ADAPTIVE BINARIZATION METHOD FOR FINGERPRINT IMAGES

Vesa Onnia[1], *Marius Tico*[2]

[1]Digital and Computer Systems Laboratory, Tampere University of Technology
[2]Nokia Research Center, Nokia
P.O.Box 553, Tampere, FINLAND
Tel: +358 3 365 3876; Fax: +358 3 365 3095
e-mail: onnia@tut.fi

ABSTRACT

An efficient method for binarization of the fingerprint images is presented in this paper. The algorithm is based on the fact that the fingerprint ridges are regions where the second directional derivative of the image is positive. The derivatives at each pixel are approximated using a facet model based on the intensity values of pixels in a certain neighborhood. It was noticed that the size of this neighborhood affects critically to the results. The size of the neighborhood is depended on the inter-ridge distance. The method based on the averaging in the direction of the ridges was used to determine inter-ridge distances. Using these inter-ridge distances, size of the neighborhood in the binarization algorithm was calculated for each pixel. Ridge directions were calculated using the gradient information of the image. The algorithm was tested using digitally acquired fingerprints. The results show that the algorithm is capable to produce very accurate binarization results.

1. INTRODUCTION

Several factors like the presence of scars, variations of the pressure between the finger and acquisition sensor, worn artifacts, etc., can affect the quality of the acquired fingerprint image. The main goals of the binarization methods are to obtain ridge information from the fingerprint image and reduce the noise in the image.

Binary fingerprint images are needed for reliable minutia point extraction, which is an important method used in person identification applications [1, 2]. It is based on small details detected in the fingerprint images. The most common detected minutia points are ridge endings and bifurcations.

It is quite difficult to model the noise present in the fingerprint images, but the signal is fairly simple to characterize due to the flow of the fingerprint ridges whose orientations are slowly changing in the fingerprint pattern [3]. Taking for instance a small local area in the fingerprint image we may note that the ridge orientation as well as the ridge period are maintained almost constant inside this area, whereas the noise present in the image does not exhibit such regularities. The majority of the fingerprint enhancement methods proposed in the literature are based on this observation [4-8].

In this paper the efficient method for the binarization is presented. This method is based on the facet model, which approximates the directional derivatives at each pixel. The length of the tap-filters and orientation angles, used in binarization, are calculated adaptively based on inter-ridge distances and image gradients.

2. ORIENTATION FIELD CALCULATION

First we apply simple histogram equalization to the whole image before any other operations to enhance the ridge information [9]. Method used for the orientation field calculation is based on the method presented earlier in the paper [6]. This method uses the gradient information for the determination of the ridge orientations. The gradient of a point in the gray scale image is oriented along the maximum gray level change. One may expect that the ridge orientation is orthogonal to the gradient value. Because of the ridge irregularities and noise one has to blur the orientations using a low-pass filter. After blurring the orientation angles are calculated using arcus tangent. The algorithm has the following steps.

1) Compute the gradients $\partial_x(i,j)$ and $\partial_y(i,j)$ at each pixel (i,j). Any appropriate method can be used, depending on the computational requirements. We used the method presented in [10].
2) Calculate the gradient angle using arcus tangent and estimate the orientation angle by rotating all gradient angles by $\pi/2$. Because of an ambiguity of π in fingerprint orientation, multiply all angles by two (this is done because the blurring process in the step 4).
3) Return to the vector form representation.

0-7803-7402-9/02/$17.00 ©2002 IEEE

Procedures 1-3 can be expressed as follows:

$$X(i,j)=\cos\left(2\left(\frac{\pi}{2}-\operatorname{atan}\left(\frac{\partial_y(i,j)}{\partial_x(i,j)}\right)\right)\right) \quad (1)$$

$$Y(i,j)=\sin\left(2\left(\frac{\pi}{2}-\operatorname{atan}\left(\frac{\partial_y(i,j)}{\partial_x(i,j)}\right)\right)\right) \quad (2)$$

$X(i, j)$ and $Y(i, j)$ denote the doubled orientations in the vector form. The formula (1) and (2) can be simplified using elementary algebra into the following forms.

$$X(i,j)=\frac{\partial_y^2(i,j)-\partial_x^2(i,j)}{\partial_y^2(i,j)+\partial_x^2(i,j)} \quad (3)$$

$$Y(i,j)=\frac{2\partial_y(i,j)\partial_x(i,j)}{\partial_y^2(i,j)+\partial_x^2(i,j)} \quad (4)$$

4) Obtain the smooth orientation fields X' and Y' from X and Y, respectively, by blurring both matrices with a Gaussian low-pass filter. The size and variance of the filter in our case were 25×25 and 4, respectively.
5) Compute the orientation angles using arcus tangent.

$$\theta(i,j)=\frac{1}{2}\operatorname{atan}\left(\frac{\partial Y'(i,j)}{\partial X'(i,j)}\right) \quad (5)$$

3. INTER-RIDGE DISTANCE CALCULATION

The inter-ridge distance calculation method used in this paper is based on the method presented in the paper [6]. This method uses the observation that the pixels located on ridges usually exhibit lower gray level values than the pixels located on the valleys. By averaging pixels in a local neighborhood, where no minutiae and singular points appear, along the ridges and valleys, a projection waveform from the ridges can be obtained (see fig. 1).

