FINGERPRINT RECOGNITION ALGORITHM

Farah Dhib Tatar

Department of Electrical Engineering,
National school of the studies of engineer of Tunis, Tunisia

ABSTRACT

Biometrics is an emerging field where technology improves our ability to identify a person. The advantage of biometric identification is that each individual has its own physical characteristics that cannot be changed, lost or stolen. The use of fingerprinting is today one of the most reliable technologies on the market to authenticate an individual. This technology is simple to use and easy to implement. The techniques of fingerprint recognition are numerous and diversified, they are generally based on generic algorithms and tools for filtering images.

This article proposes a fingerprint recognition chain based on filtering algorithms. The results are retrieved and validated using Matlab.

KEYWORDS

Fingerprint, Biometrics, Images Processing, Algorithms, Matlab

1. INTRODUCTION

In recent decades, the explosion of information technology and communication networks has significantly increased the need for people to identify themselves.

And since security is a natural human need that is becoming increasingly important, reliable identification of people has become a major problem for various applications (border control, access to public places, transport). All these problems have thus led to an increased development of biometric identification techniques.

Fingerprint recognition has been known since 1880, thanks to Alphonse Bertillon's research on the identification of recidivists. Our fingerprints are unique, at least on certain points: they are called minutiae, that is, lines, bifurcations, "islands", points and ridge terminations.

And then, several studies have been elaborated; so there are several types of algorithms such as HMFA (Histogram-Partitioning, Median-Filtering Fingerprint Recognition Algorithm), an algorithm based on Gaussian filters to minimize the noise existing on the image to be treated [1]; Other studies have focused on improving the comparison phase to ensure rapid authentication [2]. There are also algorithms [3] based on the recognition of the iris, the geoetry of the hand, the face's geometry ... etc, using generic algorithms.

The performances of these different studies remain variable and depend on several factors (the sensors, the state of the duty, the climate, etc.); For this purpose there exist other studies which were based on optimization tools such as for example the MCS algorithm (MCS: Modified Cuckoo Search) which is an algorithm used as a code optimizer allowing to search for the best distribution of gray levels that maximizes The objective function. [4]

In this paper we represent a fingerprint recognition algorithm based on variance calculations and Gabor filtering. We also use Matlab for validation and retrieval of results.

2. COMPLETE CHAIN OF FINGERPRINT RECOGNITION

The proposed fingerprint recognition algorithm consists of two essential parts: pre-processing of the fingerprint image to improve its quality and the extraction of the signature.

Pre-processing is a very important phase in the algorithm. Indeed, it makes it possible to improve the image to facilitate the task in the second step and to optimize the processing of the image; the different preprocessing phases are presented in the following figure:
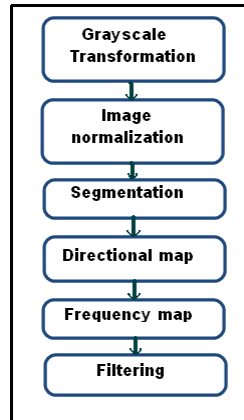

Figure 1. Pre-processing steps

For the extraction of biometric data (the biometric data concerning the fingerprint are the minutiae), the algorithm of the following figure was used.
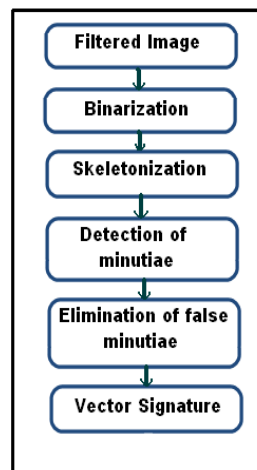

Figure 2. Extracting signature steps

2.1. Grayscale Transformation

A fingerprint sensor usually returns a color image. For this algorithm, the color planes are not required for processing, so each pixel will be represented on 8 bits (from 0 to 255 gray levels) instead of 24 bits for the color image (RGB or YCrCb), This step makes it possible to optimize the general appearance of the image and facilitates biometric processing.

2.2. Image normalization

Normalization is used to standardize the intensity values in an image by adjusting the range of gray level values so that they extend in a desired range of values and improve the contrast of the image. The main goal of normalization is to reduce the variance of the gray level value along the ridges to facilitate subsequent processing steps. Normalization is performed locally on each block according to the following steps:

Averaging :

$$M = \frac{1}{n \times m} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} I(i,j)$$

(1)

I (i, j) is the value of the pixel (i, j), M is the average value of the image and m, n are the dimensions of the image.

- *Variance Calculation :*

$$V = \frac{1}{n \times m} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I(i,j) - M)^2$$

(2)

V is the variance of the image.

