Extraction and Using Methods of Directional Information in fingerprint identification Preprocessing Algorithm

WeiAng Zhou
Police Physical Educational Training Department
Zhejiang Police Academy
Hangzhou, China
e-mail: 1579zhouwa@163.com

Ning Chang
Fundament Department
The Chinese People's Armed Police Force Academy
Langfang, China
e-mail: cncn133@163.com

Zhenhua Huang
Police Physical Educational Training Department
Zhejiang Police Academy
Hangzhou, China
e-mail: rhlucky@163.com

***Abstract*—Directional map represents the basic structural feature of fingerprint really and abstractly, so it has very important research value in Automated Fingerprint Identification system. The paper systematically introduced the calculation and smoothness of point directional map and block directional map, and how to realize oriented filtering to enhance image. In this paper, we used the predefine direction approach to compute point directional map, and a lowpass filter was applied to modifying the incorrect ridge directions. In order to improve the robust performance of the routine algorithm based on single-window, an improved method based on multi-window for computing block directional map was proposed. Considering that fingerprint is a kind of textured pattern, we used oriented filter based on block directional map to enhance images. To verify the performance of improved algorithm, contrastive experiment was conducted with routine algorithm. The results show that the set of algorithm is effective, especially to the fingerprint of poor quality.**

Keywords- fingerprint; directional map; oriented filtering

I. INTRODUCTION

The algorithm of AFIS (Automated Fingerprint Identification System) can be decomposed into the following four fundamental parts: (1)fingerprint image preprocessing, (2) fingerprint feature extraction, (3) fingerprint classification, (4) minutiae matching. The fingerprint image preprocessing is the first phase. Only after the preprocessing is performed perfectly, could the subsequent phases be reliable and satisfying.

The fingerprint is a kind of textured pattern[1], and the directional map of a fingerprint image represents an intrinsic nature of the fingerprint image and defines invariant coordinates for ridges and valleys around each local neighborhood, so it is crucial not only to the preprocessing phase but also to the subsequent phases. In the following sections we will describe in detail our extraction and using methods of fingerprint directional information in preprocessing phase.

II. CALCULATION OF POINT DIRECTIONAL MAP

The first step in this algorithm is to select a $w \times w$ rectangular window that takes $I(i, j)$ as the center (In factual performance, we should regulate the size of the window according to ridge width which is about 4 pixels in our fingerprint image database, and in our experiments, w equaling 9 is perfect. See Fig. 1.). Secondly, we sum the grey level values of the pixels marked i (i=1⋯8) to obtain the sums S_i. Finally, the ridge orientation is estimated at pixel (i, j) using the following equation[2].

7		6		5		4		3
8		7	6	5	4	3		2
		8				2		
1		1		I(i, j)		1		1
		2				8		
2		3	4	5	6	7		8
3		4		5		6		7

Figure 1. 9×9 rectangular window

$$D(i,j) = D(s_{\max}) \quad if \quad 4G(i,j) + s_{\max} + s_{\min} > \frac{3}{8}\sum_{i=1}^{8} s_i$$
$$else \quad D(i,j) = D(s_{\min}) \tag{1}$$

Where $I(i, j)$ represents the pixel at the ith row and the jth column of the input image, and $G(i, j)$ represents its gray level value. A directional map, D, is defined as a $M\times N$ matrix, where $D(i, j)$ represents the ridge orientation at pixel $I(i, j)$ and associates with a discrete ridge orientation l(l=1…8).

978-0-7695-3557-9/09 $25.00 © 2009 IEEE
DOI 10.1109/ETCS.2009.751

III. SMOOTHNESS OF POINT DIRECTIONAL MAP

Due to the presence of noise, smudges in the input image, the estimated ridge orientation may not always be correct. Since the ridge orientation varies slowly in a local neighborhood where no singular points appear, a lowpass filter can be used to modify the incorrect ridge orientation and smoothen the point directional map[3]. The lowpass filtering is performed in a 3×3 neighborhood, which is defined as follows:

$$\theta(i,j) = (D(i,j)-1)\times\frac{\pi}{8}$$

$$S_{\sin} = \sum_{u=i-1}^{i+1}\sum_{v=j-1}^{j+1}\sin 2\theta(u,v)$$

$$S_{\cos} = \sum_{u=i-1}^{i+1}\sum_{v=j-1}^{j+1}\cos 2\theta(u,v)$$

$$C(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{S_{\sin}}{S_{\cos}}\right)$$

$$if \quad C(i,j)<0 \quad C(i,j)+=\pi \tag{2}$$

where $C(i, j)$ is the modified pixel ridge orientation associated with a discrete ridge orientation $k\pi/8(k=0\cdots7)$.

IV. CALCULATION OF BLOCK DIRECTIONAL MAP

In a small local neighborhood, the pixels' orientations are uniform generally, so a local ridge orientation is usually specified for a region (block) rather than at every pixel.

The main steps of the routine method based on single-window are described as follows:

(1)An image is divided into a set of w×w non-overlapping blocks, where w is the block size. The size is decided by the ridge width, and the experiments with our fingerprint image database show that the size equal to 8 is excellent.

(2)After calculating the directional histogram of every block, we select the orientation to peak value as the corresponding block orientation.

The above describes the routine method to compute the block directional map, which is applied to oriented filtering described in section 5. Fig. 3(c) shows a thinned fingerprint binary image through an orientation filter, and Fig. 3(a) is the input image. The routine method isn't robust to translation and rotation. If we apply the same processing given above to the translated fingerprint image Fig. 3(b) (the result is shown in Fig. 3(d)), we'll find the difference between Fig. 3(c) and 3(d) in the marked positions with arrowheads where the ridge curvature changes acutely. The multi-windows method can be adopted to improve the algorithm.

The main steps of the improved method based on multi-window are described as follows:

(1) The input fingerprint image is divided into non-overlapping 4×4 blocks, as shown in Fig. 2. Every grid represents a 4×4 block whose block orientation depends on the four 8-connected neighborhoods around it. For example, if we want to calculate the orientation of the 4×4 black grid marked with $P_{11}P_{21}P_{22}P_{12}$, we have to compute the directional histograms of the four 8×8 grids $P_{00}P_{20}P_{22}P_{02}$, $P_{10}P_{30}P_{32}P_{12}$, $P_{01}P_{21}P_{23}P_{03}$, and $P_{11}P_{31}P_{33}P_{13}$. Suppose D_i (i=1 ⋯ 4) represents respectively the orientation to peak value of each of the four grid histograms, and N_i (i=1⋯4) represents the amount of pixels in the corresponding block along D_i orientation:

$$\begin{aligned} D_1 &= f(P_{00}P_{20}P_{22}P_{02}) \quad D_2 = f(P_{10}P_{30}P_{32}P_{12}) \\ D_3 &= f(P_{01}P_{21}P_{23}P_{03}) \quad D_4 = f(P_{11}P_{31}P_{33}P_{13}) \end{aligned} \tag{3}$$

$$N_i = sum(D_i) \qquad i=1,2,3,4 \tag{4}$$

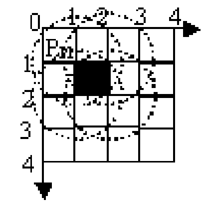


Figure 2. The diagram of the multi-windows method

where $f(A)$ means the orientation to peak value of the directional histogram of the block A, and sum (D_i) means the amount of pixels along D_i orientation in the corresponding block.

(2)The orientation D is selected as the block orientation of the 4×4 block $P_{11}P_{21}P_{22}P_{12}$, along which the amount of pixels in the corresponding 8×8 block is biggest among the four 8×8 blocks. The definition is given by:

$$D=j, \quad if \quad N_j = \max(N_i) \quad i,j=1,2,3,4 \tag{5}$$

(3)Take measures to reduce the workload of the new method. The routine algorithm spends more time than the improved method. If we want to obtain a pixel's block orientation, we only calculate a directional histogram of the 8×8 block around it in the traditional algorithm, but have to calculate the histograms of the four 8×8 blocks around it in the improved algorithm. So the workload of the latter is 4 times than the former.

