Fingerprint Enhancement

Lin Hong[1], Anil Jain[1], Sharathcha Pankanti[2], and Ruud Bolle[2]

[1]Pattern Recognition and Image Processing Laboratory
Department of Computer Science
Michigan State University
East Lansing, MI 48824
{honglin,jain}@cps.msu.edu

[2]Exploratory Computer Vision Group
IBM J. T. Watson Research Center
Yorktown Heights, NY 10598
{sharat,bolle}@watson.ibm.com

Abstract

Fingerprint images vary in quality. In order to ensure that the performance of an automatic fingerprint identification system (AFIS) will be robust with respect to the quality of input fingerprint images, it is essential to incorporate a fingerprint enhancement module in the AFIS system. In this paper, we introduce a new fingerprint enhancement algorithm which decomposes the input fingerprint image into a set of filtered images. From the filtered images, the orientation field is estimated and a quality mask which distinguishes the recoverable and unrecoverable corrupted regions in the input image is generated. Using the estimated orientation field, the input fingerprint image is adaptively enhanced in the recoverable regions. The performance of our algorithm has been evaluated by conducting experiments on an online fingerprint verification system using the MSU fingerprint database containing over 600 fingerprint images. Experimental results show that our enhancement algorithm improves the performance of the online fingerprint verification system and makes it more robust with respect to the quality of input fingerprint images.

1 Introduction

An automatic fingerprint identification system (AFIS) is based on a comparison of minute details of ridge/valley structures of fingerprints [3]. A total of eighteen different types of local ridge/valley descriptions have been identified [6]. Among them, ridge endings and ridge bifurcations (Figure 1(a)), which are usually called minutiae, are the two most prominent structures used in an automatic fingerprint identification system. Automatic and reliable extraction of minutiae from a digital fingerprint image is an extremely difficult task. The performance of currently available minutia extraction algorithms relies heavily on the quality of input digital fingerprint images. Due to a number of factors (aberrant formations of epidermal ridges of fingerprints, postnatal marks, occupational marks, problems with acquisition devices, *etc.*), digital fingerprint images may not always have well-defined ridge/valley structures which the minutia extraction algorithms usually assume. This, of course, results in the failure of minutia extraction algorithms, which, in turn, results in the poor performance of the fingerprint matching module. Figures 1 and 2 show typical examples of applying a minutia extraction algorithm [9] to fingerprint images of both good and poor quality which are acquired with an inkless fingerprint scanner.

In order to ensure that the performance of an automatic fingerprint identification system will be robust with respect to the quality of input fingerprint images, an enhancement algorithm which will improve the clarity of the ridge/valley structures is necessary. Ideally, the ridge/valley structures in a fingerprint image are well-defined. Each ridge is separated by two parallel narrow valleys; each valley is separated by two parallel narrow ridges; and minutiae are defined as the ridge endings and ridge bifurcations. However, in practice, such well-defined ridge/valley structures are not always visible in the scanned fingerprint images. Usually, a fingerprint image is corrupted by various kinds

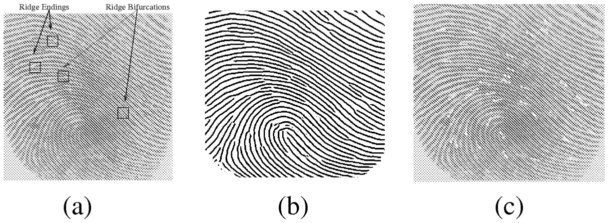


Figure 1. Results of applying a minutia extraction algorithm to a fingerprint image of good quality; (a) input image; (b) extracted ridge map; (c) extracted minutiae superimposed on the input fingerprint image.

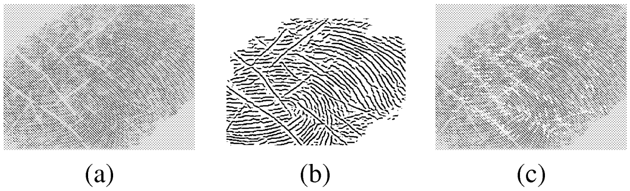


Figure 2. Results of applying a minutia extraction algorithm to a fingerprint image of poor quality; (a) input image; (b) extracted ridge map; (c) extracted minutiae superimposed on the input fingerprint image.

of noise, such as creases, smudges, holes, *etc.* Despite the existence of such noise, a trained fingerprint expert is often able to correctly identify the minutiae by using various visual clues such as local ridge/valley orientation, ridge/valley continuity, *etc.* Therefore, we need to develop an algorithm that can rely on these visual clues to improve the quality of input fingerprint images.

Generally, for a given fingerprint image, the region of interest can be divided into the following three categories:

- Well-defined region, in which ridges and valleys are clearly visible for a minutia extraction algorithm to operate reliably.
- Recoverable corrupted region, in which ridges and valleys are corrupted by a small amount of creases, smudges, *etc.* such that they can still be correctly recovered by an enhancement algorithm.
- Unrecoverable corrupted region, in which ridges and valleys are corrupted by such a severe amount of noise and distortion that it is impossible to recover them from the corrupted image.

