Fingerprint Image Enhancement: Algorithm and Performance Evaluation

Lin Hong, Yifei Wan, and Anil Jain
Pattern Recognition and Image Processing Laboratory
Department of Computer Science
Michigan State University
East Lansing, MI 48824
{honglin,wanyifei,jain}@cps.msu.edu

Abstract

A critical step in automatic fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint images. However, the performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. In order to ensure that the performance of an automatic fingerprint identification/verification system will be robust with respect to the quality of input fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module. We present a fast fingerprint enhancement algorithm, which can adaptively improve the clarity of ridge and furrow structures of input fingerprint images based on the estimated local ridge orientation and frequency. We have evaluated the performance of the image enhancement algorithm using the goodness index of the extracted minutiae and the accuracy of an online fingerprint verification system. Experimental results show that incorporating the enhancement algorithm improves both the goodness index and the verification accuracy.

1 Introduction

Fingerprint identification is one of the most important biometric technologies which has drawn a substantial amount of attention recently [12, 14]. A fingerprint is the pattern of ridges and furrows on the surface of a fingertip. Each individual has unique fingerprints. The uniqueness of a fingerprint is exclusively determined by the local ridge characteristics

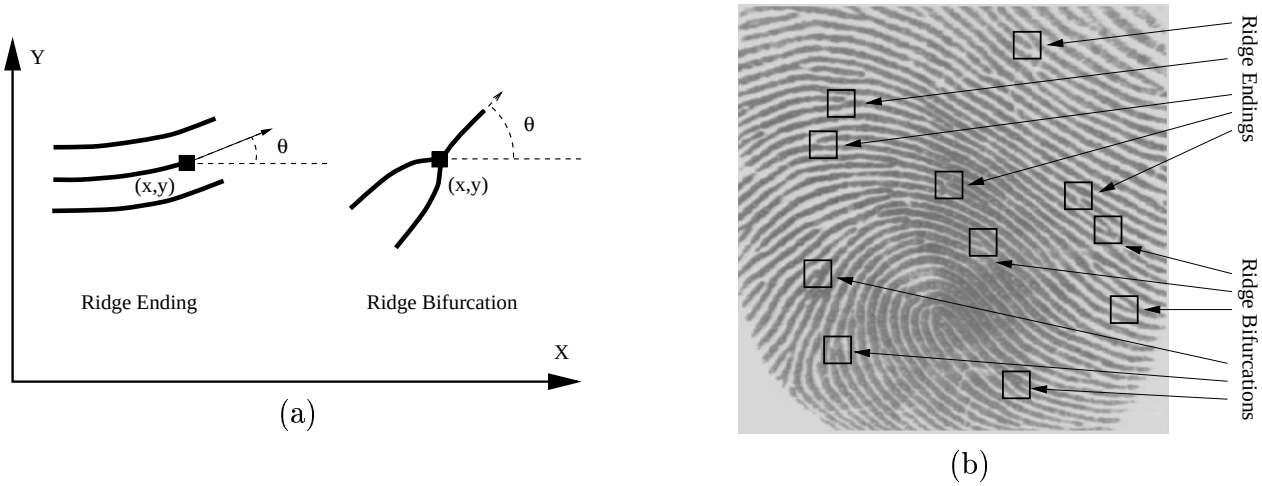


Figure 1: Examples of minutiae; (a) a minutiae can be characterized by its position and its orientation, (b) minutiae overlaid on a fingerprint image.

and their relationships [12, 13]. A total of one hundred and fifty different local ridge characteristics, called minute details, have been identified [13]. These local ridge characteristics are not evenly distributed. Most of them depend heavily on the impression conditions and quality of fingerprints and are rarely observed in fingerprints. The two most prominent ridge characteristics, called minutiae, are (*i*) *ridge ending* and (*ii*) *ridge bifurcation*. A ridge ending is defined as the point where a ridge ends abruptly. A ridge bifurcation is defined as the point where a ridge forks or diverges into branch ridges. A good quality fingerprint typically contains about 40–100 minutiae. Examples of minutiae are shown in Figure 1.

Automatic fingerprint matching depends on the comparison of these local ridge characteristics and their relationships to make a personal identification [12]. A critical step in fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint images, which is a difficult task. The performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. In an ideal fingerprint image, ridges and furrows alternate and flow in a locally constant direction and minutiae are anoma-

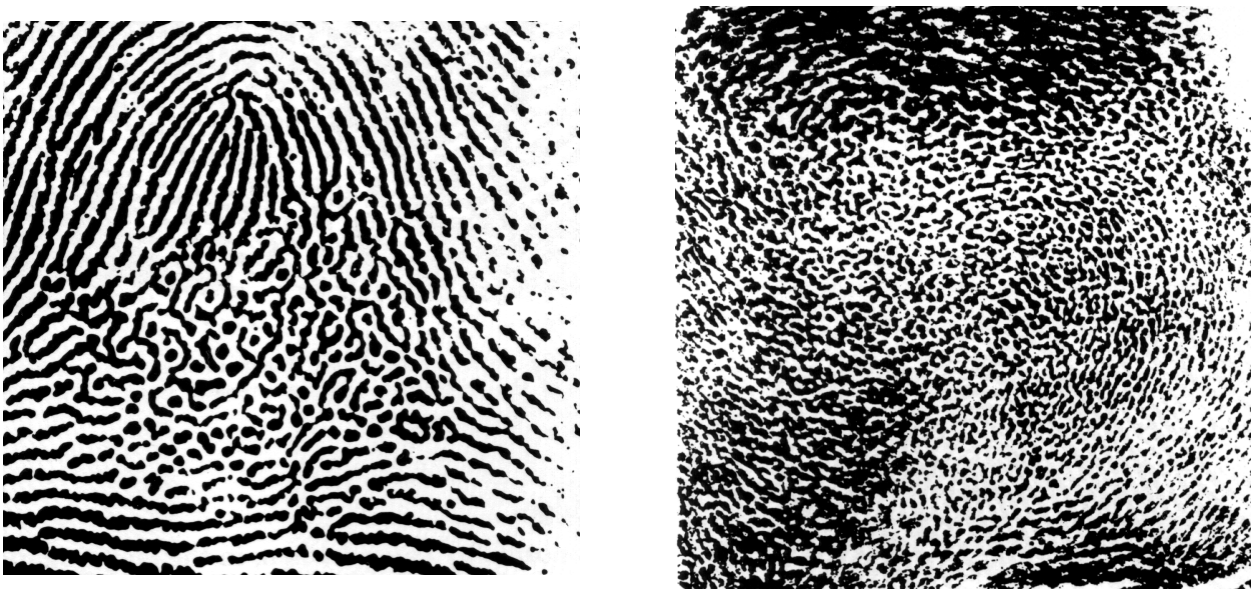
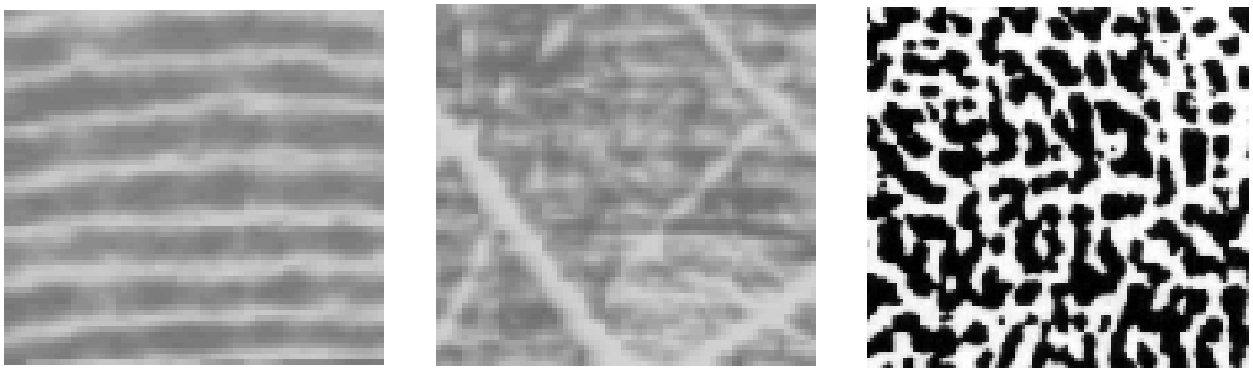