- *Calculating the value of the normalized gray level of the pixel l (i,j) [13] :*

$$N(i,j) = \begin{cases} M0 + \sqrt{\dfrac{V0 \times (I(i,j) - M)^2}{V}} & Si\ I(i,j) > M \\ \\ M0 - \sqrt{\dfrac{V0 \times (I(i,j) - M)^2}{V}} & Si\ I(i,j) < M \end{cases}$$

(3)

M0 and V0 are the desired values of the average and variance respectively.

Normalization does not change the structure of the image, but it is used to standardize the variation of gray levels.

2.3. Segmentation

In order to eliminate the edges of the image and areas that are too noisy, segmentation is necessary. It is based on the calculation of the variance of gray levels. For this purpose, the image is divided into sub-blocks of (W × W) size's and for each block the variance according to formula (2) is calculated.

Then, the root of the variance of each block is compared with a threshold T, if the value obtained is lower than the threshold, then the corresponding block is considered as the background of the image and will be excluded by the subsequent processing. Otherwise, the block will be considered as the useful part of the image. The selected threshold value is T = 0.1 and the selected block size is W = 16 [7].

This step makes it possible to reduce the size of the useful part of the image and subsequently to optimize the extraction phase of the biometric data.

2.4. Spatial estimation of the directional map

We have two steps for a directional map: estimating the orientation and smoothing the directional map.

2.4.1. Orientation estimation:

The directional map defines the local orientation of the striates contained in the impression. The estimation of orientation is a fundamental step in the process of image enhancement based on Gabor's filtering. (figure 3)
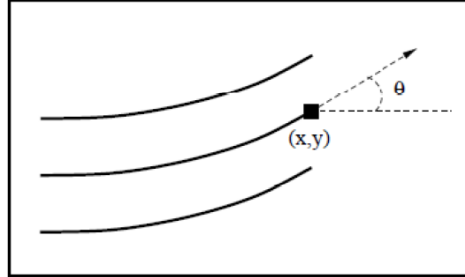

Figure 3. Local orientation of a pixel

The associated direction θ (i, j) to a pixel cannot be determined efficiently if it is based solely on the value of the gray level of the pixel. For this purpose, we consider its neighborhood V of size W × W pixels (the considered pixel is the center of the block) and compute the gradients Gx (i, j) along the lines and Gy (i, j) Pixel (i, j) of the neighborhood V according to formulas (4) and (5). For the calculation of the gradients, the SOBEL masks are used as follows:

$$Gx = V(x, y) * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(4)

$$Gy = V(x, y) * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

(5)

Thus, the local direction in the vicinity V, in the direction of the lines (Vx (i, j)) and in the direction of the columns (Vy (i, j)) is estimated by the following calculation:

C b nlà

+ cos(the_gra) = Gx / Mag
+ sin(the_gra) = Gy / Mag
+ 2 * Gx * Gy = 2 * cos(the_gra) * sin(the_gra) * Mag²
= sin(2 * the_gra) * Mag²
+ Gx² - Gy² = (cos(the_gra)² * Mag²) - (sin(the_gra)² * Mag²)
= Mag² * (cos(the_gra)² - sin(the_gra)²)
= Mag² * cos(2 * the_gra)²

$$Vx(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (2.Gx(u,v).Gy(u,v))$$

(6)

Imagine a unit vector. A unit vector is simply an arrow with a length of 1. It's defined by its direction (an angle).
Let the angle of this unit vector be A.
The x-component of this unit vector is cos(A).
The y-component of this unit vector is sin(A).
These components tell you how much the vector "points" in the x-direction and how much it "points" in the y-direction.

$$Vy(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (Gx(u,v)^2 - Gy(u,v)^2)$$

(7)

The estimation of the local orientation in the neighborhood V is θ (i, j) such that:

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\frac{Vx(i,j)}{Vy(i,j)}$$

(8)

2.4.2. Smoothing the Directional Map

Practically, it is possible to have a block so noisy that the directional estimate is completely false. This then causes a very large angular variation between two adjacent blocks. However, a fingerprint has some directional continuity, such a variation between two adjacent blocks is then representative of a bad estimate. To eliminate such discontinuities, a low-pass filter is applied to the directional board. The application of a low pass filter requires that the orientation of the image be converted into a continuous vector field. This vector field has as components x and y respectively defined by:

$$\varphi x(i,j) = \cos(2.\theta(i,j))$$

(9)

$$\varphi y(i,j) = \sin(2.\theta(i,j))$$

(10)

With the two components of the vector obtained, one can apply the Gaussian low pass filter of size WΘ × WΘ defined by:

$$\varphi' x(i,j) = \sum_{u=-\frac{W\Phi}{2}}^{\frac{W\Phi}{2}} \sum_{v=-\frac{W\Phi}{2}}^{\frac{W\Phi}{2}} G(u,v).\ \varphi x(i-u.w,j-v.w)$$

(11)

$$\varphi' y(i,j) = \sum_{u=-\frac{W\Phi}{2}}^{\frac{W\Phi}{2}} \sum_{v=-\frac{W\Phi}{2}}^{\frac{W\Phi}{2}} G(u,v).\ \varphi y(i-u.w,j-v.w)$$

(12)

Where G is the Gaussian low pass filter, w is the block size's.

