Advanced Materials Research *Online: 2011-11-29*
ISSN: 1662-8985, Vols. 403-408, pp 4499-4506
doi:10.4028/www.scientific.net/AMR.403-408.4499

Fingerprint Singular Point Detection Using Orientation Field Reliability

Ravinder Kumar[1], Pravin Chandra[2] and M. Hanmandlu[3]

[1]Deptt. of Computer Science and IT Ansal Institute of Technology Gurgaon, India

[2]Institute of Informatics and Communication South Campus, University of Delhi New Delhi, India

[3]Department of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi -110016, India.

ravinder.kumar@aitgurgaon.org ,chandra.pravin@gmail.com, mhmandlu@ee.iitd.ac.in

Keywords: Orientation field, Orientation field reliability, fingerprint enhancement, singular points, thinning.

Abstract. Singular point detection is the most important step in Automatic Fingerprint Identification System (AFIS) and is used in fingerprint alignment, fingerprint matching, and particularly in classification. The computation of orientation field of a fingerprint can be verified by computing orientation field reliability. The most unreliable portion in orientation field can be the possible location of singular points. In this paper we have proposed a novel algorithm for detecting singular points using reliability of the fingerprint orientation field. Experimental results show that the proposed algorithm accurately detects singular points (core and delta) with the detection rate of 92.6 %.

Introduction

Fingerprint is a two dimensional (2D) well defined directed pattern of ridge-valley, captured by various sensors like capacitive, optical etc. In each fingerprint, there are usually two kinds of singular points: core and deltas, where the direction of the ridges changes abruptly [1]. These singular points can be used for classification of fingerprints [2, 3].

Methods for locating singular points were proposed by Kawagoe and Tojo in 1984 [4], due to its robustness to the image rotation and computational simplicity. Poincare - index based methods [1, 7] usually suffer from spurious singular point detection, particularly in low quality images. The interesting implementation of the Poincare method for locating singular points was proposed by Bazen and Gerez (2002) [5]. The authors compute the rotation of the orientation image and then perform a local integration in a small neighborhood of each element. The authors also provided a method for associating an orientation with each singularity; this is done by comparing the orientation image around each detected singular point with the orientation image of an ideal singularity of the same type. Jain, Prabhakar, Hong and Pankanti [6] proposed geometry of region (GR) technique to locate core point, which is then combined with a technique called the direction of curvature (DC).

Zhou, Chen and Gu in 2009 [7], proposed a novel algorithm for detecting singular points. Differences of the ORIentation values along a Circle (DORIC) feature is used to remove spurious singular points, after an initial detection using the conventional poincare index method.

This paper proposes a novel approach for detecting the singular point core and delta by computing the orientation reliability field of the enhanced image. The main steps of the proposed algorithm include fingerprint enhancement, computing orientation field reliability, threshold, thinning, and finally obtaining the single x and y coordinates of each singular point by applying morphological open and close operations.

The paper is organized in the following sections. Section II describes the proposed algorithm. Section III discusses the experimental results. Section IV concludes the proposed algorithm.

Proposed Algorithm

The flowchart of the singular point detection algorithm using proposed orientation field reliability is shown in Figure 1. The main steps of the algorithm include:

A. Fingerprint Enhancement
B. Compute Orientation field Reliability
C. Singular Point Detection
   Step 1. Thresholding
   Step 2. Thinning
   Step 3. Morphological Operation

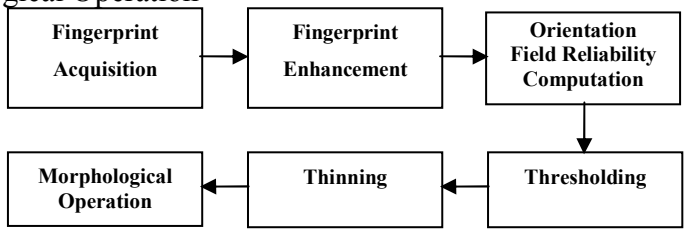

Figure 1. Flowchart of the proposed algorithm

Fingerprint Enhancement. The fingerprint enhancement includes the following steps and are shown in figure 2.