Figure 2: Fingerprint images of very poor quality.

lies of ridges, *i.e.* ridge endings and ridge bifurcations. In such situations, the ridges can be easily detected and minutiae can be precisely located from the binary ridges. Figure 1(b) shows an example of good quality live-scan fingerprint image. However, in practice, due to variations in impression conditions, ridge configuration, skin conditions (aberrant formations of epidermal ridges of fingerprints, postnatal marks, occupational marks), acquisition devices, and non-cooperative attitude of subjects, *etc.* a significant percentage of acquired fingerprint images (approximately 10% according to our experience) is of poor quality. The ridge structures in poor-quality fingerprint images are not always well-defined and hence they can not be correctly detected. This leads to following problems: (*i*) a significant number of spurious minutiae may be created, (*ii*) a large percent of genuine minutiae may be ignored, and (*iii*) large errors in their localization (position and orientation) may be introduced. Examples of fingerprint images of very poor quality, in which ridge structures are completely corrupted, are shown in Figure 2. In order to ensure that the performance of the minutiae



(a) (b) (c)

Figure 3: Fingerprint regions; (a) well-defined region; (b) recoverable corrupted region; (c) unrecoverable corrupted region.

extraction algorithm will be robust with respect to the quality of input digital fingerprint images, an enhancement algorithm which can improve the clarity of the ridge structures is necessary.

A fingerprint expert is often able to correctly identify the minutiae by using various visual clues such as local ridge orientation, ridge continuity, ridge tendency, *etc.*, as long as the ridge and furrow structures are not corrupted completely. It is possible to develop an enhancement algorithm that exploits these visual clues to improve the clarity of ridge structures in corrupted fingerprint images. Generally, for a given digital fingerprint image, the region of interest can be divided into the following three categories (Figure 3):

- *Well-defined region*, where ridges and furrows are clearly differentiated from one another such that a minutiae extraction algorithm is able to operate reasonably.

- *Recoverable corrupted region*, where ridges and furrows are corrupted by a small amount of creases, smudges, *etc.* But, they are still visible and the neighboring regions provide sufficient information about the true ridge and furrow structures.

- *Unrecoverable corrupted region*, where ridges and furrows are corrupted by such a severe amount of noise and distortion that no ridges and furrows are visible and the neighboring regions do not provide sufficient information about the true ridge and furrow structures either.

We refer to the first two categories of regions as *recoverable* and the last category as *unrecoverable*. The goal of an enhancement algorithm is to *improve the clarity of ridge structures* of fingerprint images in recoverable regions and to remove the unrecoverable regions. Since the objective of a fingerprint enhancement algorithm is to improve the clarity of ridge structures of input fingerprint images to facilitate the extraction of ridges and minutiae, a fingerprint enhancement algorithm should not result in any spurious ridge structures. This is very important because spurious ridge structure may change the individuality of input fingerprints.

Fingerprint enhancement can be conducted on either (*i*) *binary ridge images* or (*ii*) *gray-level images*. A binary ridge image is an image where all the ridge pixels are assigned a value 1 and non-ridge pixels are assigned a value 0. The binary image can be obtained by applying a ridge extraction algorithm on a gray-level fingerprint image [6]. Since ridges and furrows in a fingerprint image alternate and run parallel to each other in a local neighborhood, a number of simple heuristics can be used to differentiate the spurious ridge configurations from the true ridge configurations in a binary ridge image [5]. However, after applying a ridge extraction algorithm on the original gray-level images, information about the true ridge structures is often lost depending on the performance of the ridge extraction algorithm. Therefore, enhancement of binary ridge images has its inherent limitations.

In a gray-level fingerprint image, ridges and furrows in a local neighborhood form a

sinusoidal-shaped plane wave which has a well-defined frequency and orientation. A number of techniques that take advantage of this information have been proposed to enhance gray-level fingerprint images [2, 15, 8, 18, 19]. However, they usually assume that the local ridge orientations can be reliably estimated. In practice, this assumption is not valid for fingerprint images of poor quality, which greatly restricts the applicability of these techniques. Hong *etal.* [4] proposed a decomposition method to estimate the orientation field from a set of filtered images obtained by applying a bank of Gabor filters on the input fingerprint images. Although this algorithm can obtain a reliable orientation estimate even for corrupted images, it is computationally expensive which makes it unsuitable for an on-line verification system. We will present a fast enhancement algorithm which is able to adaptively enhance the ridge and furrow structures using both the local ridge orientation and local frequency information. Instead of using a computational expensive method to precisely estimate the local ridge orientation, a simple but efficient method is used. In addition, since this algorithm is designed to be integrated in an online system, a computationally efficient filtering technique is used.

In the following sections we will describe in detail our fast fingerprint enhancement algorithm. Section 2 addresses the main steps of our algorithm. A goal-directed performance evaluation of the implemented fingerprint enhancement algorithm on fingerprint databases is described in section 3. Section 4 contains the summary and discussion.

2 Fingerprint Enhancement

A fingerprint image enhancement algorithm receives an input fingerprint image, applies a set of intermediate steps on the input image, and finally outputs the enhanced image. In order

to introduce our fingerprint image enhancement algorithm, a list of notations and some basic definitions are given below.

2.1 Notation

A *gray-level fingerprint image*, $\mathcal{I}$, is defined as a $N \times N$ matrix, where $\mathcal{I}(i,j)$ represents the intensity of the pixel at the ith row and jth column. We assume that all the images are scanned at a resolution of 500 *dots per inch* (*dpi*), which is the resolution recommended by FBI. The *mean* and *variance* of a gray-level fingerprint image, $\mathcal{I}$, are defined as

$$M(\mathcal{I}) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{I}(i,j) \quad and \tag{1}$$

$$VAR(\mathcal{I}) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\mathcal{I}(i,j) - M(\mathcal{I}))^2, \tag{2}$$

respectively.