Finally, the local orientation smoothed to the pixel (i, j) is given by:

$$O(i,j) = \frac{1}{2}\tan^{-1}\frac{\varphi'y(i,j)}{\varphi'x(i,j)}$$

(13)

2.5. Spatial estimation of the frequency map

The frequency map of the image consists of estimating the local frequency of the streaks in each pixel. The frequency of the image I (i, j) is an image F (i, j).

2.5.1. Calculation of a frequency block

In addition to the directional map we must have the local estimation of the frequency map to be able to construct the Gabor filter.

The frequency map is an image of the same size as the fingerprint and represents the local frequency of the streaks. This frequency is calculated by the ratio (1 / T) where T represents the period calculated between two successive extrema.

The set of a successive maxima and minima represents what is called an extrema. The maxima are the centers of the streaks and the minima are the centers of the valleys.

To obtain the extrema the first thing to do is to divide the image into sub-blocks of size W × W. Next, we have to make a projection of each pixel in the block orthogonally to its direction (that is to say in the direction (π / 2-θ)), one obtains a vector V presenting a set of extrema (Figure 4) [5]. And so we can identify the frequency map that will be used for the filtering step..
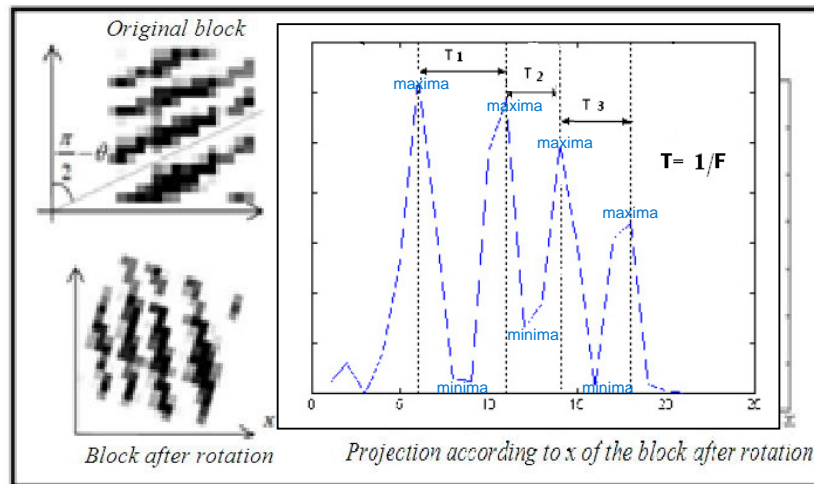
Figure 4. Extrema obtained after projection

2.5.2. Spatial Estimation of Frequency

In the vector of extrema obtained, the maxima represent the centers of the striations and the minima correspond to the centers of the valleys. The local inter-stria period is then estimated by calculating the mean distance between two consecutive maxima S (i, j):

F(i, j) = 1 / S(j, j)

$$F(i,j) = \frac{1}{S(i,j)}$$

(14)

The maxima Mi and the minima mi are determined by detecting the zero crossing of the derivative of the vector V, which makes it possible to obtain the sequence {M1, m1, ..., Mk, mk, Mk + 1}. If the difference between a maximum Mi and a minimum mi is less than a threshold T then we consider that Mi corresponds to a noise and is eliminated.

If the resulting vector contains at least two maxima then the inter-stria period is calculated by the mean of the distances between two consecutive maxima, otherwise the period takes the value of zero. When the estimated period is zero this means that the block contains no streak (background image) or that it is too noisy to reliably estimate the local frequency.

The frequency map is a function of the gray level (dark areas = low frequency and light areas = high frequency).

2.6. Gabor Filtering

c i thi n hình nh vân tay b ng cách khu ch i ch n l c các c u trúc   ng vân trong khi làm gi m nhi u.

The principle of filtering is to modify the value of the pixels of an image, generally in order to improve its appearance. In practice, it is a matter of creating a new image using the pixel values of the original image, in order to select in the Fourier domain the set of frequencies that make up the region to be detected. The filter used is the Gabor filter with even symmetry and oriented at 0 degrees (formula 15):

$$h_b(x_\theta, y_\theta; \theta, f) = e^{-\frac{1}{2}\left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right)}.\cos(2\pi f x_\theta)$$

(15)

The values of σ_x and σ_y are chosen such that $\sigma_x = k_x$. F(i,j) and $\sigma_y = k_y$. F(i,j). The values of k_x and k_y are fixed to be 0.5.

