurcations and the white dots represent endings. Thirty-eight minutiae were initially detected. We see that the proposed approach produced very clean skeleton ridges and contributed to a lower number of spurious minutiae. Isolated points, blobs, extremely short ridges and small islands do not exist in our skeleton image. For a closer examining, Fig. 8(c) shows a combination of Figs. 8(a) and (b). Fig. 8(d) depicts the finally obtained minutiae after post-processing. Six minutiae were removed in the post-processing, where one ending pair satisfied Eq. (22), another bifurcation pair satisfied Eq. (25) and one ending and one bifurcation together made the conditions in Eq. (26) true. For this example the whole time of minutiae extraction taken by a IBM™ PC compatible with Pentium I, 133 MHz processor is 0.892 s, in which the post-processing took only 0.074 s.

5. Experiments and performance evaluation

Like many other problems in the pattern recognition, the performance of the proposed minutiae detection algorithm has to be evaluated through proper experiments. Some experimental evaluation methods for minutiae detection were proposed in Refs. [1,2,7]. Because these evaluation methods depend on the sample minutiae detected by human experts, the database used is very small (several to several tens of fingerprints). However, the result of minutiae detection is heavily dependent on the characteristics and quality of the fingerprint images. As such a meaningful comparison and evaluation can only be achieved with a large database.

In our experiments we avoid the use of sample minutiae detected by a human expert so that a large database can be used in the experiments. Moreover, we include the performance of the fingerprint matching as well since it is also a performance indicator for the minutiae detection process. It is difficult to assess the performance of minutiae detection based solely on the detected minutiae alone.

5.1. Evaluation of the adaptive effects

The proposed algorithm employs an adaptive tracing step as opposed to a fix step as proposed in Ref. [2]. Intuitively, the adaptive step provides a more accurate minutiae location as compared to a large fixed tracing step; or a higher processing speed as compared to a small fixed tracing step. In order to verify this, we proposed a goal-directed performance evaluation to test the minutiae location accuracy and processing speed of the proposed adaptive tracing algorithm relative to various fixed tracing steps.

Let $M_n = (f_n^1, f_n^2, \dots)$ be the set of minutia location vectors detected by minutiae extraction algorithm using a fixed tracing step of n pixels ($n \geqslant 1$). Let M_n with $n = 0$ represent the set of minutia location vectors detected by minutiae extraction algorithm using an adaptive tracing step. Let M_n with minimal tracing step $n = 1$ be a sample minutiae set. The location error of a minutia in M_n related to the kth minutia of the sample minutiae set is defined as

$$e_n^k = \begin{cases} \min(\|f_n^i - f_l^k\|) & \text{if } \min_i (\|f_n^i - f_l^k\|) \leqslant d, \\ d & \text{otherwise,} \end{cases} \tag{27}$$

Fig. 8. A sample fingerprint (a), the detected skeleton ridges and minutiae before the post processing (b), (c) and after the post processing (d).

where d is a constant. In this evaluation, we set d to 10. The location error index is then defined as follows:

$$EI_n = \sqrt{\frac{\sum_{k=1}^{K}(e_n^k)^2}{K \times d^2}}, \tag{28}$$

where K is the number of the minutiae of the sample set. The location error index $EI_n (0 \leqslant EI_n \leqslant 1)$, measures the average normalized location error of minutiae set M_n related to sample minutia set M_1.

Our minutiae detection algorithm was implemented using visual C program in an IBM™ PC compatible with Pentium I, 133 MHz processor. This program was tested on 410 fingerprint images of size 300×300 captured by a Veridicom™ CMOS sensor and on 4000 fingerprint images of size 512×512 from NIST Database 4. Table 1 shows the mean of EI_n values and the processing time of the algorithms with various fixed tracing steps and with an adaptive tracing step. For adaptive step, the average tracing step length is calculated.