An *orientation image*, $\mathcal{O}$, is define as a $N \times N$ image, where $\mathcal{O}(i,j)$ represents the *local ridge orientation* at pixel (i,j). Local ridge orientation is usually specified for a block rather than at every pixel; an image is divided into a set of $w \times w$ non-overlapping blocks and a single local ridge orientation is defined for each block. Note that in a fingerprint image, there is no difference between a local ridge orientation of 90^o and 270^o, since the ridges oriented at 90^o and the ridges oriented at 270^o in a local neighborhood can not be differentiated from each other.

A *frequency image*, $\mathcal{F}$, is a $N \times N$ image, where $\mathcal{F}(i,j)$ represents the *local ridge frequency*, which is defined as the frequency of the ridge and furrow structures in a local neighborhood

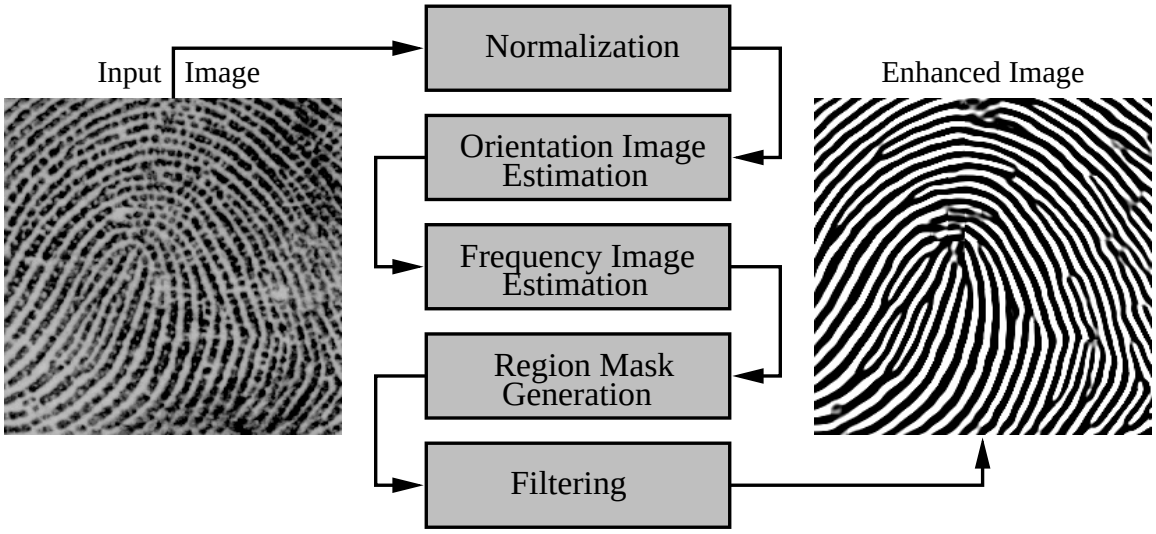


Figure 4: A flowchart of the proposed fingerprint enhancement algorithm.

along a direction normal to the local ridge orientation. The ridge and furrow structures in a local neighborhood where minutiae or *singular points* [9] appear do not form a well-defined sinusoidal-shaped wave. In such situations, the frequency is defined as the average frequency of its neighbors. Like orientation image, frequency image is specified block-wise.

The *region mask*, $\mathcal{R}$, is defined as a $N \times N$ image with $\mathcal{R}(i,j)$ indicating the category of the pixel. A pixel could be either (i) a *non-ridge-and-furrow* (unrecoverable) pixel (with value 0) or (ii) a *ridge-and-furrow* (recoverable) pixel (with value 1). Region mask is also specified block-wise.

2.2 Algorithm

The flowchart of the fingerprint enhancement algorithm is shown in Figure 4. The main steps of the algorithm include:

1. *Normalization*: an input fingerprint image is normalized so that it has a pre-specified mean and variance.

2. *Local orientation estimation*: the orientation image is estimated from the normalized input fingerprint image.

3. *Local frequency estimation*: the frequency image is computed from the normalized input fingerprint image and the estimated orientation image.

4. *Region mask estimation*: the region mask is obtained by classifying each block in the normalized input fingerprint image into a recoverable or a unrecoverable block.

5. *Filtering*: A bank of *Gabor filters* which is tuned to local ridge orientation and ridge frequency is applied to the ridge-and-furrow pixels in the normalized input fingerprint image to obtain an enhanced fingerprint image.

2.3 Normalization

Let $\mathcal{I}(i,j)$ denote the gray-level value at pixel (i,j), M and VAR denote the estimated mean and variance of $\mathcal{I}$, respectively, and $\mathcal{G}(i,j)$ denote the normalized gray-level value at pixel (i,j). The normalized image is defined as follows:

$$\mathcal{G}(i,j) = \begin{cases} M_0 + \sqrt{\frac{VAR_0(\mathcal{I}(i,j)-M)^2}{VAR}}, & \text{if } \mathcal{I}(i,j) > M \\ M_0 - \sqrt{\frac{VAR_0(\mathcal{I}(i,j)-M)^2}{VAR}}, & \text{otherwise,} \end{cases} \tag{3}$$

(4)

where M_0 and VAR_0 are the desired mean and variance values, respectively. Normalization is a pixel-wise operation. It does not change the clarity of the ridge and furrow structures. The main purpose of normalization is to reduce the variations in gray level values along

(a) (b)

Figure 5: The result of normalization; (a) input image; (b) normalized image ($M_0 = 100, VAR_0 = 100$).

ridges and furrows, which facilitates the subsequent processing steps. Figure 5 shows an example of image normalization.

2.4 Orientation Image

The orientation image represents an intrinsic property of the fingerprint images and defines invariant coordinates for ridges and furrows in a local neighborhood. By viewing a fingerprint image as an oriented texture, a number of methods have been proposed to estimate the orientation field of fingerprint images [11, 16, 10, 1]. We have developed a least mean square orientation estimation algorithm. Given a normalized image, $\mathcal{G}$, the main steps of the algorithm are as follows:

1. Divide $\mathcal{G}$ into blocks of size $w \times w$ (16×16).

2. Compute the gradients $\partial_x(i,j)$ and $\partial_y(i,j)$ at each pixel, (i,j). Depending on the

computational requirement, the gradient operator may vary from the simple *Sobel* operator to the more complex *Marr-Hildreth* operator.

3. Estimate the local orientation of each block centered at pixel (i, j) using the following equations:

$$\mathcal{V}_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v), \tag{5}$$

$$\mathcal{V}_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v)), \tag{6}$$

$$\theta(i,j) = \frac{1}{2} tan^{-1}(\frac{\mathcal{V}_y(i,j)}{\mathcal{V}_x(i,j)}), \tag{7}$$

where $\theta(i, j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i, j). Mathematically, it represents the direction that is orthogonal to the dominant direction of the *Fourier spectrum* of the $w \times w$ window.