1. Normalization
2. Ridge Segmentation
3. Computing Ridge Orientation
4. Ridge Frequency Estimation
5. Ridge Filtering

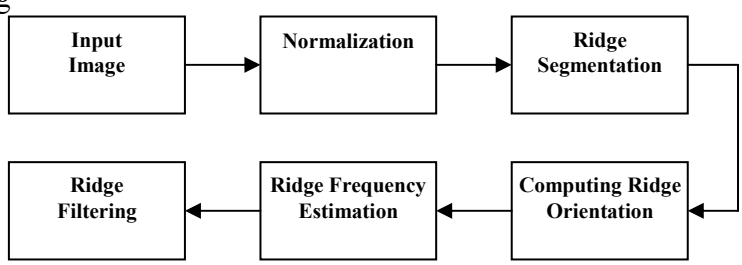

Figure 2. Flowchart for fingerprint enhancement

1) Normalization

Let $I(i,j)$ denote the gray-level value at pixel (i,j), M_I and V_I denote the estimated mean and variance of image I, respectively, and $N(i,j)$ denote the normalized gray-level value at the pixel (i,j). The normalized image is then defined as follows [8]:

$$N(i,j)=\begin{cases} M_0 + \sqrt{\dfrac{V_0\left(I(i,j)-M_I\right)^2}{V_I}} & \text{if } I(i,j) > M_I \\ M_0 - \sqrt{\dfrac{V_0\left(I(i,j)-M_I\right)^2}{V_I}} & \text{otherwise} \end{cases} \quad (1)$$

Where M_0 and V_0 are the desired mean and variance respectively. The mean and variance of a gray-level fingerprint image with the of size $M \times N$ pixels, are computed as follows:

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

and

$$V_I = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(I(i,j)-M_I\right)^2 \quad (3)$$

Where $I(i,j)$ represents the intensity of the pixel at ith row and jth column. The main objective of the normalization process is to reduce the variation of gray-level values along the ridges and valleys. This operation does not change the clarity of the ridges and valleys, because this is pixel-wise operation. The normalized image at desired mean and variance is shown in figure 3.

Figure 3. Normalized image

2) Ridge Segmentation

The main objective of ridge segmentation is to extract ridges from the background. We have used mean and variance based segmentation method [1] for ridge segmentation from the background. The steps for mean and variance based segmentation method are summarized as follows:

- Divide the input image I into non-overlapping blocks of size $w \times w$.
- Compute the mean value of each block using following equation.

$$M(I) = \frac{1}{w^2} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} I(i, j) \tag{4}$$

- Compute the standard deviation *std(I)* by using mean value computed in above step as follows:

$$std(I) = \sqrt{\frac{1}{w^2} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} (I(i, j) - M(i, j))^2} \tag{5}$$

- Empirically select a threshold value. If $std(I) > threshold$ the block is considered as ridge region, otherwise the block belongs to the background.

The ridge-segmented image is shown in figure 4.

Figure 4. Ridge-segmented fingerprint image

3) Computing Ridge Orientation

The least mean square orientation estimation method developed by Hong, Wan, and Jain [8] has been used for computing ridge orientation. Orientation field image is shown in Figure 5.

Figure 5. Orientation field image

4) Computing Ridge Frequency

Local ridge frequency of the orientation image is computed in this step using the method proposed by Hong, Wan, and Jain [8]. The estimated ridge frequency image is shown in figure 6.