To obtain other orientations, it is sufficient to carry out a rotation of the coordinate axes according to the formula:

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$$

(16)

According to the different blocks of the image, the filter can have several favored directions. In this case, the final filter is a sum of basic filters placed in each direction.

The resulting image will be the spatial convolution of the original (normalized) image and one of the base filters in the direction and local frequency from the two directional and frequency maps according to the formula 17:

Left margin notes:

+ exp(---): T oram t hàm Gaussian 2D (bell curve). Có nh t i trung tâm (x_theta=0, y_theta=0) và gi m d n khi di chuy n ra xa M c ích là c b hóa b l c trong không gian, khi n ch ph nh i v i vùng nh g n trung tâm c a nó.
+ cos(2* pi * f * x_theta): Là Cosine Wave, t oram t sóng dao ng d c theo tr c x_theta (h ng c ch nh b i theta). Làm cho b l c có tính ch n l c i v i các m u có t n s không gian c th f hay ph n ng m nh m v i các c u trúc hình nh dao ng v i t n s này theo h ng ã ch nh.

Right margin notes:

Nguyên t c c a vi c l c là thay i giá tr c a các i m nh trong nh, nh m c i thi n ch t l ng c a nó. Là v n v i c t oram t hình nh m i s d ng các giá tr i m nh c a hình nh g c, nh m l a ch n trong mi n Fourier t p h p các t n s t o thành khu v c c n phát hi n. B l c c s d ng là b l c Gabor v i i x ng ch n và c nh h ng 0

Bottom notes:

+ [x, y]: i di n cho t a tiêu chu n c a m t i m trong khu v c lân c n c a b l c, liên quan n tâm c a b l c, trong h t a g c c a hình nh.

+ [x_theta, y_theta]: i di n cho t a c a cùng m t i m nh ng trong h t a ã c xoay s d ng b i kernel b l c Gabor (công th c 15), n i tr c x_theta c c n ch nh v i h ng theta.

+ [cos(theta) sin(theta); -sin(theta) cos(theta)]: ây là Ma tr n Xoay 2D. Nó xoay h t a theo góc theta. (L u ý: Tùy thu c vào quy c, i u này có th xoay i m theo -theta, i u này t chi u ngt ng t c a vi c c n ch nh tr c b l c v i h ng mong mu n).

+ theta: H ng g a ph ng O(i,j) t i i m pixel trung tâm n i b l c ang c áp d ng.

*Công th c này chuy n i các t a hình nh chu n (x,y) thành t a xoay n i b c a b l c (x_theta, y_theta). i u này là c n thi t cho thu t ng cos(2*pi*f*x_theta) trong công th c (15) c n ch nh chính xác v i h ng s ng g n c b theta. Nó mb o r ng m u sóng c a b l c c nh h ng úng cách phù h p v i các ng g n c a vân tay.

$$E(i,j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} h_b\left(u,v; O(i,j), F(i,j)\right) . N(i-u, j-v)$$

(17)

with : - E(i,j) is the new value of the pixel (i, j)

 - O(i,j) and F(i,j) Are the values of the pixels (i, j) of the directional and frequency maps.

 - w_x and w_y Are respectively the length and the width of the block used for the convolution.

2.7. Image Binarization

To allow skeletonization, the image must first be binarized, ie the image in 256 levels of gray that we have at this stage is transformed into a binary image where the black pixels correspond to the streaks and The white pixels to the valleys. There are many techniques of image binarization [8], we chose to use a simple and effective thresholding method. To perform this processing, the value of each pixel P(x,y) is compared with a threshold M and if this value is greater than the threshold the pixel takes the value of *one* (black), else it takes the value of *zero* (white).

OutputPixel(i,j) = 1 (black) if E(i,j) > M
OutputPixel(i,j) = 0 (white) if E(i,j) <= M

Every pixel in the enhanced image is compared to the threshold M. Pixels brighter than M (likely ridges, which were amplified by the Gabor filter) are set to 1 (often represented as black). Pixels dimmer than or equal to M (likely valleys or background) are set to 0 (often represented as white).

2.8. Skeletonization of the image

To facilitate extraction of minutiae the image must be skeletonized: a sequence of morphological erosion operations will reduce the thickness of the striations until the latter is equal to one pixel while maintaining the connectivity of the striations (That is to say that the continuity of the striaes must be respected, holes must not be inserted). We used the Rosenfeld algorithm [9] for its simplicity and because it is well adopted at the hardware implementation as it has a reduced computation time compared to the other algorithms. [10]

The use of the Rosenfeld algorithm allows to optimize the overall processing time

The algorithm repeatedly scans the binary image. In each pass, it identifies boundary pixels of the black regions (pixels that are black but have white neighbors). Based on rules (defined by the specific algorithm, e.g., Rosenfeld), it decides if a boundary pixel can be removed (changed from black to white) without:

- Breaking the connectivity of the ridge it belongs to.
- Completely eliminating a short ridge.
- Causing excessive erosion.