Table 1
Means of EI_n values and means of processing times for various fixed tracing steps and adaptive tracing step

For 410 CMOS fingerprint images			For 4000 NIST 4 fingerprint images		
Step length	EI_n	Time	Step length	EI_n	Time
1	0	1.719	1	0	6.93
3	0.33	1.181	3	0.33	4.99
5	0.44	1.079	5	0.39	4.47
7	0.56	0.981	7	0.50	4.19
9	0.67	0.938	9	0.68	4.04
11	0.73	0.895	11	0.83	3.87
Adaptive (11.6)	0.31	0.934	Adaptive (11.2)	0.24	3.90

From the Table 1 we see that the location error index, EI_n, for adaptive tracing step of average length 11 is smaller than those for fixed tracing steps of 3 and above. The processing speed of the algorithm for adaptive tracing step is higher than those for fixed tracing steps of 9 and below. This implies that the minutiae detection with adaptive step outperforms those with fixed steps of between 3 to 9 in both minutiae location error and processing speed.

Examples of detected minutiae with reference fixed tracing step of 1 and the adaptive tracing step are shown in Fig. 9 for a fingerprint image captured using the Veridicom™ sensor and in Fig. 10 for a fingerprint image f0001 in NIST 4 database. No significant difference of minutiae location between the minutiae detected using the minimal tracing step of 1 and that using an adaptive tracing step is found.

The length of the adaptive tracing step depends on the parameter α of Eq. (13). Larger values of α speed up the tracing process but lead to poorer minutiae extraction results. Smaller values of α improve the minutiae extraction performance but consume more computation time. After some trials, α was chosen to be 0.25 as a compromise proposal in all experiments of this work. To illustrate the effectiveness of the proposed algorithm we decreased the value of α and found that the same minutiae detection results as Figs. 9(a) and 10(a) were achieved when the value of α decreased from 0.25 to 0.12. However, the average processing time was increased from 0.934 to 1.135 s for the CMOS fingerprint database and from 3.90 to 4.62 s for the NIST4 fingerprint database, respectively. At the same time, the value of the location error index EI_n was reduced from 0.31 to 0.11 for the CMOS fingerprint database and from 0.24 to 0.16 for the NIST4 fingerprint database, respectively. The average processing time given above and in Table 1 are for the whole minutiae extracting process (including fingerprint image processing, ridge detection, minutiae detection and postprocessing).

5.2. Evaluation using verification performance

After the minutiae in the fingerprint image are detected, the fingerprint can be recognized by matching it with a given template. If the same matching algorithm but different minutiae detection algorithms are used, the matching performance will objectively demonstrate the

Fig. 9. Examples of detected minutiae with fixed step 1(a) and with adaptive step (b) for an image captured with Veridicom CMOS sensor.