The following method can be adopt to reduce the workload of the new method:

(1)Compute all of the histograms of the non-overlapping 4×4 block $P_{i\,j}P_{i+1\,j}P_{i+1\,j+1}P_{i\,j+1}$.

(2)Redivide the fingerprint image into 8 × 8 blocks. These blocks are overlapped each other, and the offset between the adjacent blocks is 4 pixels.

(3)When computing the histogram of the 8×8 block $P_{i\,j}P_{i+2\,j}P_{i+2\,j+2}P_{i\,j+2}$, we can sum directly the histograms of the inside four 4 × 4 blocks which have been previously computed.

(4)Translate horizontally in the length of four pixels and calculate the histogram of the 8×8 block $P_{i+1\ j}P_{i+3\ j}P_{i+3\ j+2}P_{i+1\ j+2}$.

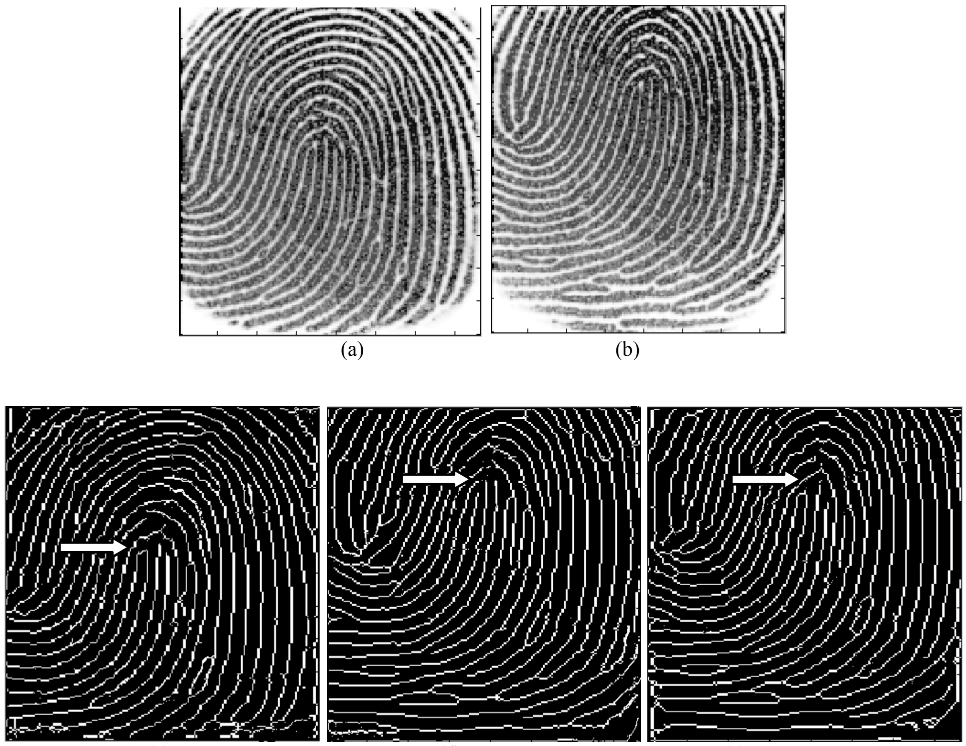
(5)Translate vertically and compute the histograms of $P_{i+1\ j+1}P_{i+3\ j+1}P_{i+3\ j+3}P_{i+1 j+3}$ and $P_{i+1\ j}P_{i+3\ j}P_{i+3\ j+2}P_{i+1\ j+2}$.

(6)The block orientation of the 4×4 block $P_{i+1\ j+1}P_{i+2\ j+1}P_{i+2\ j+2}P_{i+1\ j+2}$ is obtain as described as (3), (4), (5).

The result shows that the workload of the new method to compute the block orientation is no twice more than the traditional algorithm, but it is more robust to translation.