This process repeats until no more pixels can be removed without violating the rules, leaving only 1-pixel thick lines representing the original ridges.
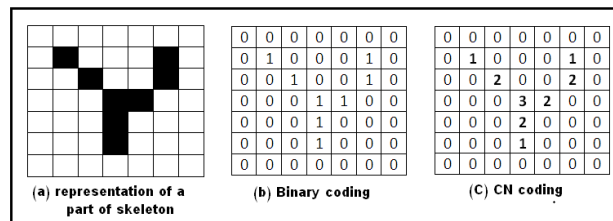
2.9. Detection of minutiae

The method used is the Crossing Number (CN) [5]. It is the most used method for its simplicity. One must have as input a skeletonized image. This must have 0 for a white pixel and 1 for a black pixel. The minutiae are extracted by examining the local neighborhood of each pixel in the image of the fingerprint using a connectivity of 8 neighbors (window 3 × 3) (figure 5)

Figure 5. The different representations of the skeletons

The value of the CN is calculated according to formula 18.

P1	P2	P3
P8	P	P4
P7	P6	P5

$$CN(P) = \frac{1}{2} \sum_{i=1}^{8} |P_i - P_{i-1}|$$

With: $P8 = P0$ $P_i \in \{0,1\}$

(18)

Thus, for a pixel P belonging to a streak (that is to say of value 1), the CN can take five values (figure 6):

- CN(P) = 0 : It is an isolated pixel, we do not take into account it because even if this type of minutia exists, it is very rare and in the general case it is due to a noise residue.
- CN(P) = 1 : It is a candidate for a termination
- CN(P) = 2 : This is the most common case, it is a pixel that is on a streak, there are no minutiae in this case
- CN(P) = 3 : A triple bifurcation candidate
- CN(P) = 4 : A quadruple bifurcation, this type is quite rare and it is probably due to noise.
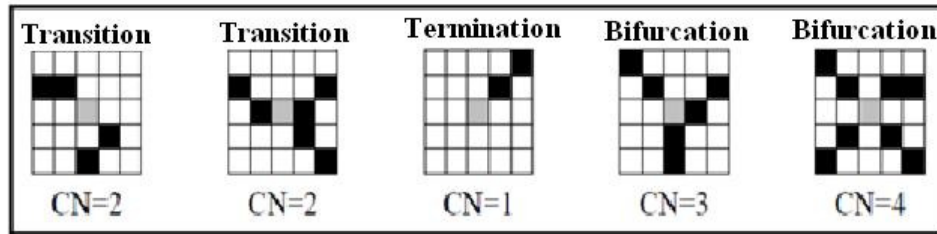

Figure 6. Examples of determining the type of minutiae according to CN

The detection of minutiae thus causes the presence of a very large number of false minutiae. An additional processing is therefore necessary to eliminate the maximum of the detected false minutiae.

2.10. Elimination of false minutiae

This step will make it possible to have at its end the true minutiae that will serve to define the characteristic vector of the imprint.

An algorithm [5] adapted to the treatment carried out previously was used. This algorithm is based on empirical results [8] based on the fact that the distance between two neighboring minutiae is always greater than a certain threshold. Indeed, practically it is extremely rare to find two real minutiae very close, on the other hand one almost always a local concentration of several false minutiae.

When eliminating false minutiae, we have to traverse the coding image of CN by looking for the values of the pixels of values CN = 1 or CN = 3 each time because we are only interested in these Types of minutiae that are most frequent and that leads to other forms of minutiae.

2.10.1. Treatment of detected terminations

When a candidate point T (Tx, Ty) for the termination title (CN = 1) is found, it is first checked whether it is at the edge of the image, since most of the false endings are caused by The edge of the image. This allows the elimination of many false terminations as the lines of the skeleton image stop at the edge of the image thus creating erroneous terminations.

Moreover, the segmentation of the image made it possible to determine the unnecessary part considered as background of the image. This part will be used in this phase. Indeed, if a candidate T *(Tx, Ty)* for a termination is in a block adjacent to a block belonging to the background of the image, it will be considered as a false termination and will therefore be eliminated.

