



# Minutiae filtering using ridge-valley method

Marco A. Ameller F.<sup>a</sup> and Angélica González<sup>b</sup>

<sup>a</sup> Computers Engineering Department, Autonomous University Tomas Frias, Potosí, Bolivia

<sup>b</sup> Computers and Automation Department, University of Salamanca, Salamanca, Spain

{ameller, angelica}@usal.es

## KEYWORD

*Minutiae filtering;  
inverse fingerprint;  
union - bifurcation filter*

## ABSTRACT

*In order to identify subjects in a convenient and efficient way one must use some special feature that makes it possible to discriminate between persons. Some of the features are biometric in nature, such as iris texture, hand shape, the human face, and of course finger prints. These play an important role in many automatic identification systems, since every person is believed to have a unique set of fingerprints. Before a fingerprint image can be looked up in a database, it has to be classified into one of 5 types in order to reduce processing times.*

## 1. Introduction

A AFIS based on minutiae makes several processes in a fingerprint, one of them is the extraction of characteristics, in our case the minutiae extraction, minutiae filtering is very useful in removing the spurious minutiae detected in corrupt zones of the fingerprint, a free false minutiae set offers less comparisons and a better result in minutiae matching, in this article we will see a detailed method to minutiae filtering using the inverse of the fingerprint, a similar method is in (Hung, 1993), this method works with two types of minutiae endings and bifurcations these common types are shown in figure 1, filter details will be discussed in the next sections.

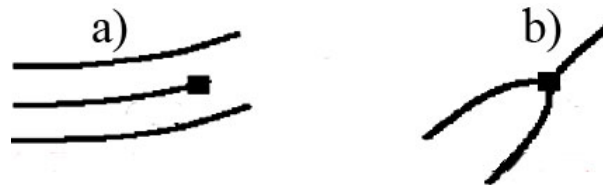


Figure 1: Common minutiae types, a) Ending and b) Bifurcation.

## 2. Minutiae extraction main process description

In order to extract a minutiae set of a fingerprint it is necessary to execute several processes that can vary depending the system, the processes used in this investigation are the following.

- a) Estimation of local ridge orientation.
- b) Estimation of local ridge frequency.
- c) Fingerprint enhancement (Gabor filter).
- d) Fingerprint skeletonization.
- e) Feature extraction (Minutiae).

The main process can see in figure 2.

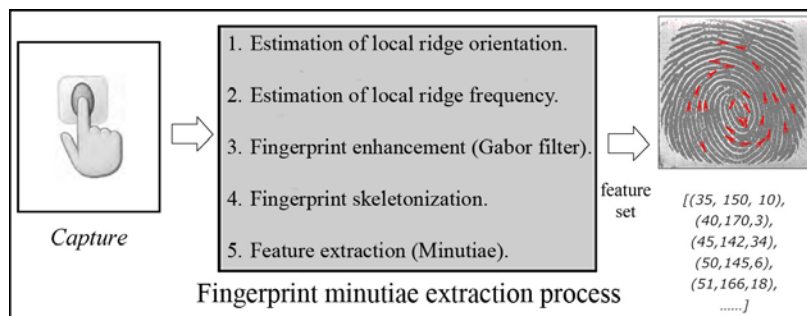


Figure 2: Fingerprint minutiae extraction process.

### 3. Used algorithms

The algorithms used in this investigation allow locate the minutiae set in fingerprint patterns, these algorithms are explained below.

#### 3.1. Estimation of local ridge orientation

An approach for computing local ridge orientation is gradient based in (Rathan *et al.*, 1995), the local ridge orientation at a pixel is the angle that the fingerprint ridges, crossing through an arbitrary small neighborhood, first we use Sobel convolution masks to extract x and y components and computing orientation as the arctangent from these components, this result is smoothed to improve the orientation in defective zones, a representation of this process is showed in figure 3.