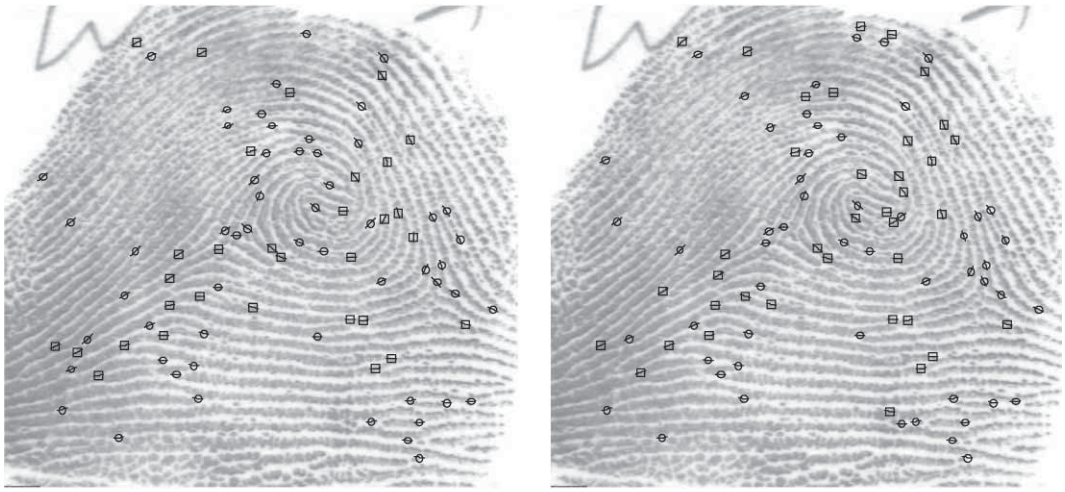

Fig. 10. Examples of detected minutiae with fixed step 1(a) and with adaptive step (b) for image f0001 from NIST 4 database.

performance of the various minutiae detection algorithms. We integrate the minutiae detection algorithm (with various fixed tracing steps and the adaptive tracing step) into our fingerprint verification system. Each of the 410 fingerprint images in our database was matched against the other 409 fingerprint images in the database, which yields 167,690 matching operations. The receiver operating curves (ROCs) [7] are used to provide an empirical assessment of the matching performance.

The ROCs of the tests resulting for different fixed tracing steps (tracing step = 1, 3, 5, 7, and 9) and the adaptive tracing step are shown in Fig. 11. In this figure, the top solid line is the ROC for adaptive tracing step, the dotted line for minimal tracing step 1, the dash-dot line for step size of 3, the dashed line for step size of 5 and the two lower solid lines for step size of 7 and 9, respectively.

From the results, we can observe that the performance of the fingerprint verification using an adaptive tracing step having an equivalent average step length of 11.6, is similar to that obtainable using minimal fixed step size of one. The performance obtained from the adaptive tracing step is better than those obtained using fixed step of sizes 3 and 5 while significantly better than those obtained using fixed step sizes of 7 and 9. In our experiments, we did not directly compare the proposed algorithm with other minutiae detection approaches. Some comparisons of different minutiae detection approaches can be found in Ref. [2].

The performance of the proposed approach was statistically evaluated based on two fingerprint databases that contain fingerprint images of high, middle and poor qualities. Our approach of image filtering, which is adaptive to the ridge orientation and ridge frequency, and the proposed postprocessing approach take efforts to make

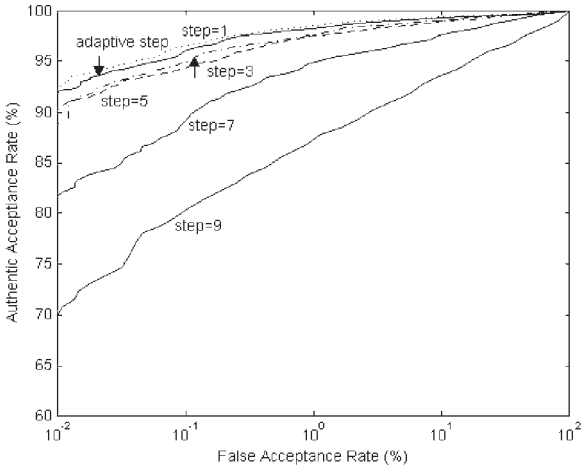

Fig. 11. Receiver operating curves for various fixed tracing steps and adaptive step.

the algorithm robust to the fingerprints of poor quality. Fig. 12(a) shows a fingerprint image of poor quality. The detected skeleton ridges and minutiae before and after postprocessing were respectively, depicted on the fingerprint image in Figs. 12(b) and (c). Sixty-four minutiae were initially detected where only 44 minutiae survived after post processing. From this example we see that the proposed algorithm can also achieve a good performance for the poor-quality fingerprint images. However, if the quality of a fingerprint is too poor to obtain the acceptable estimation error of ridge orientation ($<\pi/4$) or ridge frequency ($<50\%$ of the true ridge frequency) in the majority of the fingerprint area, the proposed approach cannot work properly.

