

Fingerprint Recognition

Wenxin Fang, Zimou Zhang

Abstract—abstract

I. INTRODUCTION

Fingerprints are one of the most common and trusted biometrics for personal identification. Because of their uniqueness and consistency over time, fingerprints have been used for identification for over a century. Fingerprint identification is popular because of the inherent ease in acquisition, the numerous sources available for collection, and their established use and collections by law enforcement and immigration.

In this paper, we propose a system

II. FINGERPRINT SEGMENTATION

An important step in fingerprint recognition is the segmentation of the region of interest (ROI). The objective of fingerprint segmentation is to extract the ROI which contains the desired fingerprint impression.

In our fingerprint recognition, we use modified gradient based method proposed in [1]. Steps for this method are summarized as follows:

- 1) Divide the input image $I(i, j)$ into non-overlapping blocks with size $w \times w$. In our implementation, we use $w = 8$.
- 2) Use histogram equalization to enhance the contrast of each block.
- 3) Compute the gradient $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel (i, j) which is the center of the block.
- 4) Compute the standard deviation for both $\partial_x(i, j)$ and $\partial_y(i, j)$ using equation 1 and 2

$$\text{std}_x = \sqrt{\frac{1}{w^2} \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} (\partial_x(i, j) - M_x(I))^2} \quad (1)$$

$$\text{std}_y = \sqrt{\frac{1}{w^2} \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} (\partial_y(i, j) - M_y(I))^2} \quad (2)$$

where M_x and M_y are the mean values of $\partial_x(i, j)$ and $\partial_y(i, j)$ for each block.

- 5) Compute the gradient deviation using equation 3.

$$\text{graddev} = \text{std}_x + \text{std}_y \quad (3)$$

- 6) Select a threshold value empirically. If graddev is greater than threshold value, the block is considered as foreground otherwise it belongs to background.

III. FINGERPRINT ENHANCEMENT

A. Orientation Estimation

We use gradient based orientation estimation method in [2]. Given an image $I(i, j)$, the main steps of the algorithm are as follows:

- 1) Divide $I(i, j)$ into blocks of size $w \times w$. In our implementation, we choose $w = 8$.
- 2) Compute the gradient $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel. In our implementation, we use Sobel operator to compute the gradient.
- 3) Estimate the local orientation of each block centered at pixel (i, j) using the following equations:

$$V_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v)\partial_y(u, v) \quad (4)$$

$$V_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (\partial_x^2(u, v) - \partial_y^2(u, v)) \quad (5)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(i, j)}{V_x(i, j)} \right) \quad (6)$$

where $\theta(i, j)$ is the estimation of the local ridge orientation at the block centered at pixel (i, j) .

- 4) Due to the presence of noise, corrupted ridge, minutiae, etc. some $\theta(i, j)$ may not always be a correct estimate. Because local ridge orientation varies slowly, we can add

a smooth filter to smooth the ridge orientations. In our implementation, we first use Gaussian filter to smooth $V_x(i, j)$ and $V_y(i, j)$ as follow:

$$V'_x(i, j) = \sum_{i-w_g/2}^{i+w_g/2} \sum_{j-w_g/2}^{j+w_g/2} K(u, v) V_x(u, v) \quad (7)$$

where $K(u, v)$ is Gaussian kernel, w_g is the block size of the filter. We choose $\sigma=2$, $w_g = 9$. After smoothing, we can calculate $\theta(i, j)$ using equation 6 except that we use $V'_x(i, j)$ and $V'_y(i, j)$ instead.

B. Ridge Frequency Estimation

Local fingerprint image patches are spatially and spectrally similar to a sinusoidal signal, where the dominant peaks in the magnitude spectrums of the two signals are co-located. The location of the dominant peak in the magnitude spectrum of a local image area carries information about the local orientation and frequency of the fingerprint pattern. Because the sample rate in the image is 1, so when the spectrum maximum is in the edge of the image, the ridge frequency is 0.5. So we can estimate the local ridge frequency using equation

$$f = \frac{D}{N}$$

where D is the distance from the spectrum maximum to the center of the image. We can easily obtain wavelength by reciprocating f .