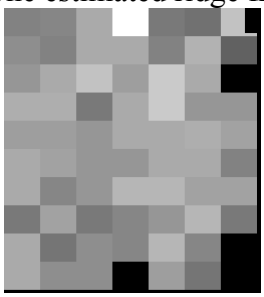

Figure 6. Estimated ridge frequency image

5) Ridge Filtering

As proposed in [10], [11] Gabor filters have both frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains. In this paper, we have used Gabor filters as bandpass filter to remove the noise and preserve true ridge valley structure [8]. The ridge-filtered image is shown in figure 7.

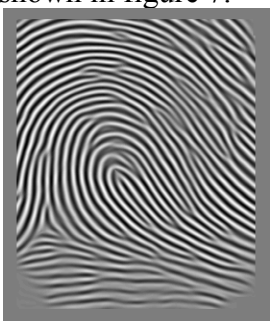

Figure 7. Ridge filtered image

Computing orientation field Reliability. Orientation field is an important parameter in Automated Fingerprint Identification Systems (AFIS), which influences the image segmentation, filtering enhancement and matching of fingerprint images.

For orientation computation the algorithm described in [8] is used with slight modification. The gaussian operator (6) has been used to compute the G_x and G_y, the gradient magnitude in x and y directions, respectively

$$h_g(x,y) = e^{-\left(x^2+y^2\right)/\left(2\sigma^2\right)} \tag{6}$$

The computation of orientation field is elaborated in the following steps:

The image is divided into non-overlapping block of size $W \times W$, and a single orientation is assigned to the block that corresponds to the most apparent or dominant orientation of the block. In the proposed method, W is set to 16.

The horizontal and vertical gradients $G_x(x,y)$ and $G_y(x,y)$ at each pixel (x,y) respectively is computed using gaussian mask. The size of the mask is set to 7×7.

Compute the ridge orientation of each pixel (x,y) by averaging the squared gradients within a $W \times W$ window centered at (x,y) as follows:

$$G_{xx} = \sum_{(x,y)\in W} G_x^2(x,y) \tag{7}$$

$$G_{yy} = \sum_{(x,y)\in W} G_y^2(x,y) \tag{8}$$

$$G_{xy} = \sum_{(x,y)\in W} G_x(x,y).G_y(x,y) \tag{9}$$

$$\theta(x,y) = \frac{1}{2}\tan^{-1}\left(\frac{2G_{xy}}{G_{xx}-G_{xy}}\right) \tag{10}$$

Because of noise, corrupted ridge, valley structures and low gray value contrast, a low-pass filter can be used to adjust the erroneous local ridge orientation. However, to perform the low-pass filtering, the orientation image needs to be converted into a continuous vector field as follows [14]:

$$\boldsymbol{\phi_x} = \cos(2\boldsymbol{\theta}(\boldsymbol{x},\boldsymbol{y})), \tag{11}$$

and

$$\boldsymbol{\phi_y} = \sin(2\boldsymbol{\theta}(\boldsymbol{x},\boldsymbol{y})), \tag{12}$$

Where ϕ_x and ϕ_y are the x and y components of the vector field, respectively. With the resulting vector field, the Gaussian low-pass filter can be applied as follows:

$$\phi_x'(x,y) = \sum_{u=-l}^{l}\sum_{v=-l}^{l} w(u,v)\phi_x(x-uw, y-vw), \tag{13}$$

$$\phi_y'(x,y) = \sum_{u=-l}^{l}\sum_{v=-l}^{l} w(u,v)\phi_y(x-uw, y-vw), \tag{14}$$

Where w is a two-dimensional low-pass filter with unit integral. Orientation field image is shown in Figure 8.