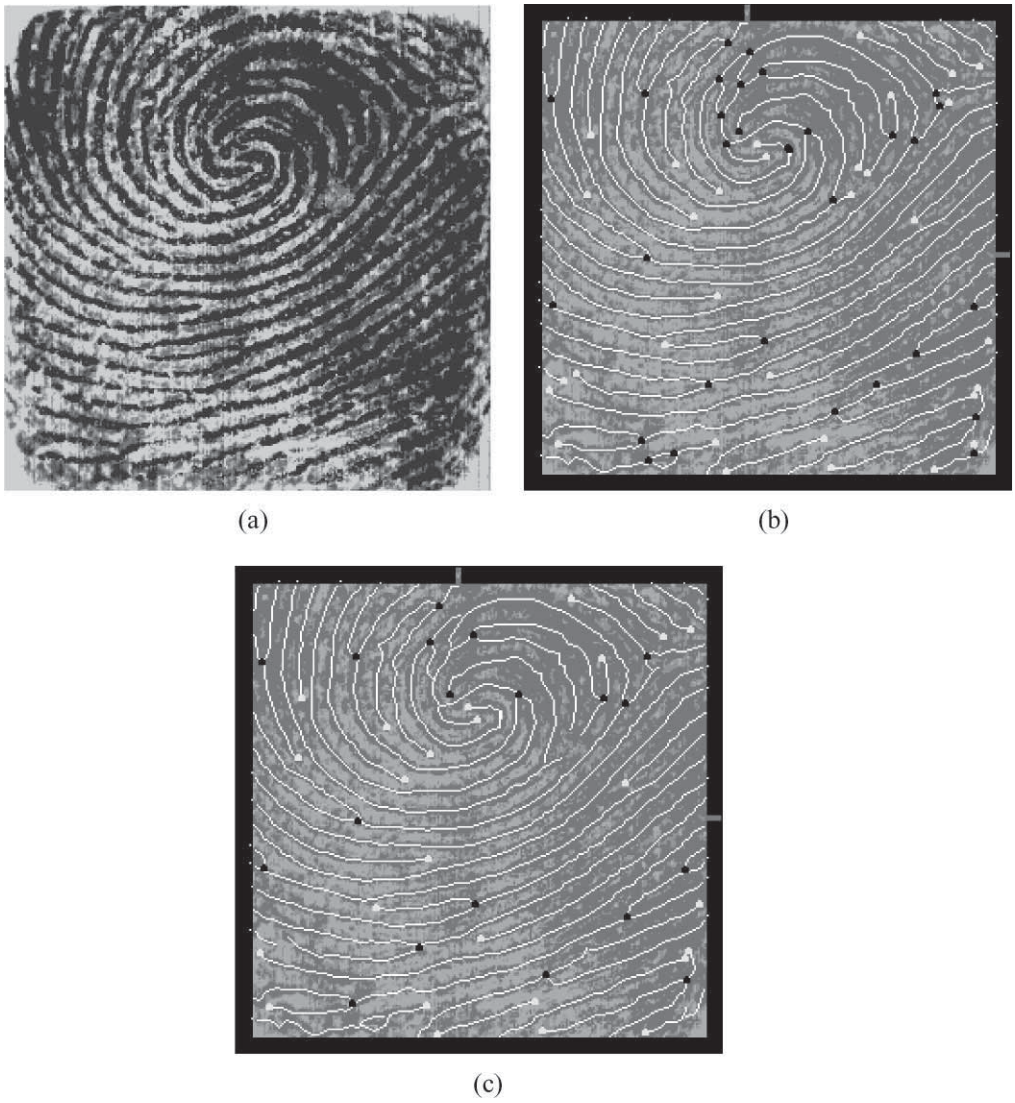

Fig. 12. A poor-quality fingerprint (a), the detected skeleton ridges and minutiae before the post processing (b) and after the post processing (c).