V. ORIENTED FILTERING

Due to the existence of noise, creases, corrupted ridge and valley structure, the out binary image often contains the spurious minutiaes. An enhancement algorithm which can remove the spurious minutiaes and improve the clarity of the ridge structure is necessary. Considering that the fingerprint is a kind of textured pattern, we use oriented filters[4] based on block directional map to enhance images. We designed 8 filter masks, $F_k(i,j)$, k=0,⋯7, each one associated with a discrete ridge orientation $k\pi/8$.



(a) (b)

(c) (d) (e)

Figure3. The contrast between the original image and the translated image with different methods

Now, applying the improved algorithm to computing the block directional map, and performing oriented filtering in the translated fingerprint image. The result is shown in Fig. 3(e), we'll find that the variance between Fig. 3(c) and Fig. 3(e) is negligible. It should be owed to the introducing of the 8-connected neighborhood. Let's see why the improved method can constrain the impact of the image translation. Suppose the displacement is L pixels in input image, which is divided by 4 in our improved algorithm, and the remainder is the final displacement. The value l varies among 1,2 and 3. if l is equal to 3, the anti-displacement is 1, so the maximal displacement is 2. The improved algorithm greatly constrains the impact of the image translation. The equation is given by:

$$l = \mathrm{mod}(L/4) \qquad (6)$$

The first step in the filter mask design is to decide the filter's size w according to ridge width. The experiments show that w equal to 7 is perfect.

In succession, the coefficients of horizontal mask is computed. The coefficient spatial arrangement of the horizontal mask is shown in Fig. 4. In fact, the filter is composed of the average filter and the detaching filter. The former is used to join the breaks in the ridge, and the latter is to eliminate the bridges between two parallel ridges. The coefficients of the average filter should meet the conditions given by:

$$u > x > y >= 0, z > 0 \qquad (7)$$

The coefficients of the detaching filter should meet the condition given by:

$$u + 2x + 2y - 2z = 0 \qquad \mathbf{(8)}$$

The coefficients in each row quickly attenuate from center to edge which can avoid corrupting the ridge whose curvature changes acutely.

-z/3	-2z/3	-z	-z	-z	-2z/3	-z/3
y/3	2y/3	y	y	y	2y/3	y/3
x/3	2x/3	x	x	x	2x/3	x/3
u/3	2u/3	u	u	u	2u/3	u/3
x/3	2x/3	x	x	x	2x/3	x/3
y/3	2y/3	y	y	y	2y/3	y/3
-z/3	-2z/3	-z	-z	-z	-2z/3	-z/3

Figure4. The coefficient spatial arrangement

To Generate the other 7 masks, we rotate the horizontal filter mask according to the following equation:

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} \quad \theta = k\pi/8(k = 0,1,2\cdots 7) \quad (9)$$

where (i, j) represents the coordinates in the horizontal mask, and (i', j') represents the ones in the rotated mask. But (i, j) is usually not an integer, so we needs to use nearest neighbor interpolation to get the coefficients of the rotated mask.

$g_\theta(i', j')$, the coefficients in the rotated mask, is equal to $g_1(i, j)$, the coefficients in the horizontal mask. But (i, j) is usually not an integer, so we needs to use nearest neighbor interpolation to get the coefficients of the rotated mask. suppose the four points (i_L, j_L), (i_L, j_U), (i_U, j_L), (i_U, j_U) compose a 1×1 square centered at pixel(i, j) ,and $g_1(i_L, j_L)$, $g_1(i_L, j_U)$, $g_1(i_U, j_L)$, $g_1(i_U, j_U)$ are their corresponding coefficients in the horizontal mask, where $i_L < i < i_U$, $j_L < j < j_U$.