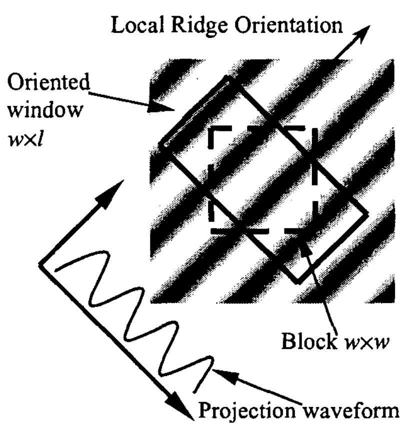

Figure 1. Oriented window and projection waveform [6].

By taking an average from the distances between adjacent maximum points of the projection waveform, an average inter-ridge distance inside certain block can be calculated. The algorithm is as follows.

1) Filter the image I with $m\times m$ mean filter. We used 7×7 window. This operation is applied because otherwise the pores detected on the ridges disturb the inter-ridge distance calculation.
2) Divide image I into blocks of size $w\times w$ (w=16).
3) For each block centered at pixel (i,j), compute the projection waveform, $Z[0]$, $Z[1]$, ..., $Z[l-1]$, of the ridges and valleys within $w\times l$ window oriented in the direction orthogonal to the ridge orientation, where

$$Z[k]=\frac{1}{w}\sum_{d=0}^{w-1} I(u,v), \qquad k=0,1,\ldots,l-1 \quad (6)$$

$$u=i+\left(d-\frac{w}{2}\right)\cos\theta(i,j)+\left(k-\frac{l}{2}\right)\sin\theta(i,j) \quad (7)$$

$$v=j+\left(d-\frac{w}{2}\right)\sin\theta(i,j)+\left(\frac{l}{2}-k\right)\cos\theta(i,j) \quad (8)$$

We used 16×32 windows. $\theta(i,j)$ is the center value of the block from the orientation field matrix (5). (See figure 1.)
4) Find the maximum points from the projection waveform.
5) Calculate average distance between adjacent maximum points in the projection waveform. If there is no acceptable maximum points, assign value -1 to that block. If calculated inter-ridge distance value is not in the certain range, assign value -1 to that block. This range is depended on a fingerprint scanner's resolution. We used 500 dpi images. The range is [1/3, 1/25] for them [6].
6) Interpolate the values for each block assigned with the value -1. We used the following method, where $D(i,j)$ denotes the calculated inter-ridge distance value for the block centered at (i,j) and D' denotes the interpolated inter-ridge distance field.
(i) For each block centered at (i, j)

$$D'(i,j)=\begin{cases} D(i,j) & \text{if } D(i,j)\neq -1 \\ \dfrac{\sum_{u=i-1}^{i+1}\sum_{v=j-1}^{j+1} D(u,v)\delta(D(u,v))}{\sum_{u=i-1}^{i+1}\sum_{v=j-1}^{j+1}\delta(D(u,v))} & \text{otherwise} \end{cases}$$

$$\text{where } \delta(x)=\begin{cases}0 & \text{if } x\leq 0\\ 1 & \text{otherwise}\end{cases} \quad (9)$$

(ii) If there exists at least one block with the inter-ridge distance value of -1, swap D and D' and go to step (i).
7) Remove the outliers from the interpolated inter-ridge distance field by applying a median filter. We used 3×3 window.

8) Calculate the final inter-ridge distance field F by interpolating the obtained distance field to the size of the original image. We used bi-cubic interpolation included in MATLAB 5.3.

4. BINARIZATION

The fingerprint ridges are detected using the approach introduced in [3,11]. This approach relies on the observation that fingerprint ridges can be identified as image zones where the second directional derivative of the discrete gray-level intensity surface along the orthogonal direction to the local ridge orientation is positive.