6. Conclusion

We have presented an improved approach of minutiae detection, based on adaptive tracing the gray-level ridge of the original fingerprint image. One of the main advantages of ridge detection by tracing the gray-level ridge is that it approximates the ridge with piece wise linear lines. Ridge detection is therefore not pixel wise but step wise. Because of the large variance of the bending level of the ridge at different points of the image, the tracing step of our algorithm is adaptive to the bending level of the ridge to speed up the tracing while maintaining the tracing precision. The original fingerprint image is smoothed with an adaptive orientated low-pass filter only at some pixels that need to be smoothed. The filter size along the orientation normal to the ridge orientation is adaptive to the estimated ridge distance. This prevents the filtering from linking two not well-separated parallel ridges. In the proposed minutiae detection algorithm, the procedures of the ridge frequency estimation, adaptive image filtering, section set building, adaptive ridge tracing and detecting as well as minutiae extraction are not sequentially employed throughout the image one by one, but incorporated and alternately applied to the original raw

gray-level fingerprint image. This will avoid repeating some of the common computation, which significantly reduces the computation time. After tracing all ridges a piece-wise linear skeleton is obtained. Each ridge in the skeleton is labeled with a number so that each minutia is associated with one or two ridge numbers. Such labeling is shown useful for post-processing. The post-processing is then based not only on the location relationship of the minutiae, but also the associated ridge relationship and the certainty level of the minutiae. This is important because reliably differentiating the false minutiae from true minutiae is crucial for a robust detection of minutiae in noisy fingerprint images. The efficiency and performance of the proposed algorithm have been objectively shown in the experiments using large fingerprint databases.

References

[1] N.K. Ratha, S. Chen, A.K. Jain, Adaptive flow orientation based feature extraction in fingerprint images, Pattern Recognition 28 (1995) 1657.
[2] D. Maio, D. Maltoni, Direct gray-scale minutiae detection in fingerprints, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 27.
[3] B.M. Mehtre, Fingerprint image analysis for automatic identification, Mach. Visual. Appl. 6 (1993) 124.
[4] D.M. Weber, A cost effective fingerprint verification algorithm for commercial application, Proceedings of the South African Symposium on Communication and Signal Processing, 1992, p. 99.
[5] L. O'Gorman, J.V. Nickerson, An approach to fingerprint filter design, Pattern Recognition 22 (1989) 29.
[6] B.G. Sherlock, D.M. Monro, K. Millard, Fingerprint enhancement by directional Fourier filtering, IEEE Proc. Visual. Image Signal Process. 141 (2) (1994) 7.
[7] L. Hong, Y. Wan, A.K. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 777.
[8] M.R. Verma, A.K. Majumdar, B. Chatterjeee, Edge detection in fingerprints, Pattern Recognition 20 (1987) 513.
[9] R.M. Stock, C.M. Swonger, Development and evaluation of a reader of fingerprint minutiae, Cornell Aeronautical Laboratory Technical Report CAL No. XM-2478-X-1, 1969, p. 13.
[10] B. Moayer, K. Fu, A tree system approach for fingerprint pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 376.
[11] R. Stefanelli, A. Rosenfeld, Some parallel thinning algorithms for digital pictures, J. ACM 18 (1971) 255.
[12] A. Rosenfeld, A.C. Kak, Digital Picture Processing, Academic Press, New York, 1976.
[13] H. Tamura, A comparison of line thinning algorithms from digital geometry viewpoint, Proceedings of the Fourth International Joint Conference on Patttern Recognition, Japan, 1978, p. 715.
[14] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital pattern, Commun. of ACM 27 (1984) 236.
[15] C. Arcelli, G.S.D. Baja, A width independent fast thinning algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 7 (1984) 463.
[16] D.C.D. Hung, Enhancement and feature purification of fingerprint images, Pattern Recognition 26 (1993) 1661.
[17] Q. Xiao, H. Raafat, Fingerprint image postprocessing: a combined statistical and structural approach, Pattern Recognition 24 (1991) 985.
[18] A. Farina, Z.M. Kovacs-Vajna, A. Leone, Fingerprint minutiae extraction from skeletonized binary images, Pattern Recognition 32 (1999) 877.
[19] M.T. Leung, W.E. Engeler, P. Frank, Fingerprint image processing using neural network, Proceedings of the 10th Conference on Computer and Communication System, Hong Kong, 1990, p. 582.
[20] M.F. Leung, S.H. Leung, W.H. Lau, A. Luk, Fingerprint recognition using neural network, Proceedings of the IEEE Workshop Neural Network for Signal Processing, 1991, p. 226.
[21] M. Kass, A. Witkin, Analyzing oriented patterns, Comput. Vision Graph. Image Process. 37 (1987) 362.
[22] M. Kawagoe, A. Tojo, Fingerprint pattern classification, Pattern Recognition 17 (1984) 295.
[23] A.R. Rao, A taxonomy for texture description and identification, Springer, New York, 1990.
[24] M.J. Donahue, S.I. Rokhlin, On the use of level curves in image analysis, Image Understanding 57 (1993) 185.
[25] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (1997) 1365.
[26] A.K. Jain, L. Hong, R. Bolle, On-line fingerprint verification, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 302.
[27] A.V. Oppenheim, R.W. Schafer, Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1975.