First, make interpolation between (i_L, j_U) and (i_U, j_U):

$$g(i, j_U) = g_1(i_L, j_U) + (i - i_L)[g_1(i_U, j_U) - g_1(i_L, j_U)] \quad (10)$$

Then, make interpolation between (i_L, j_L) and (i_U, j_L):

$$g(i, j_L) = g_1(i_L, j_L) + (i - i_L)[g_1(i_U, j_L) - g_1(i_L, j_L)] \quad (11)$$

Finally, make interpolation between (i, j_U) and (i, j_L) :

$$g_\theta(i', j') = g(i, j_L) + (i - i_L)[g(i, j_U) - g(i, j_L)] \quad (12)$$

Now, based on(10), (11) and (12), we obtain:

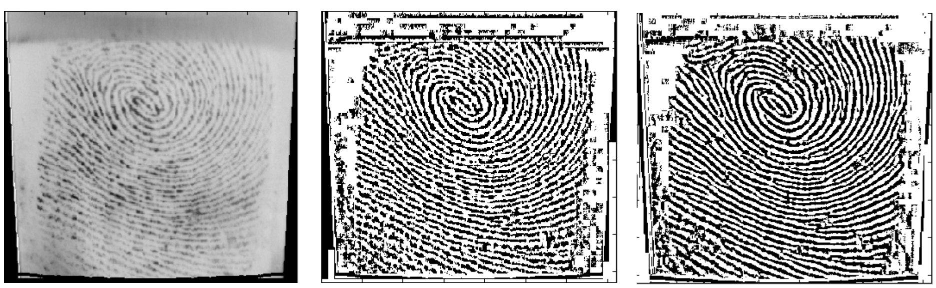
$$\begin{aligned} g_\theta(i', j') &= (j_U - j)(i_U - i)g_1(i_L, j_L) \\ &+ (i - i_L)(j_U - j)g_1(i_U, j_L) \\ &+ (j - j_L)(i_U - i)g_1(i_L, j_U) \\ &+ (j - j_L)(i - i_L)g_1(i_U, j_U) \end{aligned} \quad (13)$$

Once we have got all L filter masks, their coefficients can be hard-coded into the image enhancement algorithm and used without recalculation.

Now, aiming at every pixel $I(i, j)$ in the input image, we select a 7-connected neighborhood that takes $I(i, j)$ as the center. It is filtered with the mask that corresponds to the block orientation of the center $I(i, j)$. This adaptive filtering technique is given by:

$$G'(i, j) = \sum_{x=-7}^{7} \sum_{y=-7}^{7} G(i + x, j + y) g_\theta(i', j') \quad (14)$$

where $G'(i, j)$ represents the filtered grey level image, and $g_\theta(i', j')$ represents the coefficients in the rotated mask. Fig.5(c) shows the result of applying orientation filtering method to the input image in Fig. 5(a). The quality of Fig. 5(a) is very poor, in which there are many breaks and bridges. Fig.5(b) shows the binary image without orientation filtering. The binarization method[5] is given by LUO Xi-ping. Contrasting Fig.5(b) with 5(c), we'll find that the number of the spurious minutiaes is greatly reduced.



(a) (b) (c)

Figure 5. Contrast result with adopting oriented filtering

VI. CONCLUSION

The fingerprint is a kind of textured pattern, and the directional map of a fingerprint image represents an intrinsic nature of the fingerprint image. Aiming at the characteristic, we introduced directional information into the fingerprint image preprocessing phase in a simple but effective way. Experiments results demonstrate the effectiveness of the set of algorithm in this paper.

REFERENCES

[1] Naint K. Ratha, Kalle KARU, CHEN SHAO-YUN, et al. A real-time matching system for large fingerprint database[J]. *IEEE Trans on Pattern Analysis and Machine intelligence*, vol.18, pp. 799–812, 1996.

[2] Caudela G T, Grother P J. PCASYS-A Pattern-Level Classification Automation System for Fingerprints[R]. [s.l.]:*Technical Report NISTIR* 5647, 1995.

[3] LU Zhaoyang, YANG Deying, ZHANG Lei. Comparing of Several Algorithms of Fingerprint Directional Map[J]. *Microcomputer Development*, vol.13, pp. 24–26, 2003.

[4] Maio D,Maltoni.D.derect Gray –scale Minutiae Detection in fingerprints[J], *IEEE. Trans. PAMI*, vol.19, pp**.** 27–35, 1997.

[5] LUO Xi-ping, TIAN Jie. Image Enhancement and Minutia Matching Algorithms in Automated Fingerprint Identification System[J]. *Journal of Software*, vol.13, pp. 947–956*,* 2002.