4. Due to the presence of noise, corrupted ridge and furrow structures, minutiae, *etc.* in the input image, the estimated local ridge orientation, $\theta(i, j)$, may not always be a correct estimate. Since local ridge orientation varies slowly in a local neighborhood where no singular points appear, a low-pass filter can be used to modify the incorrect local ridge orientation. In order to perform the low-pass filtering, the orientation image needs to be converted into a *continuous vector field*, which is defined as follows:

$$\Phi_x(i,j) = \cos(2\theta(i,j)), \quad and \tag{8}$$

$$\Phi_y(i,j) = \sin(2\theta(i,j)), \tag{9}$$

where δ_x and δ_y, are the x and y components of the vector field, respectively. With the resulting vector field, the low-pass filtering can then be performed as follows:

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_x(i-uw, j-vw) \ \ and \tag{10}$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_y(i-uw, j-vw), \tag{11}$$

where W is a 2-dimensional low-pass filter with unit integral and $w_\Phi \times w_\Phi$ specifies the size of the filter. Note that the smoothing operation is performed at the block level. The default size of the filter is 5×5.

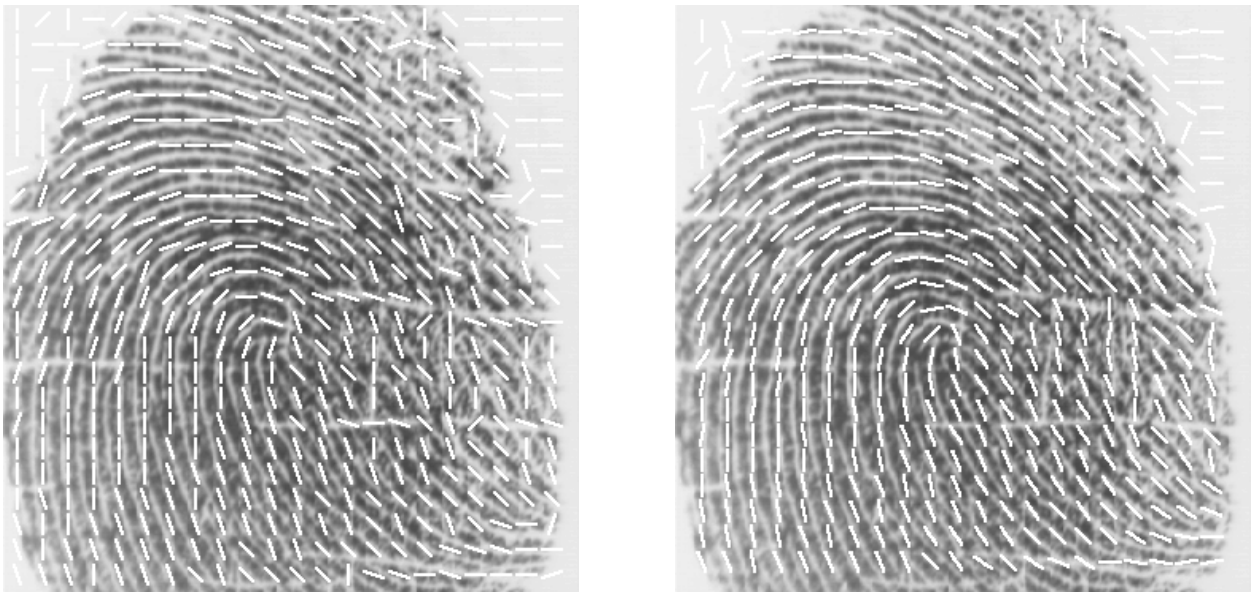
5. Compute the local ridge orientation at (i,j) using

$$\mathcal{O}(i,j) = \frac{1}{2}\tan\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right). \tag{12}$$

With this algorithm, a fairly smooth orientation field estimate can be obtained. Figure 6 shows an example of the orientation image estimated with our algorithm.

2.5 Ridge Frequency Image

In a local neighborhood where no minutiae and singular points appear, the gray levels along ridges and furrows can be modeled as a sinusoidal-shaped wave along a direction normal to the local ridge orientation (see Figure 7). Therefore, local ridge frequency is another intrinsic property of a fingerprint image. Let $\mathcal{G}$ be the normalized image and $\mathcal{O}$ be the orientation image, then the steps involved in local ridge frequency estimation are as follows:



(a) (b)

Figure 6: Comparison of orientation fields by the method proposed in [16] and our method; $w = 16$ and $w_\Phi = 5$.

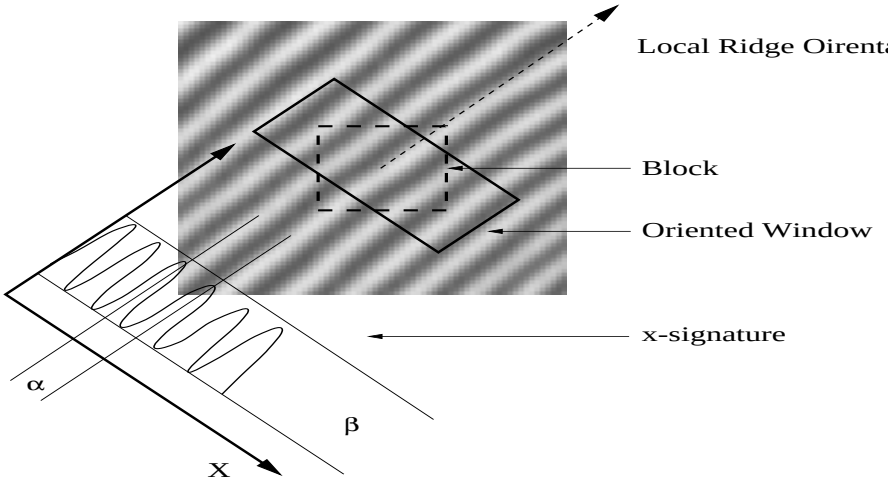


Figure 7: Oriented window and x-signature.

1. Divide $\mathcal{G}$ into blocks of size $w \times w$ (16×16).

2. For each block centered at pixel (i, j), compute an *oriented window* of size $l \times w$ (32×16) that is defined in the ridge coordinates system (Figure 7).