Let $I(i,j)$ denote the value of the gray-level intensity at the pixel (i,j) where i denotes the horizontal coordinate that increases from left to right, and j denotes the vertical coordinate that increases from bottom to top. Also, let $\theta(i,j)\in[0,\pi)$ denote the ridge orientation at the pixel (i,j). The second derivative of the intensity surface along the direction $\mathrm{v}=[-\sin\theta(i,j)\ \cos\theta(i,j)]^{\mathrm{T}}$ is the following.

$$I''(i,j)=I^{(2,0)}(i,j)\sin^2\theta(i,j)+I^{(0,2)}(i,j)\cos^2\theta(i,j)-I^{(1,1)}(i,j)\sin 2\theta(i,j) \quad (10)$$

$I^{(p,q)}(i,j)$ denotes the $(p+q)^{\mathrm{th}}$ partial derivative of the intensity surface at the location (i,j), p along the horizontal axis and q along the vertical axis.

The partial derivatives can be approximated using a facet model based on discrete Chebyshev polynomials up to the third order [12]. A continuous surface $z=f(x,y)$ is fitted over the intensity values of image pixels located in a $(2L+1)\times(2L+1)$ window centered at (i,j). The partial derivatives at (i,j) are approximated by the corresponding partial derivatives of the z at (0,0).

$$I^{(p,q)}(i,j)\approx\left.\frac{\partial^{p+q}f(x,y)}{\partial x^p\partial y^p}\right|_{x=0,y=0} \quad (11)$$

The following parametric form is assumed for the continuous surface.

$$f(x,y)=\sum_{m=0}^{T}\sum_{n=0}^{T}g_{m,n}h_m(x)h_n(y), \quad (12)$$

where $\{g_{m,n};\ 0\le m,n\le T\}$ and $\{h_m(\cdot),\ h_n(\cdot);\ 0\le m,n\le T\}$ denote the set of unknown parameters and $T+1$ one-dimensional real functions defined on $[-L,L]$, respectively. Using a least square estimator of the unknown parameters, the partial derivative (11) can be approximated by the following equation [3,11].

$$I^{(p,q)}(i,j)\approx\sum_{c=-L}^{L}f_{p,L}(c)\left(\sum_{r=-L}^{L}I(i-c,j-r)f_{q,L}(r)\right) \quad (13)$$

where $f_{p,L}$ and $f_{q,L}$ are the impulse responses of one-dimensional finite impulse response filters of length $2L+1$, whose coefficients are given by

$$f_{0,L}(l)=\frac{3(3L^2+3L-1-5l^2)}{(2L+1)(2L-1)(2L+3)},$$

$$f_{1,L}(l)=\frac{-5l(15L^4+30L^3-15L+5-21L^2l^2-21Ll^2+7l^2)}{L(L+1)(2L+1)(L-1)(2L-1)(L+2)(2L+3)},$$

$$f_{2,L}(l)=\frac{30(3l^2-L^2-L)}{L(L+1)(2L+1)(2L-1)(2L+3)}, \quad (14)$$

for all $-L\le l\le L$.

The choice of the parameter L, which controls the length of the filters (14), is critical. If L is too small the continuous surface z may overfit the observed data being unable to distinguish and cancel the noise. If L is too large z may be too simple for fitting the fingerprint ridges included in the approximation window, and hence it may alter the signal. In contrast with the methods in [3,11], the parameter L is calculated separately for each image pixel using the interpolated inter-ridge distance field F (see section 3). At each pixel (i,j) we calculated the value of L by multiplying the inter-ridge distance $F(i,j)$ using experimentally found coefficient. In our case this coefficient was 0.5. The block representation of the binarization algorithm is in figure 2.

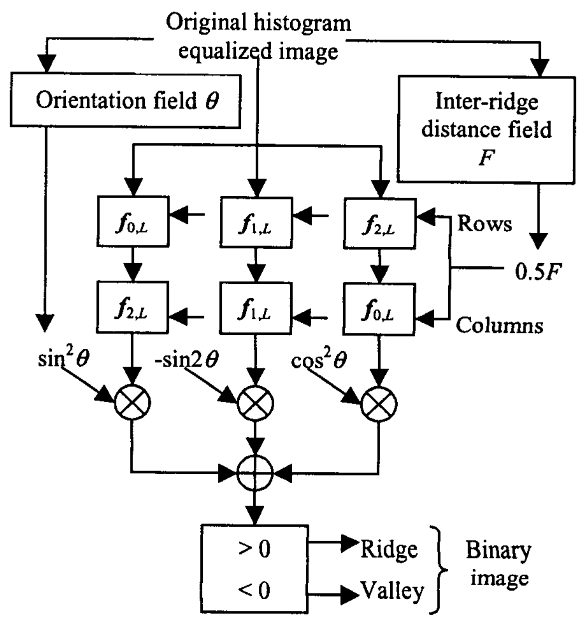

Figure 2. Diagram of the fingerprint binarization algortihm.