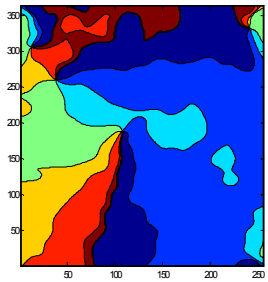

Figure 8. Surface view of Orientation image

Reliability of the orientation field at any pixel (x,y) is computed as follows:

$$\min\_inertia(x,y) = \left((G_{yy}+G_{xx}) - (\phi_x' G_{xx} - G_{yy}) - (\phi_y' G_{xy})\right)/2 \tag{15}$$

$$\max\_inertia(x,y) = G_{yy} + G_{xx} - \min\_inertia(x,y) \tag{16}$$

We calculate the area moment of inertia about the orientation axis (this will be the minimum inertia) and an axis perpendicular (which will be the maximum inertia). Then reliability measure is given by

$$reliability(x,y) = 1 - \frac{\min\_inertia(x,y)}{\max\_inertia(x,y)} \quad (17)$$

The reasoning being that if the ratio of the minimum to maximum inertia is close to one we have little orientation information. Figure 9 shows the orientation field reliability map with black region indicating the less reliable region (possible location of singular points). 3-D Mesh Plot with Contour of orientation field reliability map and its peak(s) indicating the location of singular point is shown in figure 10.

Figure 9. Orientation field reliability map

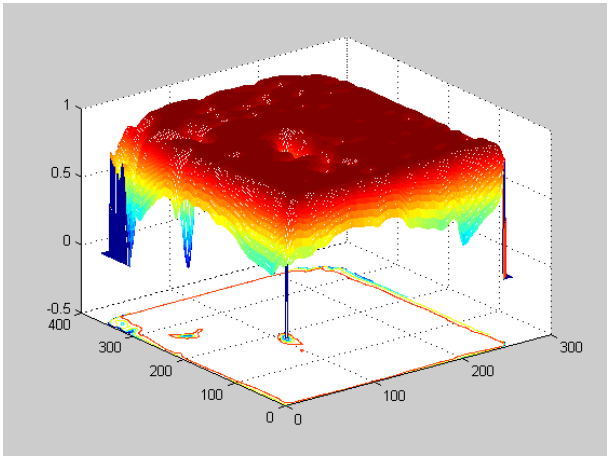

Figure 10. 3-D Mesh Plot with Contour of orientation field reliability map indicating the location of singular point

Singular point Detection. The orientation field reliability image computed in section B is used to locate the coordinate of singular points by applying the following three steps:

Step 1. Thresholding

In this step, we have to locate the candidate area for singular point by finding the area of least reliability. Thresholding will be used to identify such area. In this paper, we apply, a threshold $0.5 > T < 0.1$ to the orientation field reliability image.

Step 2. Thinning

The singular point area obtained in step 1 may still have undesirable effects due to noise resulting in more than one pixel for the singular point. To reduce the singular point contour to the one pixel, we have applied MATLAB thinning function to the threshold image. The resulting image has been shown in figure 11 indicating location of singular points by white dots.

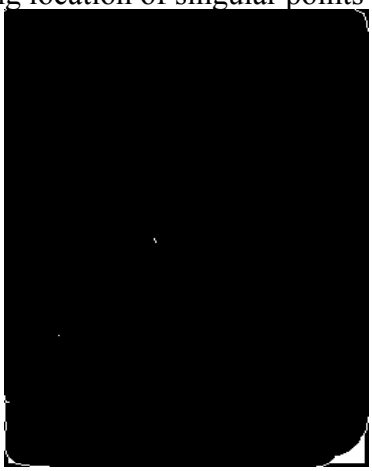

Figure 11. White dots indicate the location of Singular points

Step 3. Morphological Operation

The image obtained after applying thinning in Step 2 has more than one pixel for the singular point. In order to compute the single x and y coordinate corresponding to each singular point, we have used morphological open and close operations.

Experimemtal results and discussion

The proposed algorithm for fingerprint singular point detection has been evaluated on databases FVC2002 (set B). It contains four different databases (DB1_B, DB2_B, DB3_B and DB4_B), each database is 10 finger wide (w) and 8 impressions per finger (d) (80 fingerprints in all) and are described in table 1[12].