We refer to the first two categories of noisy fingerprint regions as recoverable and the last category as unrecoverable. Because it is impossible to recover the true ridge/valley structures in the unrecoverable regions, any effort to improve the quality of the fingerprint image in these regions is futile. Therefore, the goal of a reasonable enhancement algorithm is to improve the clarity of ridge/valley structures of fingerprint images in recoverable regions and to mask out the unrecoverable regions. Another very important aspect concerning a fingerprint enhancement algorithm is that it should not result in any spurious ridge/valley structures.

A number of techniques have been proposed to enhance fingerprint images [1, 4, 5, 7, 11, 16, 17]. These techniques take advantage of the information about the local ridge/valley structures and are capable of adaptively improving the quality of input fingerprint images [1, 4, 5, 7, 17]. However, all of these techniques make an assumption that the local ridge/valley orientations can be reliably estimated from input fingerprint images. In practice, this assumption is not true for fingerprint images of poor quality. Figure 3 shows some examples of estimated orientation field of fingerprint images of poor quality. Therefore, in a fingerprint enhancement algorithm, reliable computation of orientation field is a central issue.

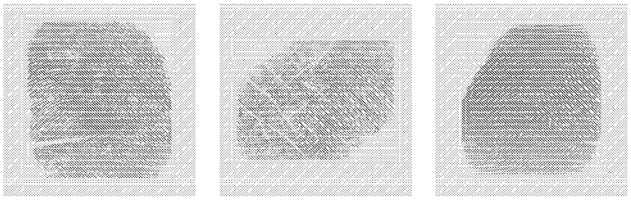


Figure 3. Estimated orientation fields of fingerprint images of poor quality superimposed on input images.

In this paper we introduce a fingerprint enhancement algorithm. As described above, the purpose of this fingerprint enhancement algorithm is to improve the clarity of ridge/valley structures in recoverable regions and make them suitable for minutia extraction algorithms. Our algorithm also identifies all the corrupted regions in which it does not have the capability of recovering the true ridge/valley structures and labels them as unrecoverable regions. The overview of the algorithm is shown in Figure 4. Its main steps are described as follows:

- A bank of even-symmetric Gabor filters is applied to an input fingerprint image and a set of filtered images is produced.
- A ridge extraction algorithm is applied to each of the filtered images and the corresponding ridge map is obtained.
- From the extracted ridge maps of filtered images, a voting algorithm is used to generate a coarse-level ridge

map and unrecoverable-region mask. The generated coarse-level ridge map is used for orientation field estimation.

- An orientation estimation algorithm is applied to the generated coarse-level ridge map, and the local orientation at each pixel is obtained.
- From the computed orientation field and filtered images, an enhanced image is obtained.

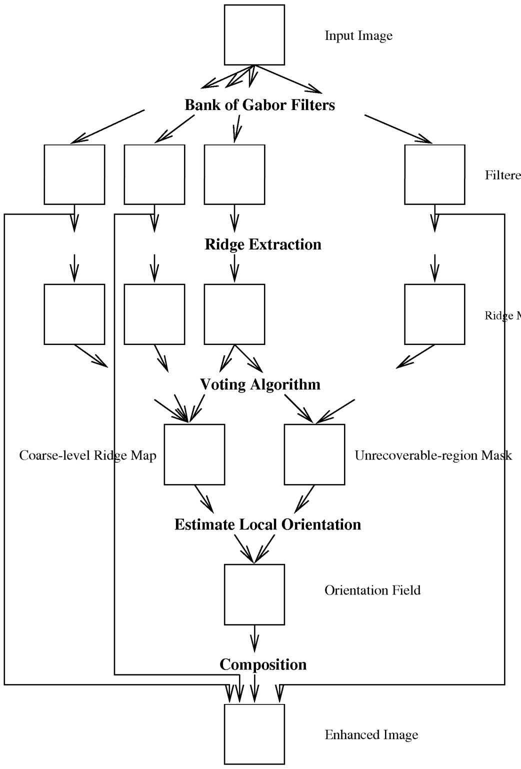


Figure 4. An overview of the fingerprint enhancement algorithm.

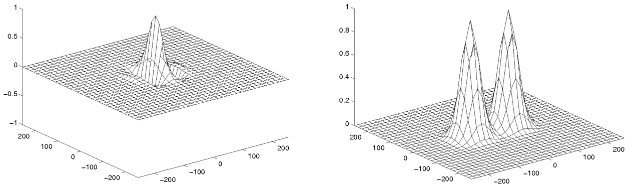
In the following sections we will describe in detail our fingerprint enhancement algorithm. Section 2 addresses the main idea of our algorithm. Experimental results on fingerprint databases captured with inkless scanners are described in section 3. Section 4 contains the summary and discussion.