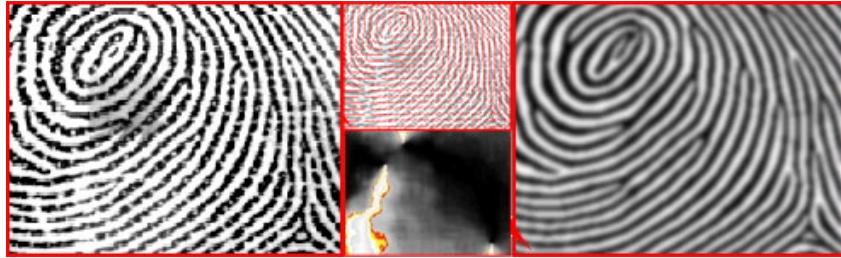


Figure 5: Fingerprint enhancement process.

### 3.4. Fingerprint skeleton

With the enhanced image of the fingerprint we can obtain a skeleton that will facilitate the minutiae extraction, in this investigation we used the method described in (Lam *et al.*, 1992), one pixel wide ridges are the result this allows easy minutiae extraction from binarized and thinned image, a skeleton example is showed in figure 6.

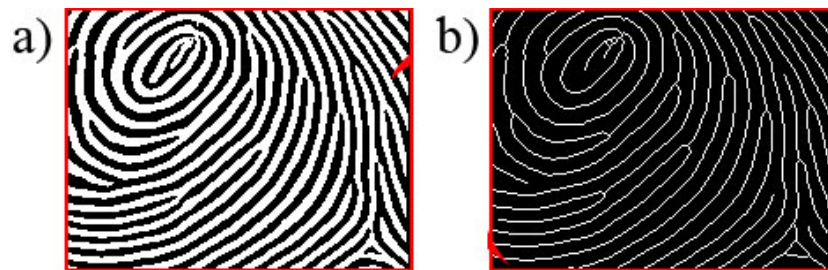


Figure 6 (a) Binary enhanced image b) Fingerprint skeleton.

### 3.5. Minutiae extraction

The performance of minutiae extraction algorithms relies heavily on the skeleton of the fingerprint. In an ideal fingerprint skeleton, ridges and valleys can be easily detected and minutiae can be precisely located in the image, this index-based method is explained in (Arcelli *et al.*, 1984), once a binary skeleton has been obtained, a simple image scan allows the pixels corresponding to minutiae to be detected, figure 7 shows an example of this process.

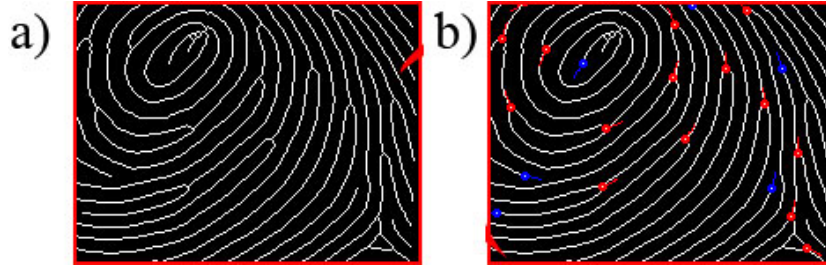


Figure 7 <Minutiae extraction a) Fingerprint skeleton b) Minutiae set.

## 4. Minutiae filtering

To improve a minutiae set is necessary to clean spurious minutiae, detected in erroneous ridges produced by unclearly zones, to validate minutiae and filter spurious minutiae our filter requires to extract both ridge and valley skeletons, to match pairs of ending and bifurcation and inverse, because a ridge ending correspond to a valley bifurcation and a valley ending correspond to a ridge bifurcation, Let  $\theta'(x, y)$  represent the orientation of the anisotropy of the non-overlapping block centered at  $(x, y)$ ,  $I(i, j)$  the grey level of the pixel  $(i, j)$ ; and  $\theta(x, y)$  the local dominant orientation. The local dominant orientation  $\theta(x, y)$  equals  $\theta'(x, y) + \pi/2$  because the orientation is perpendicular to the direction of anisotropy. For fingerprint images, in case the opposite directions cancel each other out, we define the range of the direction angles as  $(0, \pi)$ . Let  $\chi(x, y)$  represent the coherence of the orientation.