C. Gabor Filter

Gabor filter is a linear filter used for edge detection. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. Gabor filter have both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains. Therefore, it is appropriate to use Gabor filter as bandpass filter to remove the noise and preserve true ridge structures.

$$g(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \quad (8)$$

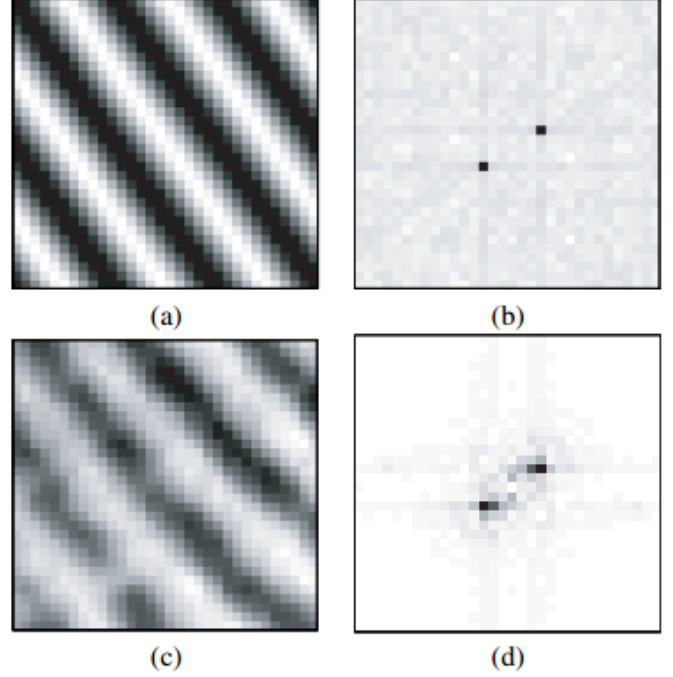


Fig. 1

where

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \end{aligned}$$

θ is the ridge orientation and λ is the ridge wavelength, both of which can be obtained using methods we mentioned above. In our implementation, we set $\gamma = 1$, $\psi = 0$ and $\sigma = 5$.

D. Experimental Results

Some experimental results are shown in Fig. 2. We can see that using Gabor filter, we can remove the noise and preserve true ridge structures.

IV. BINARISATION AND THINNING

After the fingerprint image is enhanced, it is then converted to binary form, and submitted to the thinning algorithm which reduces the ridge thickness to one pixel wide[3]. In our implementation, we simply set a threshold 0, all values large than 0 will be changed to 1, and all values less than 0 will be changed to 0.

After binarisation, we use thinning algorithm proposed in [4]. The main steps of the algorithm is as follows:

- 1) Define neighbor points as in Fig. 3.

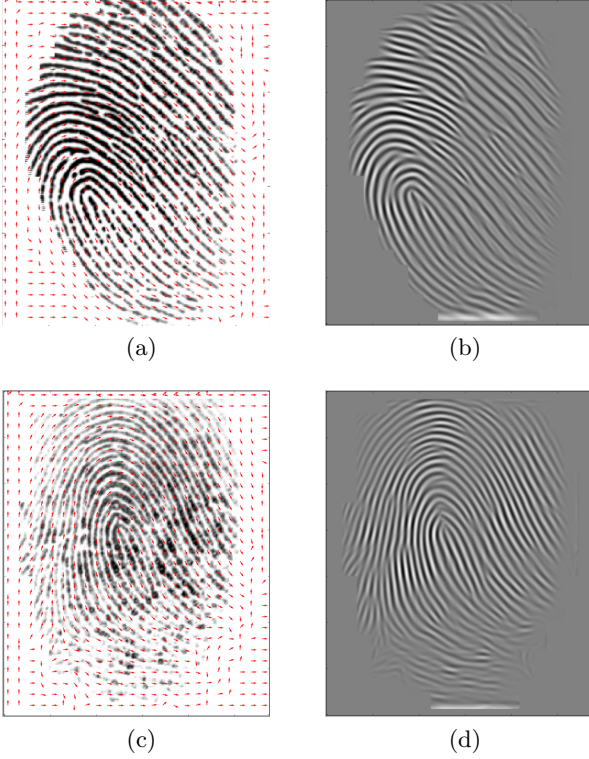


Fig. 2: Some experimental results. (a) Orientation image of a fingerprint image. (b) Enhanced image. (c) Orientation image of another fingerprint image. (d) Enhanced image.