2 Fingerprint Enhancement

Our fingerprint enhancement algorithm consists of two main stages: (i) orientation field estimation, and (ii) enhancement. Instead of estimating the orientation field directly from the input fingerprint image, we estimate it from the filtered images which is shown to be rather reliable. Because our algorithm can obtain a reliable estimate of the orientation field, a better performance can thus be achieved in the enhancement stage.

2.1 Filtering of Fingerprint Image

Fingerprints are flow-like patterns which consist of locally parallel ridges and valleys. They have well-defined local frequency and local orientation. A set of bandpass filters can efficiently remove the undesired noise and preserve the true ridge/valley structures. Gabor filters have both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains [2, 9]. Therefore, it is beneficial to use Gabor filters as bandpass filters to remove the noise and preserve true ridge/valley structures.



(a) (b)

Figure 5. An even-symmetric Gabor filter: (a) Gabor filter tuned to 60 cycles/width and 0^o orientation; (b) corresponding MTF.

The even-symmetric Gabor filter has the general form

$$h(x,y) = exp\left\{-\frac{1}{2}\left[\frac{x^2}{\delta_x^2}+\frac{y^2}{\delta_y^2}\right]\right\}\cos(2\pi u_0 x), \qquad (1)$$

where u_0 is the frequency of a sinusoidal plane wave along the x-axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively. Gabor filters with arbitrary orientation can be obtained via a rotation of the $x-y$ coordinate system. The modulation transfer function (MTF) of Gabor filter can be represented as

$$\begin{aligned} H(u,v) = 2\pi\delta_x\delta_y exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\delta_u^2}+\frac{v^2}{\delta_v^2}\right]\right\} + \\ 2\pi\delta_x\delta_y exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\delta_u^2}+\frac{v^2}{\delta_v^2}\right]\right\} \quad (2) \end{aligned}$$

where $\delta_u = 1/2\pi\delta_x$ and $\delta_v = 1/2\pi\delta_y$. Figure 5 shows an even-symmetric Gabor filter and its MTF.

An important issue in applying Gabor filters is the selection of filter parameters. We have observed that in a fingerprint image of size 512×512, the ridge frequency is generally around 60 cycles per image width (height). Therefore, in our fingerprint enhancement algorithm, the central frequency is selected as 60 cycles/width (height). The radial bandwidth is selected as 2.5 octaves. Eight values of central orientation θ_0 are used: 0^o, 22.5^o, 45^o, 67.5^o, 90^o, $112, 5^o$, 135^o, 157.5^o. The orientation bandwidth is selected as 35^o. For a given input fingerprint image, these 8 Gabor filters are applied to obtain 8 filtered images. To obtain a filtered image, a FFT is first performed on the input fingerprint image. Then the corresponding Gabor filter with tuned radial and orientation frequency is applied to the frequency image and an inverse FFT is performed to obtain the filtered image. Figures 6(b)-(i) show the eight filtered images for the fingerprint image shown in Figure 6(a).

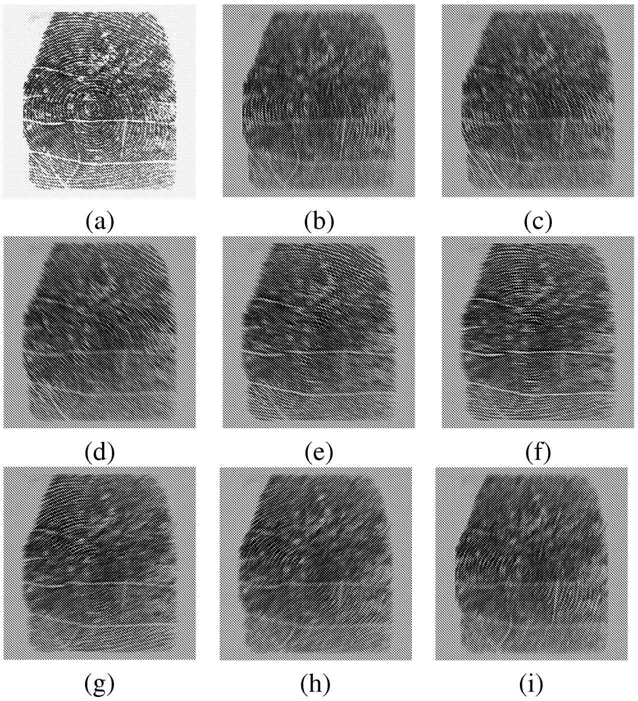


Figure 6. Examples of filtered images for a 512×512 fingerprint image: (a) input image; (b-i) filtered images with Gabor filters tuned to 60 cycles/width and orientations of 0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o, respectively.