TABLE I. DESCRIPTION OF DATABASE FVC2002 (SET B)

	Sensor Type	**Image Size**	**(w × d)**	**Resolution**
DB1	Optical Sensor	388 × 374	10 × 8	500 dpi
DB2	Optical Sensor	296 × 560	10 × 8	569 dpi
DB3	Capacitive Sensor	300 × 300	10 × 8	500 dpi
DB4	SfinGe v2.51	288 × 384	10 × 8	about 500 dpi

The experiment was conducted on set B of FVC2002 database and the performance was evaluated by the following three matrices defined as follows:

- Detection Rate – defined as the ratio of truly detected singular point in a fingerprint image to the total number of fingerprint images being evaluated.
- %age of False SP – defined as the ratio of spurious detection of singular point to the total number of fingerprint images being evaluated.
- %age of Missed SP – defined as the ratio of missed singular point to the total number of fingerprint images being evaluated.

Here detection means all singular points detection. The result of the experiment is presented in table II.

TABLE II. SUMMARY OF THE DETECTION RATE

Database	**Detection Rate**	**%age of False SP**	**%age of Missed SP**
DB1	87.5	15.0	12.5
DB2	93.4	9.3	6.6
DB3	94.7	3.5	5.3
DB4	95.9	4.1	4.1
Total	**92.6**	**8.4**	**7.4**

Conclusions

This paper presented a new algorithm to precisely locate the singular points (core and delta) in fingerprint images. The proposed algorithm has two fundamental parts 1) Fingerprint image enhanced based on the algorithm presented in Ref [8]. 2) Computation of orientation field reliability, followed by detection of singular points. The result for the proposed algorithm is detection rate of 92.6%, false singular point detection rate of 8.4%, and missed singular point rate of 7.4%. This algorithm also locates a secondary core and delta if it exists. Our future work will focus on improvements in locating the particular type of singular point of fingerprint images.

References

[1] D. Maltoni, D. Maio, A. K. Jain and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer-Verlag, June 2009.

[2] K. Karu and A. K. Jain, " Fingerprint Classification,"Pattern Recognition, vol. 17, no. 3, pp. 389-404, 1996.

[3] Q. Zhang and H. Yan, "Fingerprint Classification Based on Extraction and Analysis of Singurities and Pseudo Ridges," Pattern Recognition, vol. 37, no. 11, pp. 2233-2243, 2004.

[4] M. Kawagoe, and A. Tojo, "Fingerprint pattern classification", Pattern Recognition, Volume 17, Issue 3, pp. 295-303, 1984

[5] A. M. Bazen and S. H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints,"IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 24, no. 7, pp. 905-919, 2002.

[6] A. K. Jain, S. Prabhakar, L. Hong , and S. Pankanti, " Filterbank – based fingerprint matching," IEEE Transactions on Image Processing, vol. 9, pp. 846 – 859, 2000.

[7] Jie Zhou, Franglin Chen, and Jinwei Gu, "A Novel Algorithm for Detecting Singular Points from Fingerprint Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, No. 7, July 2009.

[8] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, No. 7, July 2009.

[9] Sen Wang and Yangsheng Wang., " Fingerprint Enhancement in the Singular Point Area," IEEE signal Processing letters, vol. 11, no. 1, pp. 16-19, January 2004.

[10] A. Jain, L. Hong, and R. Bolle , " On-Line Fingerprint Verification," IEEE Trans.Pattern Analysis and Machine Intelligence, vol. 19,no. 4, pp. 302-314, 1997

[11] A. K. Jain, F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," Pattern Recognition, vol. 2, no. 12, pp. 1,167-1,186, 1991.

[12] Fingerprint Verification Competition (FVC), "http://bias.csr.unibo.it/fvc2002/," .

[13] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed., Prentice Hall, Upper Saddle River, NJ.

[14] A. R. Rao, "A taxonomy for texture Description and Identification"Springer, New York, 1990.