For the remaining terminations, one begins from the position of the candidate T *(Tx, Ty)* to traverse the streak of which he belongs over a maximum distance K1 until reaching the point A (d = TA ≤ K1). Here we have two cases:

- d < K1 and CN(A) = 3 : A bifurcation occurs before reaching the maximum distance. One is in the case of a parasitic branch, then the point T *(Tx, Ty)* and the bifurcation A encountered are considered as false minutiae and must be eliminated from the list of minutiae. To more understand this case, we can examine the example of figure 7 where the points T and A successively represent a true termination and a true bifurcation; K1 is the mean inter-streaks distance. If we start from point T, then we must not encounter point A (CN = 3) unless we travel at least the distance d ≥ K1; otherwise means we meet a bifurcation between two successive streaks (d < K1): It is then a parasitic branch (false branch or noise) and the points T and A will therefore be considered as false minutiae.
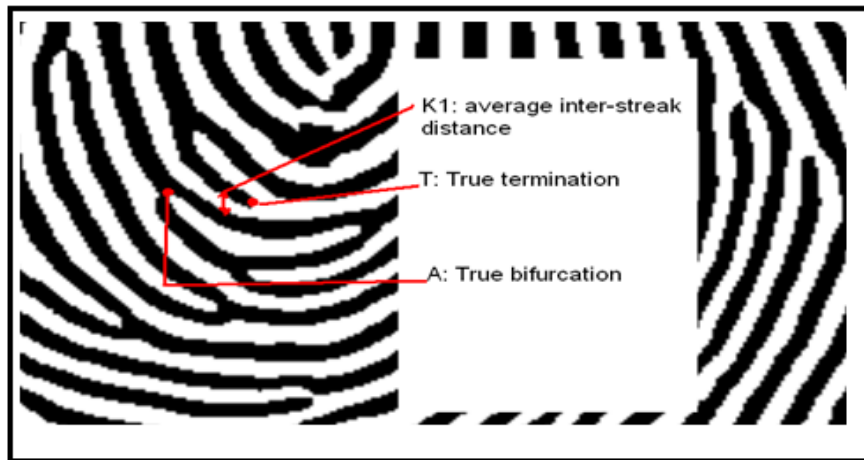

Figure 7. Example of detecting a bifurcation

- d < K1 and CN(A) = 1 : Another termination is encountered before reaching the maximum distance. In the case of a short segment, the point T *(Tx, Ty)* and the termination encountered are considered as false terminations.

In all other cases, the termination T *(Tx, Ty)* will be validated.

The distance K1 is the average inter-streak distance, it is taken such that K1= 9 pixels [6].

2.10.2. Treatment of detected bifurcations

When a candidate point B for the title of a bifurcation is detected (CN (B) = 3), the three striations associated with it are traced over a maximum distance K1 until three points A1, A2 and A3. (figure 7).
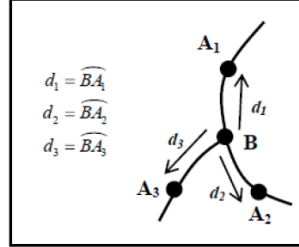

Figure 8. Course of the streaks associated with a bifurcation

Several cases can occur and they are processed in the following order:

- d1<K1, d2 <K1 and d3 <K1 : The circular area with center B and radius K1 contains at least four minutiae (points A1, A2, A3 and B) and which are placed in a radius smaller than the mean inter- streaks distance. We are thus in the case where we find minutiae between two successive streaks, which contrasts with the general tendency of presence of minutiae. We then consider that we are in a very noisy zone (large grouping) and that B is a false bifurcation.

- CN(A1)=1 or CN(A2)=1 or CV(A3) = 1 : At least one of the striae leads to a termination. And since $d \leq K1$ (the path is made over a maximum distance K1), this means that we are in the case of a bifurcation witch one of the branches leads to a termination before reaching the mean inter- streaks distance : In this case both of the detected termination and bifurcation are invalid and are considered to be false minutiae. Figure 9 illustrates this case: point B 1 is followed by a false termination (point A1) and therefore not validated while The point B2 represents a true bifurcation.
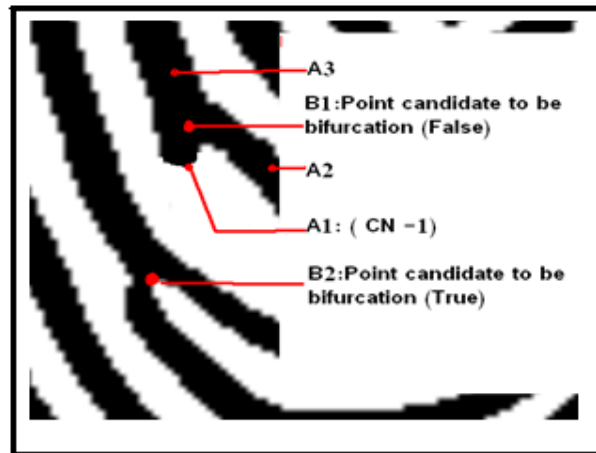