P_9 ($i-1, j-1$)	P_2 ($i-1, j$)	P_3 ($i-1, j+1$)
P_8 ($i, j-1$)	P_1 (i, j)	P_4 ($i, j+1$)
P_7 ($i+1, j-1$)	P_6 ($i+1, j$)	P_5 ($i+1, j+1$)

Fig. 3: Neighbors designations in a 3×3 window

- 2) First subiteration. The point P_1 is deleted from the image if it satisfies the following conditions:
 - a) $2 \leq B(P_1) \leq 6$, where $B(P_1)$ is the number of nonzero neighbors of P_1 , that is, $B(P_1) = P_2 + P_3 + \dots + P_9$.
 - b) $A(P_1) = 1$, where $A(P_1)$ is the number of 01 patterns in the ordered set P_2, P_3, \dots, P_9 .
 - c) $P_2 \times P_4 \times P_6 = 0$
 - d) $P_4 \times P_6 \times P_8 = 0$
- 3) Second subiteration, only conditions c) and d) are changed as follows:



Fig. 4: (a)(c) Binarisation images. (b)(d) Thinning images.

$$c') P_2 \times P_4 \times P_8 = 0$$

$$d') P_2 \times P_6 \times P_8 = 0$$

- 4) Repeat step 2 and step 3, until no point in the image can be deleted.

After binarisation and thinning, we can obtain images as shown in Fig. 4.

V. MINUTIAE EXTRACTION

A. Minutiae Extraction

After a fingerprint image has been enhanced and thinned, the next step is to extract the minutiae from the image. The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept. This method extracts the ridge endings and bifurcations from the skeleton image by examining the local neighborhood of each ridge pixel using a 3×3 window. The CN for a ridge pixel P is given by

$$CN = 0.5 \sum_{i=2}^9 |P_i - P_{i+1}|, \quad P_{10} = P_2 \quad (9)$$

where P_i is the neighbor pixel as in Fig. 3. After the CN for a ridge pixel has been computed,

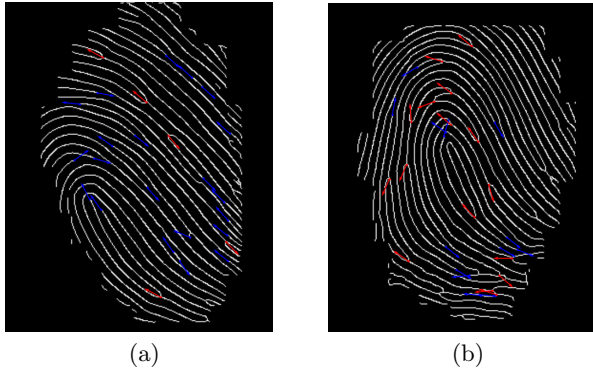


Fig. 5: Minutiae extraction

the pixel can then be classified according to the property of its CN value. If CN value equals 1 then it is an ending point, if CN value equals 3, then it is a bifurcation point.

B. Minutiae Validation

False minutiae may be introduced into the image due to factors such as noisy images, and image artifacts created by the thinning process. Hence, after the minutiae has been extracted, it is necessary to employ a validation process in order to validate the minutiae. We use the method proposed in [5] to validate the minutiae.

C. Experimental Results

After the minutiae has been extracted and validated, we can obtain the position and orientation of ending points and bifurcation points as shown in Fig. 5.

VI. MATCHING

REFERENCES

- [1] M. U. Akram, S. Nasir, A. Tariq, I. Zafar, and W. S. Khan, "Improved fingerprint image segmentation using new modified gradient based technique," in *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on*. IEEE, 2008, pp. 001 967–001 972.
- [2] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 777–789, 1998.
- [3] R. Thai, "Fingerprint image enhancement and minutiae extraction," *The University of Western Australia*, 2003.
- [4] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [5] M. Tico and P. Kuosmanen, "An algorithm for fingerprint image postprocessing," in *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on*, vol. 2. IEEE, 2000, pp. 1735–1739.