2.2 Ridge Extraction

For each filtered image, the following ridge extraction algorithm is applied and the corresponding ridge map is extracted from the filtered image. These ridge maps are used to generate a coarse-level ridge map of the input fingerprint image, which will be introduced in the next section. The most salient property corresponding to ridges in a filtered image is that grey level values on ridges attain their local maxima along a direction that is orthogonal to local ridges. Pixels can be reliably identified to be ridge pixels based on this property. The first step of the ridge extraction algorithm is to estimate the local orientation field according to the following steps:

1. Estimate the local orientation at each pixel (u, v) using the following formula:

$$\begin{aligned} \Delta_x(u,v) &= \sum_{i=1}^{W}\sum_{j=1}^{W} 2G_x(i,j)G_y(i,j), && (3)\\ \Delta_y(u,v) &= \sum_{i=1}^{W}\sum_{j=1}^{W} (G_x^2(i,j) - G_y^2(i,j)), && (4)\\ \theta(u,v) &= \frac{1}{2}tan^{-1}(\frac{\Delta_x(u,v)}{\Delta_y(u,v)}), && (5) \end{aligned}$$

   where W is the size of a local window centered at pixel (u, v) ($W = 15$ in our algorithm); G_x and G_y are the gradient magnitudes in x and y directions, respectively, and $(\Delta_x(u,v), \Delta_y(u,v))$ represents the value of the estimated vector field of the input fingerprint image at the given pixel (u, v). This operator computes the dominant orientation of the Fourier spectrum of the local window centered at pixel (u, v).

2. Compute the *consistency level* of the orientation field in the local neighborhood of a pixel (u, v) with the following formula:

$$\begin{aligned} C_o(u,v) &= \frac{1}{N}\sqrt{\sum_{(i,j)\in D} |\theta(i,j) - \theta(u,v)|^2}, && (6)\\ |\theta' - \theta| &= \begin{cases} d & \text{if } d < 180, \\ d - 180 & \text{otherwise,} \end{cases}, && (7)\\ d &= (\theta' - \theta + 360) mod 360, && (8) \end{aligned}$$

   where D represents the local neighborhood around (u, v), which is a 7×7 local window in our algorithm; N is the number of pixels within D; $\theta(i, j)$ and $\theta(u, v)$ are local ridge orientations at pixels (i, j) and (u, v), respectively.

3. If the *consistency level* is below a certain threshold T_c, then the local orientations in this region are re-estimated at a lower image resolution level until the consistency is above T_c.

After the orientation field is obtained, two adaptive filters are applied to the filtered image:

$$h_t(u,v;i,j) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-i}{\delta^2}}, & \text{if } i = l(j) - d, j \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-i}{\delta^2}}, & \text{if } i = l(j), j \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

$$h_b(u,v;i,j) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-i}{\delta^2}}, & \text{if } i = l(j) + d, j \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-i}{\delta^2}}, & \text{if } i = l(j), j \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

$$l(j) = j \tan(\theta(u,v)), \quad (11)$$

$$d = \frac{H}{2\cos(\theta(u,v))}, \quad (12)$$

$$\Omega = H \left[\left| \frac{\sin(\theta(u,v))}{-2} \right|, \left| \frac{\sin(\theta(u,v))}{2} \right| \right], \quad (13)$$

where $\theta(u,v)$ represents the local ridge orientation at pixel (u,v). These two masks are capable of adaptively accentuating the local maximum grey level values along the normal direction of the local ridge orientation. The filtered image is first convolved with these two masks, $h_t(u,v;i,j)$ and $h_b(u,v;i,j)$. If *both* the grey level values at pixel (u,v) of the convolved images are larger than a certain threshold T_{ridge}, then pixel (u,v) is labeled as a ridge. By adapting the mask width to the width of the local ridge, this algorithm can efficiently locate the ridges in a fingerprint.

Due to the presence of noise, creases, and smudges, *etc.* in the input fingerprint image, the resulting ridge maps of the filtered images often contain a large number of non-ridge pixels being labeled as ridge pixels. A postprocessing step is needed to remove these non-ridge pixels. In our fingerprint enhancement algorithm, we use the following heuristics:

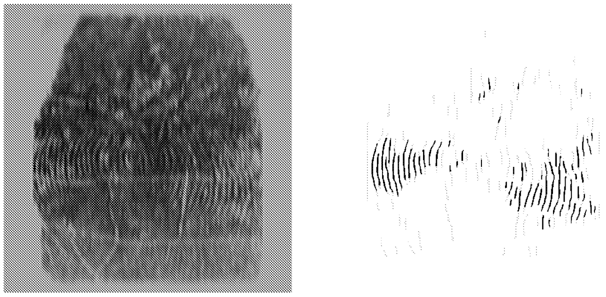
- Compute the area of each connected component appearing in the ridge map. If the area is less than a threshold T_{min}, then label this connected component as background; otherwise break the connected component into a set of short line segments and go to the next step.

- For each short line segment, if it is between a pair of narrow parallel ridges, then label it as a true ridge; otherwise label it as background.

After the above steps have been performed on a ridge map, most of the spurious ridges are removed. (see figure 7).