First the orientation and the inter-ridge distance values are calculated for each pixel using the methods presented in sections 2 and 3, respectively. Second step is to filter the fingerprint image along the rows, and after it along the columns. Filters used are presented in (14), where parameter L is tuned using the inter-ridge distance field F. Pixels in obtained three filtered images are weighted using the orientation field and the results are summed together. The final step is to obtain a binary ridge map image based on the sign of each pixel in the enhanced image.

4. EXPERIMENTAL RESULTS

We used the database provided by the reference [13] for tests. The database consists of four different sub-databases. We chose DB2 from the four alternatives, because the acquiring method for this database was low-cost capacitive sensor. Images have been scanned using 500dpi resolution, the image size is 256×364 and color depth is 8 bits. We applied our binarization algorithm for the 96×128 images cropped from the original images. One can notice from the figure 3 that using our algorithm very accurate binarization results can be obtained. Pores do not affect to the result and the algorithm is capable to adapt varying inter-ridge distances.

5. CONCLUSIONS

The binarization is an important task in person verification and recognition systems based on fingerprints. Using binarization the noise can be reduced and the ridge information can be enhanced.

In this paper the fingerprint binarization method is presented. Our binarization method uses the information of the inter-ridge distances and orientations to obtain the adaptive image dependent binarization. This method was tested using real-world fingerprints. The results show that the algorithm is capable to perform very accurate binarization.

6. REFERENCES

[1] A.K. Jain, R. Bolle, S. Pankanti, *Biometrics - Personal Identification in Networked Society*, Kluwer Academic Publishers, United States, 1999.

[2] D.D. Zhang, A. Vailaya, *Automated Biometrics Technologies and Systems*, Kluwer Academic Publishers, United States, 2000.

[3] M. Tico, *On Design and Implementation of Fingerprint-Based Biometric Systems*, PhD thesis, Tampere University of Technology, Tampere, Finland, 2001.

[4] N.K. Ratha, S. Chen, A.K. Jain, "Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images", *Pattern Recognition,* 28 (11), 1995, pp.1657-1672.

[5] A.K. Jain, L. Hong, S. Pankanti, R. Bolle "An Identity Authentication System Using Fingerprints", *Proceedings of the IEEE*, 85 (9), 1997, pp. 1365-1388.

[6] L. Hong, Y. Wan, A.K. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (8), 1998, pp. 777-789.

[7] B.G. Sherlock, D.M. Monro, K. Millard, "Fingerprint Enhancement by Directional Fourier Filtering", *IEE Proc.-Vis. Image Signal Processing*, 141 (2), pp. 87-94, 1994.

[8] A.J. Willis, L. Myers, "A Cost-Effective Fingerprint Recognition System for Use with Low-Quality Prints and Damaged Fingerprints", *Pattern Recognition*, 34 (2), pp. 255-270, 2001.

[9] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, United States, 1989.

[10] D. Maio, D. Maltoni, "Ridge-line Density Estimation in Digital Images", *Proceedings of 14th IEEE International Conference on Pattern Recognition*, pp. 534-538, vol. 1, 16-20 August, 1998.

[11] M. Tico, V. Onnia, P. Kuosmanen, "Fingerprint Image Enhancement based on Second Directional Derivative of the Digital Image", submitted to *EURASIP Journal on Applied Signal Processing*, 2001.

[12] R.M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 58-68, 6 (1), 1984.

[13] D. Maio, D. Maltoni, R. Capelli, J.L. Wayman, A.K. Jain, FVC2000: Fingerprint Verification Competition.

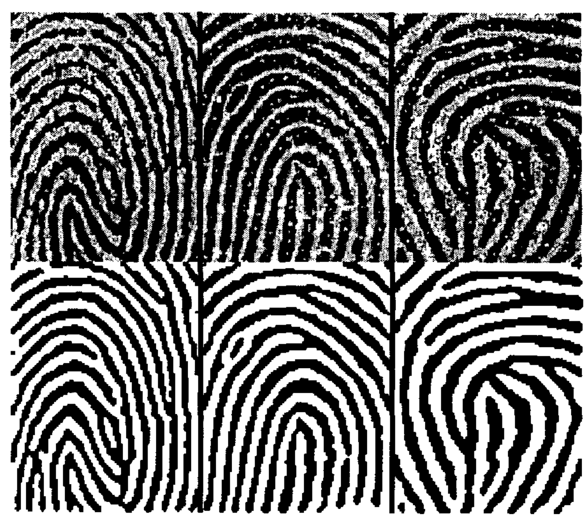

Figure 3. Binarization result for three fingers. Original fingers are on the top and binarized are beneath them.