The gradients of a Gaussian filter can give an estimation of the underlying oriented pattern; we adopt its orientation as the local direction. First, the image is convolved with a Gaussian filter whose impulse response is given by:

$$gl(x, y) = e^{-(x^2+y^2)/2\sigma^2} \quad (7)$$

The image result can be expressed as:

$$G(i, j) = gl(i, j) * I(i, j) \quad (8)$$

Then one applies a functional operator 3\*3 to obtain the gradients in horizontal and vertical directions  $G_x(x, y)$  and  $G_y(x, y)$ . The amplitude of the gradient is defined as follows:

$$|G_i(i, j)| = ma(g\nabla G(i, j)) = \sqrt{G_x^2(i, j) + G_y^2(i, j)} \quad (3) \quad (9)$$

$$\begin{cases} J_1(i, j) = 2G_x(i, j) * G_y(i, j) \\ J_2(i, j) = G_x^2(i, j) - G_y^2(i, j) \\ J_3(i, j) = G_x^2(i, j) + G_y^2(i, j) \end{cases} \quad (10)$$

Then, the anisotropy orientation estimate by blocks of 8\*8 centered at  $(x, y)$  is:

$$\bar{\theta}(x, y) = \frac{1}{2} \tan^{-1} \left( \frac{\sum_{(i,j) \in \Phi_1} J_1(i, j)}{\sum_{(i,j) \in \Phi_1} J_2(i, j)} \right) \quad (11)$$

where  $\Phi_1$  is a smoothing window centered on the block, with a size of  $W_1$  by  $W_1$  (Alonso *et al.*, 2005, Lam *et al.*, 1992). In fingerprint images, the average width of the ridge or valley is five to eight pixels, and so  $W_1 = 16$  produces a good orientation estimate. Hence,

$$\theta(x, y) = \bar{\theta}(x, y) + \frac{\pi}{2} \quad (12)$$

However, orientation measurements may be unreliable for two reasons (Lam *et al.*, 1992): missing parts of the image or perhaps very noisy zones. In order to discriminate these two cases we need to compare the magnitude of the orientation vector to the mean squares magnitude of the gradient. In fingerprint images, the background shows a constant gray value, in order to make a clear-cut distinction between the print and the background. Therefore, we first set the threshold value  $G_{th}$  of the Gradient as:

$$G_{th} = g_t * (|G_i|_{max} - |G_i|_{min}) + |G_i|_{min} \quad (13)$$

Where  $|G_i|_{max}$  and  $|G_i|_{min}$  are the global maximum and minimum  $g$  gradient amplitude of the image, respectively, and  $g_t$  is the threshold factor (Arcelli *et al.*, 1984). Smaller values of  $g_t$  will encourage weak edges to be identified, while larger values produce noise suppression. For fingerprint images with various contrasts,  $g_t$  is selected in the range [0.05, 0.3]. Therefore, block coherence is defined as:

$$\mathcal{X}(x, y) = \begin{cases} -1, & \text{if } \frac{1}{w_1 * w_1} \sum_{(i,j) \in \phi_1} J_3(i, j) < G_{th} * G_{th} \\ \frac{\left( \left( \sum_{(i,j) \in \phi_1} J_1(i, j) \right)^2 + \left( \sum_{(i,j) \in \phi_1} J_2(i, j) \right)^2 \right)^{\frac{1}{2}}}{\sum_{(i,j) \in \phi_1} J_3(i, j)} & \end{cases} \quad (14)$$

The block is a candidate for the background if the coherence is  $-1$ . Coherence ranges between 0 and 1. For ideal local orientation it is 1; for an isotropic gray value structure without a preferred orientation it is 0 which is correspondent to the noisy regions in the foreground or the regions near core or delta. We then label all of the connected regions whose coherence value is not 1, and choose the one with the largest area as the supposed foreground area of the fingerprint. For the supposed background regions, if they are surrounded by the foreground, they are labelled as noisy regions or low contrast regions. Their coherences are assigned a value of zero. That is:

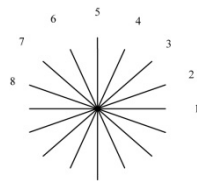
$$\mathcal{X}(x, y) = \begin{cases} -1 & \text{Background} \\ 0 & \text{Noisy} \\ \mathcal{X}(x, y) & \text{Foreground} \end{cases} \quad (15)$$

The average coherence of the foreground indicates the clarity of ridges.