3. For each block centered at pixel (i, j), compute the *x-signature*, $X[0], X[1], ...X[l-1]$, of the ridges and furrows within the oriented window, where

$$X[k] = \frac{1}{w} \sum_{d=0}^{w-1} \mathcal{G}(u, v), \quad k = 0, 1, ..., l-1, \tag{13}$$

$$u = i + (d - \frac{w}{2}) \cos \mathcal{O}(i, j) + (k - \frac{l}{2}) \sin \mathcal{O}(i, j), \tag{14}$$

$$v = j + (d - \frac{w}{2}) \sin \mathcal{O}(i, j) + (\frac{l}{2} - k) \cos \mathcal{O}(i, j). \tag{15}$$

   If no minutiae and singular points appear in the oriented window, the x-signature forms a discrete sinusoidal-shape wave, which has the same frequency as that of the ridges and furrows in the oriented window. Therefore, the frequency of ridges and furrows can be estimated from the x-signature. Let $\mathcal{T}(i, j)$ be the average number of pixels between two consecutive peaks in the x-signature, then the frequency, $\Omega(i, j)$, is computed as: $\Omega(i, j) = 1/\mathcal{T}(i, j)$. If no consecutive peaks can be detected from the x-signature, then the frequency is assigned a value of -1 to differentiate it from the valid frequency values.

4. For a fingerprint image scanned at a fixed resolution, the value of the frequency of the ridges and furrows in a local neighborhood lies in a certain range. For a 500 *dpi* image, this range is $[1/3, 1/25]$. Therefore, if the estimated value of the frequency is out of this range, then the frequency is assigned a value of -1 to indicate that an valid

frequency can not be obtained.

5. The blocks in which minutiae and/or singular points appear and/or ridges and furrows are corrupted do not form a well-defined sinusoidal-shaped wave. The frequency values for these blocks need to be interpolated from the frequency of the neighboring blocks which have a well-defined frequency. The interpolation is performed as follows:

   (i) For each block centered at (i, j),

$$\Omega'(i,j) = \begin{cases} \Omega(i,j), & \text{if } \Omega(i,j) \neq -1 \\ \frac{\sum_{u=-w_\Omega/2}^{w_\Omega/2} \sum_{v=-w_\Omega/2}^{w_\Omega/2} W_g(u,v)\mu(\Omega(i-uw, j-vw))}{\sum_{u=-w_\Omega/2}^{w_\Omega/2} \sum_{v=-w_\Omega/2}^{w_\Omega/2} W_g(u,v)\delta(\Omega(i-uw, j-vw)+1)} & \text{otherwise,} \end{cases} \quad (16)$$

   where

$$\mu(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{otherwise,} \end{cases}$$

$$\delta(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{otherwise,} \end{cases}$$

   W_g is a discrete *Gaussian kernel* where mean and variance is 0 and 9, respectively, and $w_\Omega = 7$ is the size of the kernel.

   (ii) If there exists at least one block with the frequency value of -1, then swap Ω and Ω' and go to step **(i)**.

6. Inter-ridges distances change slowly in a local neighborhood. A low-pass filter can be

used to remove the outliers in f':

$$F(i,j) = \sum_{u=-w_\Omega/2}^{w_l/2} \sum_{v=-w_\Omega/2}^{w_l/2} W_l(u,v)\Omega'(i-uw, j-vw), \tag{17}$$

where W_l is a 2-dimensional low-pass filter with unit integral and $w_l = 7$ is the size of the filter.

2.6 Region Mask

As mentioned early, a pixel (or a block) in an input fingerprint image could be either in a recoverable region or an unrecoverable region. Classification of pixels into recoverable and unrecoverable categories can be performed based on the assessment of the shape of the wave formed by the local ridges and furrows. In our algorithm, three features are used to characterize the sinusoidal-shaped wave: amplitude (α), frequency (β), and variance (γ). Let $X[1], X[2], ..., X[l]$ be the x-signature of a block centered at (i,j). The three features corresponding to pixel (block) (i,j) are computed as follows:

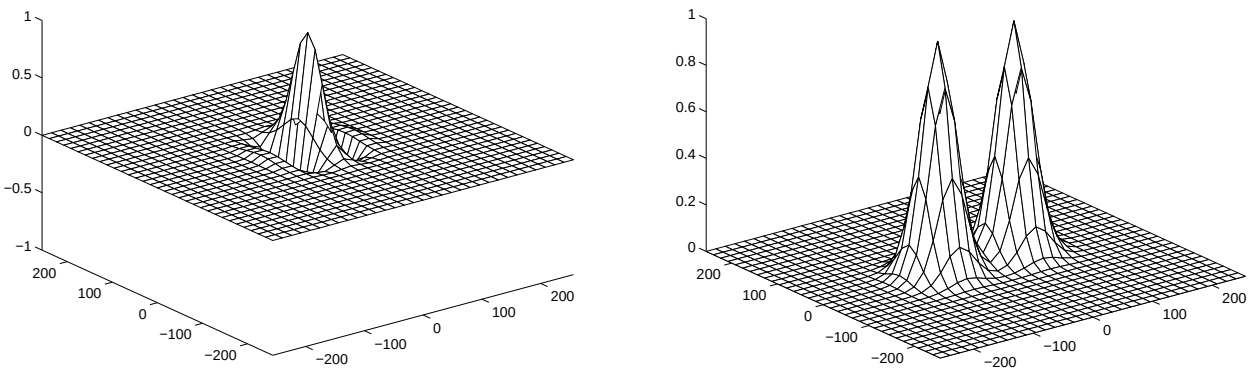
1. $\alpha =$ (average height of the peaks - average depth of the valleys).
2. $\beta = 1/T(i,j)$, where $T(i,j)$ is the average number of pixels between two consecutive peaks.
3. $\gamma = \frac{1}{l}\sum_{i=1}^{l}(X[i] - (\frac{1}{l}\sum_{i=1}^{l} X[i]))^2$.

We selected several typical fingerprint images with both labeled recoverable and unrecoverable regions and computed these three features. A total of 2,000 3-dimensional patterns were obtained. In order to find representative patterns for the two classes, we fed the 2,000

patterns to a squared-error clustering algorithm and identified six clusters. Four of these clusters correspond to recoverable regions and the remaining two correspond to unrecoverable regions. The six prototypes (corresponding to cluster centers) were used in an one-nearest neighbor (1NN) classifier to classify each $w \times w$ block in an input fingerprint image into a recoverable or an unrecoverable block. If a block centered at (i, j) is recoverable, then $\mathcal{R}(i, j) = 1$, else $\mathcal{R}(i, j) = 0$. After the image $\mathcal{R}$ is obtained, the percentage of recoverable regions is computed. If the percentage of recoverable regions is smaller than a threshold, $\Gamma_{recoverable} = 40$, then the input fingerprint image is rejected. An accepted image is then passed through the filtering stage.

2.7 Filtering

The configurations of parallel ridges and furrows with well-defined frequency and orientation in a fingerprint image provide useful information which helps in removing undesired noise. The sinusoidal-shaped waves of ridges and furrows vary slowly in a local constant orientation. Therefore, a bandpass filter that is tuned to the corresponding frequency and orientation can efficiently remove the undesired noise and preserve the true ridge and furrow structures. Gabor filters have both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains [3, 7]. Therefore, it is appropriate to use Gabor filters as bandpass filters to remove the noise and preserve true ridge/valley structures.



(a) (b)

Figure 8: An even-symmetric Gabor filter: (a) the Gabor filter with F being 1/10 and 0^o orientation; (b) the corresponding MTF.