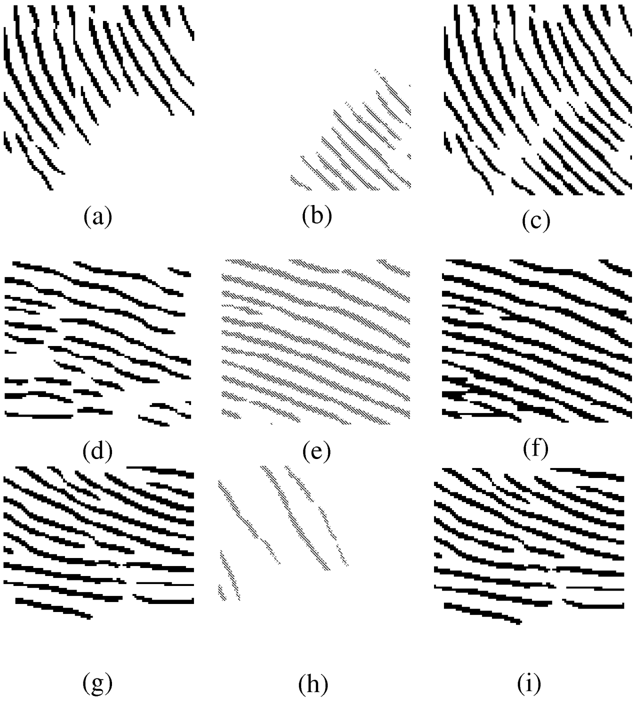
2.3 Coarse-Level Ridge Map and Unrecoverable-region Mask

After the ridge map of each filtered image is obtained, the next step in our fingerprint enhancement algorithm



(a) (b)

Figure 7. The extracted ridge map of a filtered image: (a) filtered image; (b) extracted ridge map; the dark lines represent the detected ridges; grey lines represent the spurious ridges removed by the postprocessing step.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8. Intuitive meaning of the voting algorithm; here for simplicity, we assume that the input image is decomposed into two filtered images; (a)-(c) correspond to rule 1; (d)-(f) correspond to rule 2; (g)-(h) correspond to rule 3; the left two columns show the inputs to the voting algorithm while the third column shows the voting results.

is to generate a coarse-level ridge map and a mask of unrecoverable regions of the input fingerprint image. The coarse-level ridge map is used to estimate a reliable orientation field. The only requirement for the generated coarse-level ridge map is that it should roughly reflect the orientation of the local ridge/valley structures of the input fingerprint image. It is not necessary to impose a requirement that this coarse-level ridge map should be very precise in terms of local ridge structures.

In our enhancement algorithm, the coarse-level ridge map and unrecoverable region mask are generated from the ridge maps of filtered images by using the following voting algorithm:

- Divide each ridge map of filtered images into blocks of size $W \times W$ (8×8 in our algorithm).

- Label each block as foreground (with a value 1) if there are enough ridge pixels appearing around the block; otherwise label it as background (with a value 0). After this process, a binary block map in which a pixel value of 1 represents the existence of ridges and 0 as non-ridges is obtained for each ridge map of filtered images.

- Delete all the connected components (8-connected) in the binary block maps which have an area less than a threshold (16 in our algorithm).

- For each block, examine all the eight filtered images and compute the coarse-level ridge map according to the following rules (an intuitive meaning of these rules is shown in Figure 8):

    1. If only one of the eight binary block map at pixel (x, y) has the value 1 and this pixel belongs to a connected component of size K, $K > T_{block}$, then the pixel values of the corresponding block in the coarse-level ridge map are duplicated from the associated ridge map. The pixel value of the corresponding recoverable region mask is set to the value 0 to indicate that this block is recoverable.
    2. If more than one binary block map at pixel (x, y) has the value 1 and the associated local ridge orientations are not orthogonal with one another, the pixel values of the corresponding block in the coarse-level ridge map are taken as the average values of the associated ridge maps. The pixel value of the corresponding recoverable region mask is set to the value 0 to indicate that this block is recoverable.
    3. If more than one binary block map at pixel (x, y) has the value 1, the associated local ridge orientations may be orthogonal with one another, and only one pixel with the value 1 resides in a connected component of size larger than a certain threshold T_{block}, then the pixel values of the corresponding block in the coarse-level ridge map are duplicated from the ridge map associated with the largest connected component and the pixel value of the corresponding recoverable region mask is set to the value 0 to indicate that this block is recoverable.
    4. If the above conditions are not satisfied, then the block is assigned a label 1 to indicate that it is unrecoverable.

By applying this algorithm to the set of ridge maps of filtered images, a coarse-level ridge map and an unrecoverable region mask are generated.

2.4 Local Orientation Estimation and Enhanced Image

The coarse-level ridge map generated from the ridge maps of the filtered images preserves the local orientation information of the ridge/valley structures of the input fingerprint image. The orientation field of a given input fingerprint image can now be reliably estimated from the coarse-level ridge map by ignoring the unrecoverable regions.