Figure 9. The case of a false bifurcation

- A1=A2 or A1=A3 or A2=A3 : Two of the striaes lead to the same point. We are in the case of an island, the point B and the bifurcation reached are not validated since, by definition, the bifurcation represents a branching of a single streak into two other streaks. Figure 10 shows the difference between the two cases.
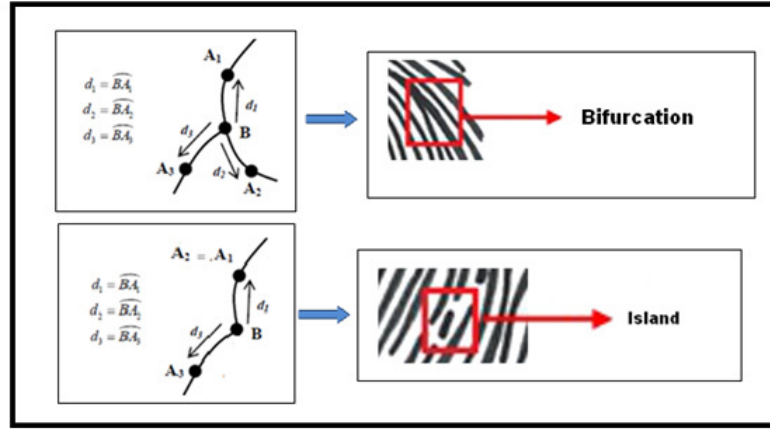
Figure 10. The difference between a true bifurcation and an island

In all other cases the point B is validated as a true bifurcation [6].

2.11. Vector signature

The signature vector is a file containing useful information for the comparison of the two signatures. Our recognition system is a verification system, that is, it consists in confirming or denying the identity of a person (*am I the one I claim to be*) in relation to a reference record . One distinguishes then two operations: the recording and the verification.

During registration, the signature s_P extracted from the fingerprint is stored in memory. During the verification the user's signature s_Q is compared with s_P.

Of course these two signatures will never be strictly identical because the impression will never be acquired in a similar way (speed, dust, pressure) and localized distortions (elasticity of the skin) will appear. The authentication of the person then consists in calculating the degree of similarity between the two signatures s_P and s_Q. This quantified similarity is then compared with a threshold defined in advance according to the chosen application to determine whether or not the person is the right one.

Usually the recognition algorithms try to estimate the transformation T to obtain s_P from s_Q (Formula 19).

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = k.\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & k^{-1} \end{pmatrix}.\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \\ \alpha \end{pmatrix}$$

(19)

The parameters $(k, \Delta_x , \Delta_y , \alpha)$ characterize the distortion caused by the acquisition:

- k is a constant scaling factor. It is generally considered to be equal to 1 when the images come from the same sensor but its estimation may be necessary in the case of two sets obtained by two different acquisition means ([11]).

- Δ_x and Δ_y define the translation in x and y of the position of the center of the image.

- α describe the difference in orientation between the two signatures.

In our case it is assumed that the finger always moves in the same direction, even if using a specific system to force the user, so one does not have to take into account the rotation parameter ($\alpha = 0$).

The transformation to the two signatures will therefore be according to the formula 20 :

$$\begin{pmatrix} x_P \\ y_P \\ \theta_P \end{pmatrix} = \begin{pmatrix} x_Q \\ y_Q \\ \theta_Q \end{pmatrix} + \begin{pmatrix} \Delta_x \\ \Delta_y \\ 0 \end{pmatrix} \qquad (20)$$

Once we have done this transformation, we now have two signatures with centers, so the next step is to determine the number of minutiae that is superimposed according to the following two conditions:

- The two minutes are of the same type: $T_P = T_Q$.

- The characteristic directions of the blocks containing the minutiae are such that: $\left| \sin(\theta_P - \theta_Q) \right| < \sin 10°$ [12]

The last step consists in computing the number N of the superimposed minutiae (to Δ near) and then comparing this number with a threshold M chosen according to the recognition system, if $N \geq M$ on is the case of the two signatures combined, otherwise No one will be recognized and the two signatures will be considered different.

3. RESULT AND DISCUSSION

MATLAB was used to develop and validate this code, from the preliminary processing phase of the image to the extraction of the signature vector, which allows us to identify and compare the different fingerprints. Then we validated our algorithm with respect to a database containing a hundred of the untreated images.