## 5. Ridge direction field detection

To compute the direction field, we define the ridge direction as one of 8 possible directions. To decide the ridge direction of each pixel in the image, we compute the average grey value in direction  $i$  ( $i=1, \dots, 8$  means one of the 8 possible directions). In a centered  $9 \times 9$  window we compute the average grey value of the pixels labeled "i" to obtain  $G[i]$ . The 8 mean grey values are divided into 4 groups with the two directions in each group perpendicular to each other ( $j=1, 2, 3, 4$ ). The absolute value of the difference of mean grey values is calculated in each group as:

$$G_{diff}[j] = |G_{mean}[j] - G_{mean}[j + 4]| \quad (j = 1, 2, 3, 4) \quad (16)$$



|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 |   |   |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|   | 8 |   |   | 2 |   |   |
| 1 | 1 | * | 1 | 1 |   |   |
|   | 2 |   |   | 8 |   |   |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3 | 4 | 5 | 6 | 7 |   |   |

Figure 8<Eight directions to compute. Figure 9<Eight directions in a block.

## 6. Singularity and Core Detection

Many methods have been proposed to detect singular points in fingerprint images, the most popular of which is the Poincare index, which is derived from continuous curves.

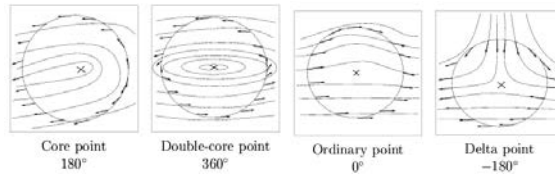


Figure 10<Orientation and Singular Points.

Next, we try to find the singular points by means of the directional image. The method described by Kalle Karu and Anil K. Jain (Rao, 1990, Hong *et al.*, 1990) is not clear at this point. One should compute the difference between two angles by a Poincare Index and take the difference that is smallest in absolute value.

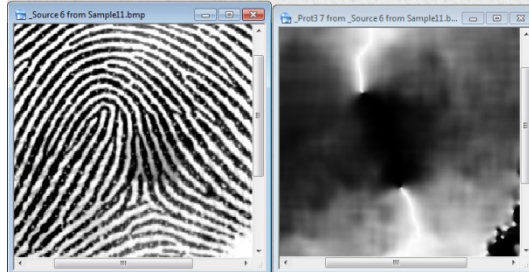


Figure 11 <Orientation changes.

If the Poincare index takes a value of 1, then it is a core point; if takes a value of -1/2 or 1/2 then is a delta point. Let  $\theta(x,y)$  denote the direction of the pixel  $(x,y)$  in an  $M*N$  fingerprint image. The Poincare Index at pixel  $(x,y)$  which is enclosed in a curve can be computed as:

$$Poincare(x, y) = \frac{1}{2\pi} \sum_{k=0}^{N-1} \Delta(k) \tag{17}$$

$$\Delta(k) = \begin{cases} \delta(k) & |\delta(k)| < \frac{\pi}{2} \\ \delta(k) + \pi & |\delta(k)| \leq \frac{\pi}{2} \\ \pi - \delta(k) & |\delta(k)| \geq \frac{\pi}{2} \end{cases} \tag{18}$$

$$\delta(k) = \theta(x_{(k+1) \bmod N}, y_{(k+1) \bmod N}) - \theta(x_k, y_k) \tag{19}$$

It goes in a counter-clockwise direction from 0 to N-1. For our method, N is 4 as one can see in figure 11.

|            |              |
|------------|--------------|
| $(x-1, y)$ | $(x-1, y+1)$ |
| $(x, y)$   | $(x, y+1)$   |

To compute the Poincare Index at pixel  $(i,j)$  by means of the modified version of Poincare Index, the closed digital curve is selected as 4 pixels. In order to make calculation easier, a direction from 0 to 7 is used to compute the Poincare Index.