About the Author—XUDONG JIANG received the B.Eng. and M.Eng. degree from the University of Electronic Science and Technology of China in 1983 and 1986, respectively, and received the Ph.D. degree from the University of German Federal Armed Forces Hamburg, Germany in 1997, all in electrical and electronic engineering. From 1986 to 1993, he was a Teaching Assistant and then a Lecturer at the University of Electronic Science and Technology of China where he was awarded two Science and Technology Progress Prizes from the Ministry for Electronic Industry of China. He was a recipient of the German Konrad-Adenauer Foundation young scientist scholarship. From 1993 to 1997, he was with the University of German Federal Armed Forces Hamburg, Germany as a scientific assistant. Since 1998, he has been with the Centre for Signal Processing, Nanyang Technological University, Singapore as a Research Fellow. His research interests include pattern recognition, neural networks, image processing, computer vision, biometrics, adaptive signal processing and spectral analysis.

About the Author—WEI-YUN YAU received his B.Eng. (Electrical) degree with Honours from the National University of Singapore in 1992. He obtained his M.Eng. in the field of biomedical image processing and Ph.D. in the area of active vision in 1995 and 1999 respectively, both from the Nanyang Technological University. He was a recipient of the Kuok Foundation Undergraduate Scholarship and his undergraduate project won the top prize in the Electronic Engineering/Telecommunications category of the Technology Fair 1992. From June 1992 to March 1997, he was a Research Assistant and then a Teaching Assistant at the Nanyang Technological University, Singapore where he successfully built an active eye-hand coordination system. Currently, he is with the Centre for Signal Processing as a Program Manager and is involved in biometrics, computer vision and image processing related projects.

About the Author—WEE SER obtained the B.Sc. (Hons) degree in 1978 and the Ph.D. degree in 1982, both from the Electronic and Electrical Engineering Department of the Loughborough University of Technology, UK. He completed a General Management Program at the National University of Singapore in 1992 and an executive education program (PMD) at the Harvard Business School in US in 1995. He joined the Defence Science Organization (DSO), Singapore in 1982 and held the position of the Head of the Communication Research Division in 1993. Since 1997, he is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore as an Associate Professor and the Director for the Centre for Signal Processing. He was a recipient of the Colombo Plan scholarship and the PSC Postgraduate scholarship. He was awarded the IEE Prize during his undergraduate studies. While at DSO, he received the Defence Technology (Individual) Prize in 1991 and an Excellence Award in 1992. He is a senior member of IEEE.