Let $f_i(x, y)$ (i=0, 1, 2, 3, 4, 5, 6, 7) denote the grey level value at pixel (x, y) of the filtered images corresponding to the orientation θ_i, $\theta_i = i * 22.5^o$. After the orientation field has been estimated, the grey level value at pixel (x, y) of the enhanced image can be obtained according to the following formula:

$$\begin{aligned} g(x, y) = \quad & a(x, y) f_{p(x,y)}(x, y) + \\ & (1 - a(x, y)) f_{q(x,y)}(x, y), \end{aligned} \qquad (14)$$

where

$$\begin{aligned} p(x, y) &= \lfloor \frac{\theta(x, y)}{22.5} \rfloor, & (15) \\ q(x, y) &= \lceil \frac{\theta(x, y)}{22.5} \rceil \bmod 8, & (16) \\ a(x, y) &= \frac{\theta(x, y) - p(x, y)}{22.5}, & (17) \\ & & (18) \end{aligned}$$

and $\theta(x, y)$ represents the value of local orientation field at pixel (x, y).

3 Postprocessing

The enhancement algorithm alone can not be expected to perform correctly in all situations. Further, in practice, an enhancement algorithm might introduce artifacts of its own. We, therefore, need to devise postprocessing techniques to reduce extraction of extraneous minutiae due to both of these effects.

The key to designing such postprocessing strategies lies in anticipating these situations where an enhancement algorithm will not be effective. A postprocessing technique is adopted to detect and overcome each of these difficult individual situations. Here, we will illustrate one such postprocessing technique to deal with postnatal cracks that can not be dealt by the enhancement algorithm.

A crack in a fingerprint image could be characterized as a narrow bright region usually running transverse to the dominant ridge direction. One of the heuristics to detect the spurious minutiae resulting from these cracks is based on the observation that these minutiae are anti-aligned and the region between them is brighter than the average brightness of the foreground region. We obtain the information necessary to identify such minutiae from the input fingerprint image and the set of minutiae detected from the enhanced image. The following objective function is used to reveal the spurious minutiae:

$$E = \frac{B}{\log(D+1)(\|(\theta_i - \theta_j)\| + 1)}, \qquad (19)$$

where θ_i and θ_j are the angles subtended by minutiae i and j to a fixed axis of reference, B is the average brightness of the line joining them, and D is the distance between them. The value of E is computed for each pair of minutiae closer than distance D_{th}. If the value of E is greater than E_{th}, both the minutiae are deleted and replaced by a line connecting the original minutiae. This connecting line facilitates obtaining the correct connectivity and ridge count information.

The matching results presented here do not take advantage of this postprocessing heuristics. Integration of the postprocessing strategies into our matching system is currently under way. Figures 4(a)-(c) show the minutiae detected from fingerprint images with cracks without postprocessing and Figures 4(d)-(f) illustrate the minutiae detected from the same images after incorporating the postprocessing step mentioned above. Figure 9 shows the intermediate results of applying the enhancement algorithm to the fingerprint image which is shown in Figure 2.

4 Experimental Results

The purpose of a fingerprint enhancement algorithm is to improve the quality of input fingerprint images and make them more suitable for the minutia extraction module in an AFIS system. Therefore, the ultimate criterion of evaluating such an enhancement algorithm is the amount of performance improvement when the algorithm is applied to the noisy fingerprint images. In order to evaluate the performance of our fingerprint enhancement algorithm, we have conducted two experiments on the online fingerprint verification system [9] using the MSU fingerprint database. The MSU fingerprint database contains 670 images of 67 individuals which were captured with a scanner manufactured by Digital Biometrics. The size of these images is 640 × 480. The fingerprint images in the database vary in quality. More than 90% of the fingerprint images in our database are captured with satisfactory quality, whereas about 10% of the fingerprint images in our database are not of good quality (Figure 11).

In the first experiment, the fingerprint enhancement algorithm was not applied. Each fingerprint image in the database was directly matched against the other fingerprint images in the database. In the second experiment, our fingerprint enhancement algorithm was first applied to each fingerprint image in the database. Then, the verification is conducted on the enhanced fingerprint images. The matching score in our experiment is defined as the square of the number of paired minutiae, normalized by the product of the numbers of minutiae in the input fingerprint image and the template [9]. The distributions of the matching scores

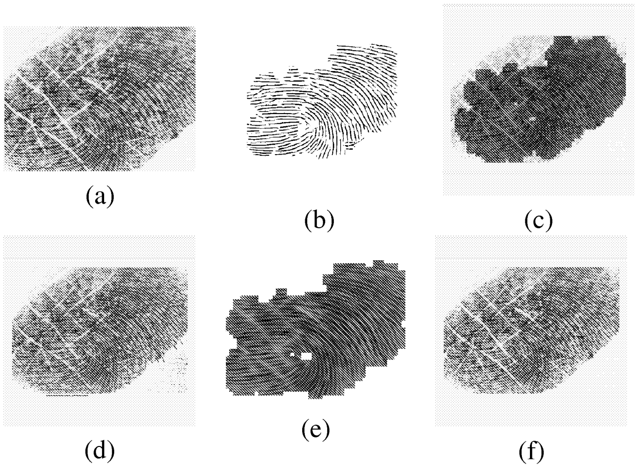


Figure 9. Results of applying the enhancement algorithm to a fingerprint image of poor quality: (a) input image; (b) coarse-level ridge map; (c) unrecoverable-region mask which consists of white pixels; (d) estimated orientation field; (e) enhanced image; (f) minutiae extracted from the enhanced image superimposed on the input image.