The even-symmetric Gabor filter has the general form [7]

$$h(x, y : \phi, f) = exp\left\{-\frac{1}{2}\left[\frac{(x\cos\phi)^2}{\delta_x^2} + \frac{(y\sin\phi)^2}{\delta_y^2}\right]\right\}\cos(2\pi f x\cos\phi), \tag{18}$$

where ϕ is the orientation of the Gabor filter, f is the frequency of a sinusoidal plane wave, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively. The modulation transfer function (MTF) of the Gabor filter can be represented as

$$\begin{aligned} H(u, v : \phi, f) = \quad & 2\pi\delta_x\delta_y exp\left\{-\tfrac{1}{2}\left[\tfrac{[(u-2\pi/f)\sin\phi]^2}{\delta_u^2} + \tfrac{(v\cos\phi)^2}{\delta_v^2}\right]\right\} + \\ & 2\pi\delta_x\delta_y exp\left\{-\tfrac{1}{2}\left[\tfrac{[(u-2\pi/f)\sin\phi]^2}{\delta_u^2} + \tfrac{(v\cos\phi)^2}{\delta_v^2}\right]\right\} \end{aligned} \tag{19}$$

where $\delta_u = 1/2\pi\delta_x$ and $\delta_v = 1/2\pi\delta_y$. Figure 8 shows an even-symmetric Gabor filter and its MTF.

To apply Gabor filters to an image, three parameters must be specified: (i) the frequency

of the sinusoidal plane wave, u_0, (ii) the filter orientation, and (iii) the standard deviations of the Gaussian envelope, δ_x and δ_y. Obviously, the frequency characteristic of the filter, f, is completely determined by the local ridge frequency and the orientation is determined by the local ridge orientation. The selection of the values of δ_x and δ_y involves a trade-off. The larger the values, the more robust to noise the filters are but the more likely the filters will create spurious ridges and furrows. On the other hand, the smaller the values, the less likely the filters will create spurious ridges and furrows but then they will be less effective in removing the noise. The values of δ_x and δ_y were set to 4.0 and 4.0, respectively based on empirical data. Let $\mathcal{G}$ be the normalized fingerprint images, $\mathcal{O}$ be the orientation image, $\mathcal{F}$ be the frequency image, and $\mathcal{R}$ be the recoverable mask, the enhanced image $\mathcal{E}$ is obtained as follows:

$$\mathcal{E}(i,j) = \begin{cases} 255, & \text{if } \mathcal{R}(i,j) = 0, \\ \sum_{u=-w_g/2}^{w_g/2} \sum_{v=-w_g/2}^{w_g/2} h(u,v : \mathcal{O}(i,j), \mathcal{F}(i,j)) \mathcal{G}(i-u, j-v), & otherwise, \end{cases} \tag{20}$$

where $w_g = 11$ specifies the size of the Gabor filters.

3 Experimental Results

The purpose of a fingerprint enhancement algorithm is to improve the clarity of ridges and furrows of input fingerprint images and make them more suitable for the minutiae extraction algorithm. The ultimate criterion for evaluating such an enhancement algorithm is the total amount of "quality" improvement when the algorithm is applied to the noisy input fingerprint images. Such an improvement can be assessed subjectively by a visual

inspection of a number of typical enhancement results. However, a precise and consistent characterization of the quality improvement is beyond the capability of subjective evaluation. Examples of the enhancement results are shown in Figure 9. From these examples, we can see that our enhancement algorithm does improve the clarity of the ridge and furrow structures of input fingerprint images.

A goal-directed performance evaluation assesses the overall improvement in the system performance that incorporates the enhancement module as a component. Therefore, it is capable of providing a more reliable assessment of the performance benchmark and is directly associated with the ultimate goal of the system [20]. In the following, we present the results of the goal-directed performance evaluation of our enhancement algorithm.

3.1 Evaluation Using Goodness Index

We have used the *goodness index* (GI) of the extracted minutiae to quantitatively assess the performance of our fingerprint enhancement algorithm. Let $M_d = (f_d^1, f_d^2, ..., f_d^n)$ be the set of n minutiae detected by the minutiae extraction algorithm and $M_e = (f_e^1, f_e^2, ..., f_e^m)$ be the set of m minutiae identified by human expert in an input fingerprint image. We define the following terms:

- *Paired minutiae* (p): Minutiae f_d and f_e are said to be paired if f_d is located in a *tolerance box centered around* f_e. In this evaluation, the tolerance box size is 8×8.

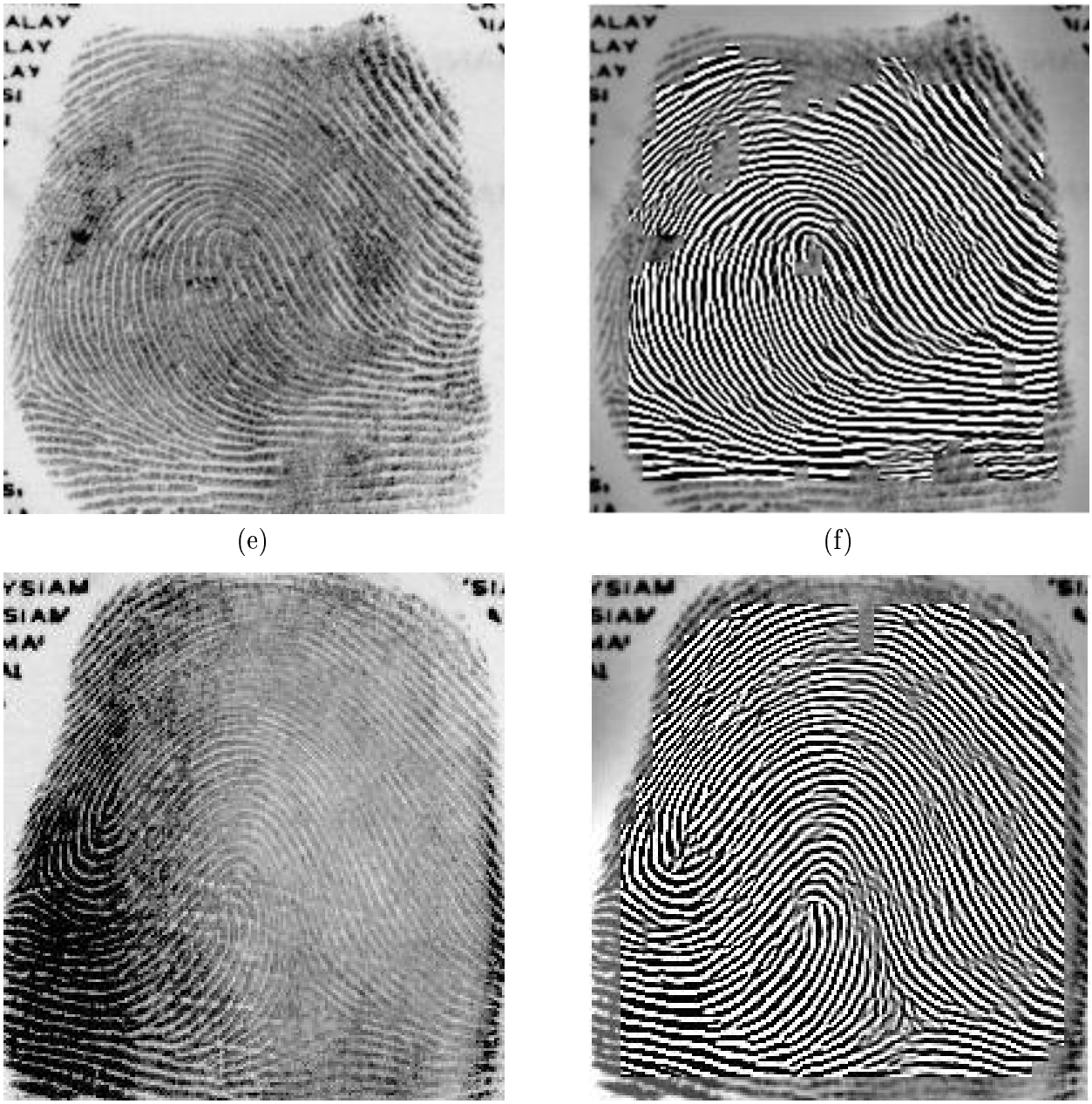
- *Missing minutiae* (a): A minutiae that is not detected by the minutiae extraction algorithm.