## 7. Poincare Index

Blocks that may contain singularities are detected by this method. Then the Poincare Index at pixel  $(x,y)$  which is enclosed by a digital curve of 4 pixels can be computed in the detected blocks, and the resulting value is  $Poincare(x,y)$ .

- If  $Poincare(x, y) = +0.5$ , the point is a core point;
- If  $Poincare(x, y) = -0.5$ , the point is a delta point;
- Otherwise, the point is not a singularity.



If the number of core points is greater than 2, or the number of delta points  $N_d$  is greater than 2, we smooth the orientation field until the number of core points or delta points is not greater than 2.

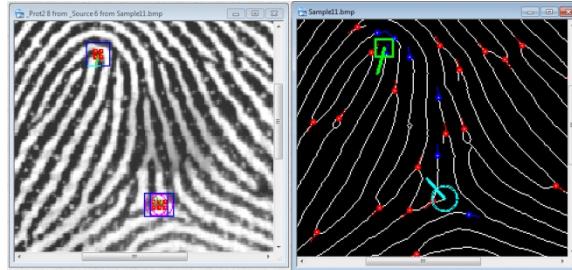


Figure 12-Poincaré detection and classification

## 8. Experimental Results and Conclusions

For methods to find singular points like those of Kawagoe and Tojo, Bowen, Karu and Jain, the improvements derived of working in blocks produce an appreciable increase in processing speed (Woods *et al.*, 1997). Thus, these procedures can be implemented on a general-purpose AFIS. For 50 images, 38 correct singular points and 12 false alarms can be found. This means that 76% of images with singular points are correctly identified, even for bad quality images; several problems can make this method produce incorrect results: noisy images, scars or damage in the fingerprint, and images of reduced clarity.

The method described in this paper produces rather quick calculations and makes it possible to take advantage of the directional field in order to recognize singular points. It is also able to precisely find a mask in order to separate the image from the background. Finally, its ability to find the direction of minutiae suggests other possible applications.

Fingerprint identification in a large dataset is a very time consuming task. Traditionally, fingerprints have been classified into categories based on information found in the global patterns of ridges. In large-scale fingerprint identification systems, elaborate methods of manual fingerprint classification systems were developed in order to index individuals into bins based on classification of their fingerprints; these methods can eliminate the need to match an input fingerprint to the entire fingerprint database in identification and significantly reduce the computing requirements. Fingerprint indexing can evidently reduce the number of comparisons, and the proposed method accelerates the indexing process.

## 9. References

Alonso Fernandez F.- Fierrez-Aguilar J. and Ortega-Garcia J., 2005. An Enhanced Gabor Filter-Based Segmentation Algorithm for Fingerprint Recognition Systems, In *Int. Symp. on Image and Signal Processing and Analysis*.

- Arcelli C. and Baja G.S.D., 1984. A width independent fast thinning algorithm. In *IEEE Transactions on Pattern Analysis Machine Intelligence*.
- Hong L., Wan Y., and Jain A. K., 1998. Fingerprint image enhancement: Algorithm and performance evaluation. In *IEEE Trans. Pattern Anal. Machine Intell.*, 20: 777–789.
- Hung D.C.D., 1993. Enhancement and feature purification of fingerprint images, *Pattern Recognition*.
- Lam L. - Lee S.W. and Suen C.Y., 1992. Thinning methodologies: A comprehensive survey. In *IEEE Transactions on Pattern Analysis Machine Intelligence*.
- Miller B., 1994. Vital signs of identity. *IEEE Spectrum*, 31 (2): 22-30.
- Rao A. R., 1990. *A Taxonomy for Texture Description and Identification*, New York: Springer-Verlag.
- Ratha N.K. - Chen S.Y. and Jain A.K., 1995. Adaptive flow orientation-based feature extraction in fingerprint images, *Pattern Recognition*, 1995.
- Woods K., Kegelmeyer W.P., and Bowyer K.W., 1997. Combination of Multiple Classifiers Using Local Accuracy Estimates. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(4): 405-410.