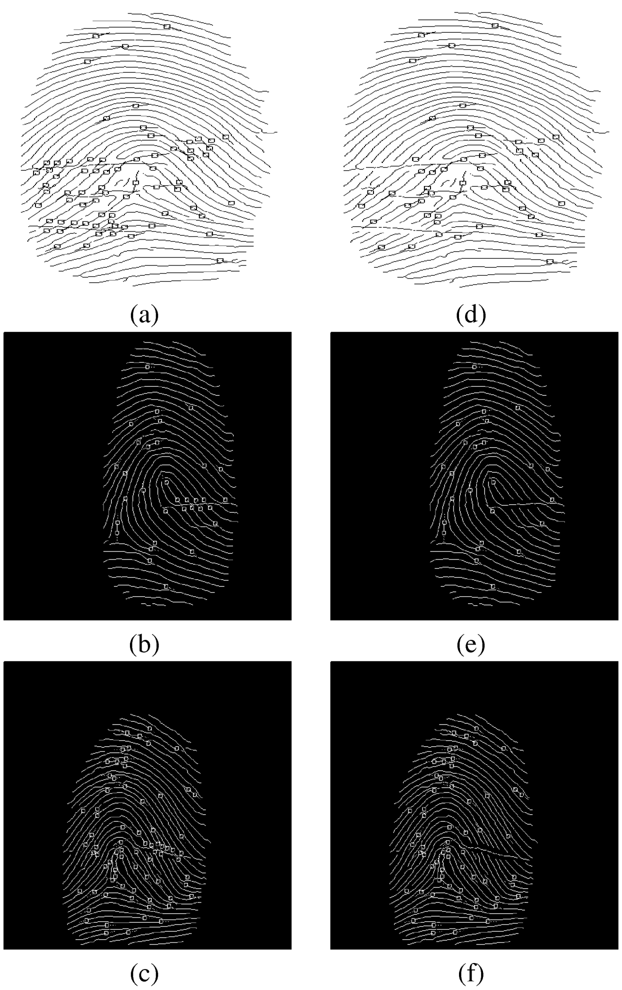


Figure 10. Postprocessing technique for reducing spurious minutiae due to cracks. (a)-(c) show the minutiae before the postprocessing step. (d)-(f) show the minutiae after the postprocessing step.

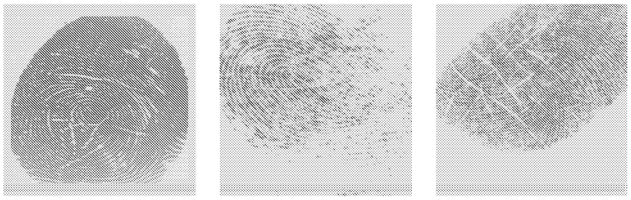


Figure 11. Fingerprint images of poor quality in the MSU fingerprint database.

obtained in the two experiments are shown in Figure 12. Table 1 shows the recognition rates and reject rates with different threshold values on the matching score. From these experimental results, we can observe that the performance of the online fingerprint verification system has significantly improved when our fingerprint enhancement algorithm is applied to the input fingerprint images. In particular, using the enhancement algorithm has substantially reduced the reject rate while maintaining essentially the same recognition rate.

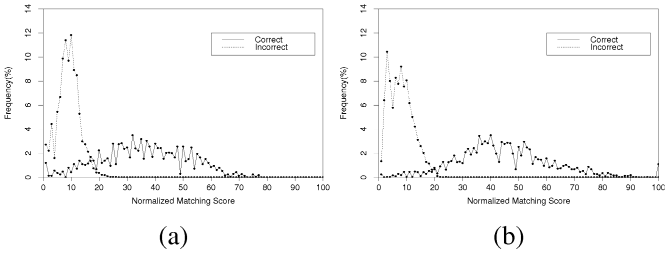


Figure 12. Distributions of correct and incorrect matching scores: vertical axis represents distribution of matching scores in percentage; (a) distribution of matching scores on original fingerprint images; (b) distribution of matching scores on enhanced fingerprint images.

Threshold Value	Recognition Rate	Reject Rate	Recognition Rate Enhanced	Reject Rate Enhanced
20	99.42%	11.23%	99.25%	7.37%
22	99.86%	14.56%	99.95%	9.66%
24	99.89%	16.78%	99.99%	11.07%
26	99.96%	20.20%	100%	14.84%
28	99.98%	23.15%	100%	16.28%
30	99.99%	27.45%	100%	18.21%

Table 1. Recognition and reject rates with different threshold values on the matching score.