(a)

(b)

(c)

(d)



(e) (f)

(g) (h)

Figure 9: Examples of enhancement results; (a), (c), (e), and (g) are the input images; (b), (d), (f) and (h) show enhanced recoverable regions superimposed on the corresponding input images.

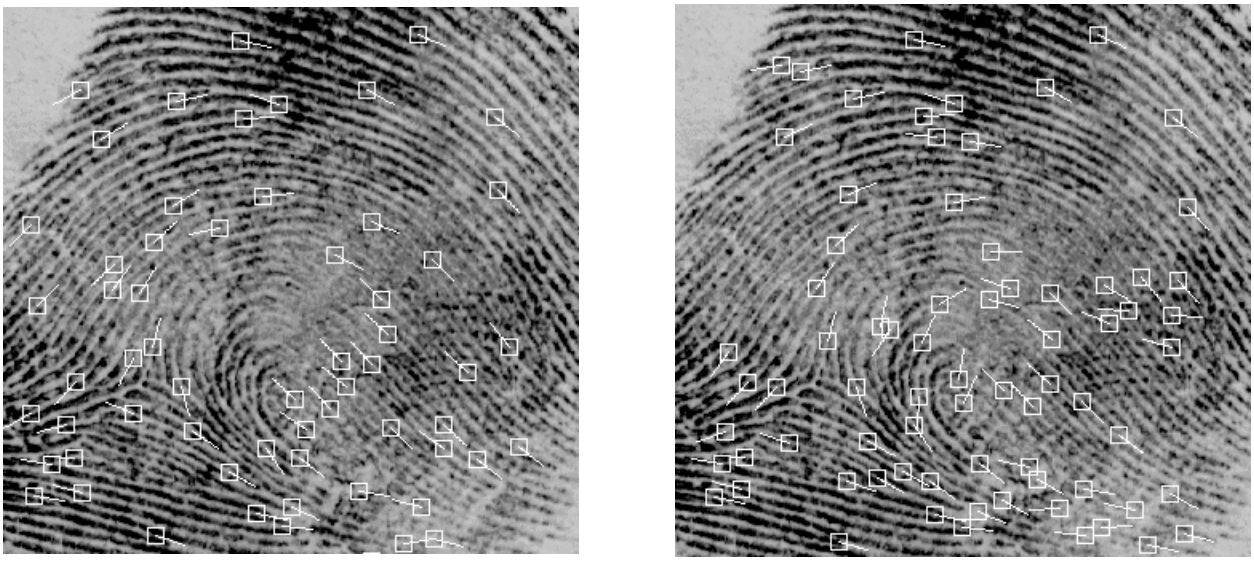
- *Spurious minutiae* (b): A minutiae that is detected by the minutiae extraction algorithm, but which is not in the tolerance box of any minutiae, f_e.

The goodness index (GI) is defined as follows [17]:

$$GI = \frac{\sum_{i=1}^{r} q_i[p_i - a_i - b_i]}{\sum_{i=1}^{r} q_i t_i}, \tag{21}$$

where r is the number of 16×16 windows in the input fingerprint image, p_i represents the number of minutiae paired in the ith window, q_i represents the quality factor of the ith window (good=4, medium=2, poor=1), a_i represents the number of missing minutiae in the ith window, b_i represents the number of spurious minutiae in the ith window, and t_i represents the number of true minutiae in the ith window. GI penalizes both the missing minutiae and spurious minutiae. It is a reasonable measure of the quality of the extracted minutiae. The larger the value of GI, the better the minutiae extraction algorithm. The maximum value of GI equals 1, which means there are no missing and spurious minutiae.

Our fingerprint enhancement algorithm was tested on 50 typical poor fingerprint images obtained from IBM. First, we computed the goodness index of the extracted minutiae without applying the enhancement algorithm and then the goodness index of the extracted minutiae was computed with the enhancement algorithm applied to the input fingerprint images before the minutiae extraction was performed. Examples of minutiae extraction with/without enhancement are shown in Figure 10. Table 1 shows the GI values of 8 typical fingerprint images and the mean and standard deviation of GI values for all the 50 images. The GI values after applying the enhancement algorithm are always larger than that without the



(a) (b)

(c) (d)

Figure 10: Examples of minutiae extraction with/without enhancement; (a) and (c) show the extracted minutiae without applying the enhancement algorithm; (b) and (d) show the extracted minutiae with the enhancement algorithm applied before the minutiae extraction.

Image #	Goodness Index (GI)	
	Without Enhancement	With Enhancement
1	0.46	0.55
2	0.38	0.52
3	0.29	0.42
4	0.26	0.39
5	0.21	0.35
6	0.12	0.31
7	0.11	0.26
8	0.10	0.29
mean	0.24	0.39
std	0.05	0.04

Table 1: The GI values of 8 typical fingerprint images and the mean and standard deviation of 50 IBM fingerprint images.

enhancement algorithm. Thus, we can conclude that our fingerprint enhancement algorithm does improve the quality of the fingerprint images, which, in turn, improves the accuracy and reliability of the extracted minutiae.

3.2 Evaluation Using Verification Performance

The performance of the enhancement algorithm was also assessed on the first volume of the MSU fingerprint database (700 live-scan images; 10 per individual) using the verification accuracy of an online fingerprint verification system [6]. We demonstrated that incorporating the enhancement algorithm in the fingerprint verification system improves the system performance. In the first test, the fingerprint enhancement algorithm was not applied. Each fingerprint image in the data set was directly matched against the other fingerprint images in the database. In the second test, the fingerprint enhancement algorithm was applied to each fingerprint image in the data set. Then, the verification was conducted on the enhanced fingerprint images. The receiver operating curves (ROC) resulting from these two

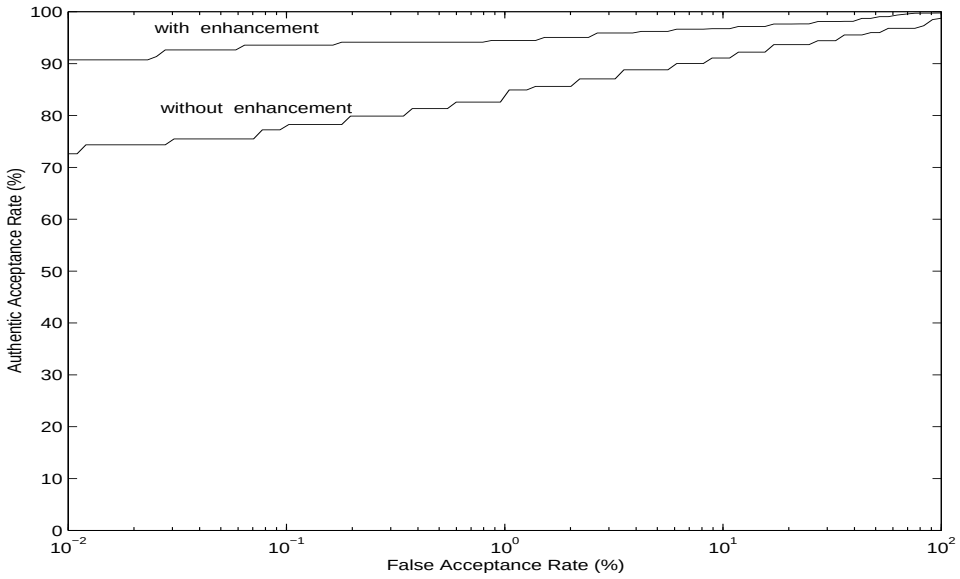


Figure 11: Receiver Operating Curves (ROC); the ROC shows the improvement in verification performance using the enhancement algorithm.

Normalization (seconds)	Orientation (seconds)	Frequency (seconds)	Region Mask (seconds)	Filtering (seconds)	Total (seconds)
0.11	0.14	0.09	0.07	2.08	2.49

Table 2: The wall time of the enhancement algorithm on a Pentium 200MHz PC

tests are shown in Figure 11. From these experimental results, we can observe that the performance of the fingerprint verification system is significantly improved when our fingerprint enhancement algorithm is applied to the input fingerprint images. In particular, the enhancement algorithm substantially reduced the false reject rate while maintaining the same false acceptance rate.

In order to incorporate the enhancement algorithm into an online fingerprint verification/identification system, the whole enhancement process should take only a few seconds. Table 2 shows the wall time for different stages of the enhancement algorithm and the total time.

4 Summary and Conclusions

We have developed a fast fingerprint enhancement algorithm which can adaptively improve the clarity of ridge and furrow structures based on the local ridge orientation and ridge frequency estimated from the inputed images. The performance of the algorithm was evaluated using the goodness index of the extracted minutiae and the performance of an online fingerprint verification system which incorporates our fingerprint enhancement algorithm in its minutiae extraction module. Experimental results show that our enhancement algorithm is capable of improving both the goodness index and the verification performance. The algorithm also identifies the unrecoverable corrupted regions in the fingerprint and removes them from further processing. 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 minutiae extraction. These properties suggest that our enhancement algorithm should be integrated into an online fingerprint verification/identification system.

The global ridge and furrow configuration of fingerprint images presents a certain degree of regularity. A global model of the ridges and furrows that can be constructed from partial "valid" regions can be used to correct the errors in the estimated orientation images, which, in turn, will help the enhancement. Currently, we are investigating such a model-based enhancement algorithm.

The configurations of ridges and furrows within a local neighborhood vary with the quality of input fingerprint images, so a well-defined sinusoidal-shaped waves of ridges and furrows may not always be observed. Global features are needed for a more precise region mask classification.

Acknowledgments

We gratefully acknowledge our many useful discussions with Ruud Bolle, Sharath Pankanti, and Nalini Ratha of the IBM T. J. Watson Research Lab.

References

[1] T. Chang. Texture analysis of digitized fingerprints for singularity detection. In *Proc. 5th ICPR*, pages 478–480, 1980.

[2] P. E. Danielsson and Q. Z. Ye. Rotation-invariant operators applied to enhancement of fingerprints. In *Proc. 9th ICPR*, pages 329–333, Rome, 1988.

[3] J. G. Daugman. Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.*, 2:1160–1169, 1985.

[4] L. Hong, A. K. Jain, S. Pankanti, and R. Bolle. Fingerprint enhancement. In *Proc. 1st IEEE WACV*, pages 202–207, Sarasota, FL, 1996.

[5] D. C. Huang. Enhancement and feature purification of fingerprint images. *Pattern Recognition*, 26(11):1661–1671, 1993.

[6] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Trans. Pattern Anal. and Machine Intell.*, 19(4):302–314, 1997.

[7] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[8] T. Kamei and M. Mizoguchi. Image filter design for fingerprint enhancement. In *Proc. ISCV' 95*, pages 109–114, Coral Gables, FL, 1995.

[9] K. Karu and A. K. Jain. Fingerprint classification. *Pattern Recognition*, 29(3):389–404, 1996.

[10] M. Kass and A. Witkin. Analyzing oriented patterns. *Comput. Vision Graphics Image Process.*, 37(4):362–385, 1987.

[11] M. Kawagoe and A. Tojo. Fingerprint pattern classification. *Pattern Recognition*, 17(3):295–303, 1984.

[12] H. C. Lee and R. E. Gaensslen. *Advances in Fingerprint Technology.* Elsevier, New York, 1991.

[13] A. Moenssens. *Fingerprint Techniques.* Chilton Book Company, London, 1971.

[14] E. Newham. *The Biometric Report.* SJB Services, New York, 1995.

[15] L. O'Gorman and J. V. Nickerson. An approach to fingerprint filter design. *Pattern Recognition*, 22(1):29–38, 1989.

[16] A. R. Rao. *A Taxonomy for Texture Description and Identification.* Springer-Verlag, New York, 1990.

[17] N. Ratha, S. Chen, and A. K. Jain. Adaptive flow orientation based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, 1995.

[18] D. Sherlock, D. M. Monro, and K. Millard. Fingerprint enhancement by directional fourier filtering. *IEE Proc. Vis. Image Signal Processing*, 141(2):87–94, 1994.

[19] A. Sherstinsky and R. W. Picard. Restoration and enhancement of fingerprint images using m-lattice: A novel non-linear dynamical system. In *Proc. 12th ICPR-B*, pages 195–200, 1994.

[20] O. Trier and A. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(12):1191–1201, 1996.