5 Conclusions and Future Work

We have introduced a new fingerprint enhancement algorithm. This algorithm, unlike other algorithms, concentrates a large amount of effort on the estimation of the orientation field which plays a critical role in the minutia extraction algorithm. Experimental results show that our algorithm is capable of obtaining a relatively good estimate of orientation field even though the quality of the input fingerprint image is poor. This improvement, in turn, results in an improvement in the quality of input fingerprint images. Our algorithm also picks up the unrecoverable corrupted regions in the fingerprint and mask them out. This is a very important property because such unrecoverable regions do appear in some of the corrupted fingerprint images and they are extremely harmful to minutia extraction. We note that our algorithm does not perform very well around singular regions where ridges and valleys have relatively high curvature values. It tends to mask these regions as unrecoverable regions. However, because minutiae around singular regions are usually assigned lower weights during matching, such a deficiency is not serious. It currently takes approximately 195 seconds for our enhancement algorithm to process one 512×512 fingerprint image on a SPARC 20 workstation. Obviously, this is too slow for an online application. We are studying various methods to speedup the algorithm.

In the process of extracting ridge maps from filtered images, we currently use only the local orientation information and orthogonal peak information. However, we have observed that there is a great difference between the ridge regions and non-ridge regions in terms of texture properties. Therefore, our results can be improve by incorporating texture features in ridge extraction of filtered images. Because this algorithm is slow and since it is futile to improve the clarity of ridge/valley structures of input fingerprint images of good quality, a quality checking module is necessary in an AFIS system. This module checks the quality of the input fingerprint image. The enhancement module is applied to the input fingerprint image if and only if the quality of the input fingerprint image is poor and the true ridge/valley structures are recoverable. We are currently designing such a module.

References

[1] P. E. Danielsson and Q. Z. Ye, Rotation-Invariant Operators Applied to Enhancement of Fingerprints, *Proc. 8th ICPR*, Rome, pp. 329-333, 1988.

[2] J. G. Daugman, Uncertainty Relation for Resolution in Space, Spatial-frequency, and Orientation Optimized by Two-dimensional Visual Cortical Filters, *J. Opt. Soc. Am.* 2, pp 1160-1169, 1985.

[3] Federal Bureau of Investigation, The Science of Fingerprints: Classification and Uses, U.S. Government Printing Office, Washington, D. C. 1984.

[4] D. C. Douglas Hung, Enhancement and Feature Purification of Fingerprint Images, *Pattern Recognition*, Vol. 26, No. 11, pp. 1661-1671, 1993.

[5] L. O'Gorman and J. V. Nickerson, An Approach to Fingerprint Filter Design, *Pattern Recognition*, Vol. 22, No. 1, pp. 29-38, 1989.

[6] Henry C. Lee and R. E. Gaensslen, editors, Advances in Fingerprint Technology, Elsevier, New York, 1991.

[7] T. Kamei and M. Mizoguchi, Image Filter Design for Fingerprint Enhancement, *Proc. ISCV'95*, Coral Gables, FL, pp. 109-114, 1995.

[8] K. Karu and A. K. Jain, Fingerprint Classification, *Pattern Recognition,* Vol. 29, No. 3, pp. 389-404, 1996.

[9] A. K. Jain and F. Farrokhnia, Unsupervised Texture Segmentation Using Gabor Filters, *Pattern Recognition*, Vol. 24, No. 12, pp. 1167-1186, 1991.

[10] A. K. Jain and Lin Hong, On-line Fingerprint Verification, *to appear in Proc. 13th ICPR*, Vienna, Austria, 1996.

[11] Louis Coetzee and Elizabeth C. Botha, Fingerprint Recognition in Low Quality Images, *Pattern Recognition*, Vol. 26, No. 10, pp. 1441-1460, 1993.

[12] B. Miller, Vital Signs of Identity, *IEEE Spectrum,* Vol. 31, No. 2, pp. 22-30, 1994.

[13] A. Ravishankar Rao, A Taxonomy for Texture Description and Identification, Springer-Verlag, New York, 1990.

[14] N. Ratha, K. Karu, S. Chen and A. K. Jain, A Real-time Matching System for Large Fingerprint Database, to appear in the *IEEE Trans. on PAMI*, 1996.

[15] N. Ratha, S. Chen, and A. K. Jain, Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images, *Pattern Recognition*, Vol. 28, No. 11, pp. 1657-1672, 1995.

[16] A. Sherstinsky and R. W. Picard, Restoration and Enhancement of Fingerprint Images Using M-Lattice-A Novel Non-Linear Dynamical System, *Proc. 12th ICPR-B*, Jerusalem, pp. 195-200, 1994.

[17] D. B. G. Sherlock, D. M. Monro, and K. Millard, Fingerprint Enhancement by Directional Fourier Filtering, *IEE Proc. Vis. Image Signal Processing,* Vol. 141, No. 2, pp. 87-94, 1994.

[18] C. I. Watson and C. L. Wilson, NIST Special Database 4, Fingerprint Database, National Institute of Standards and Technology, March 1992.