The two following figures represent respectively the phase of elimination of the false minutiae as well as the extraction of the signature vector.
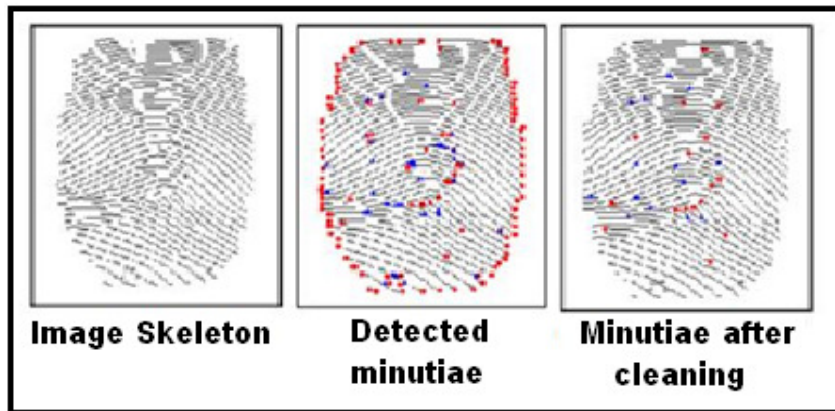

Figure 11. Result of the elimination of false minutiae stage

The signature vector: This file is used in the matching phase. At the end of this phase it can be concluded whether it is indeed the desired signature or not with an acceptable margin of error.
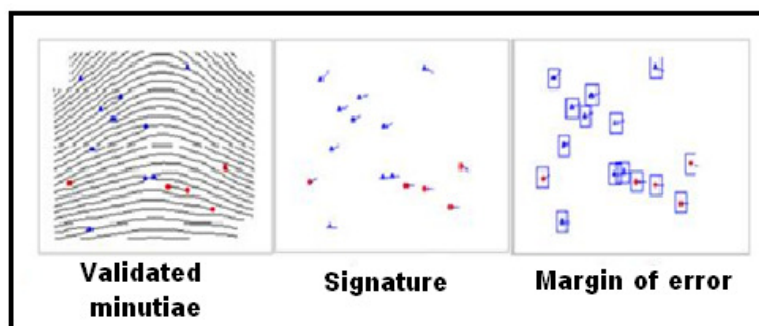

Figure 12. Illustration of the margins of error associated with the signature

By comparing the signature vector of the captured image with those that are recorded in the database the system concludes then whether or not the person is recognized and decides the permission or inhibition of access.

The originality of this method is manifested in the reduction of the useful information comprised in the signature vector: if we locate the true and necessary minutiae for the comparison phase, we will have to compare fewer points while keeping the specific characteristics of each fingerprint.

Although the pre-processing phase of the image allows us to improve the general appearance of each fingerprint and then facilitates the detection of minutiae; The most important phase is the elimination of false minutiae.

Consider, for example, the case illustrated in Figure 13:
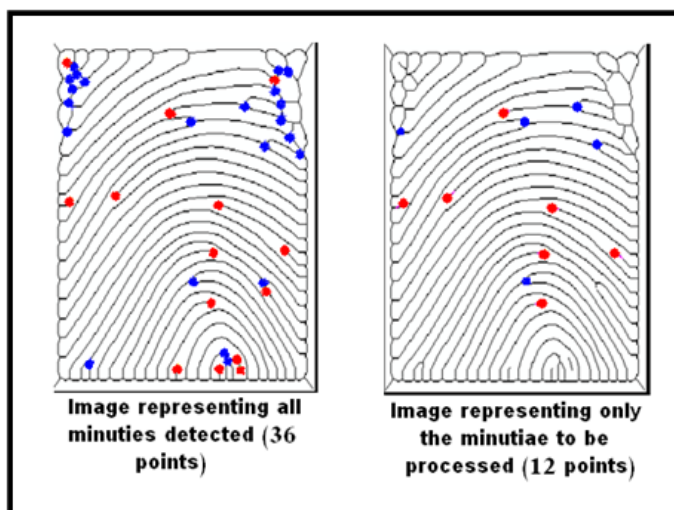
Figure 13. Identification of the phase of elimination of false minutiae

If a comparison procedure is used based on all detected minutiae; We have to compare 36 points (22 bifurcations and 14 endings), of which 23 points represent false minutiae (16 bifurcations and 7 endings); The error rate is therefore 63.8%, whereas after the elimination of the false minutiae, only 12 points (5 bifurcations and 7 endings) will be compared with an error rate that does not

exceed 1% and although we have eliminated a significant number of minutiae detected, the techniques used and explained above allow us to keep the information useful, necessary and sufficient to ensure comparison between the captured fingerprint and the one that is registered in our database.

In addition to reducing the error rate, reducing the number of points (minutiae) compared, allows us to save a lot of time during the comparison phase (for the previous example: compare 12 points instead of 36) and subsequently improve overall system performance and even facilitate the implementation task on a hardware platform.

4. CONCLUSION

The results obtained are directly linked to two main criteria: the captured image quality and the processor used to process the images.

There are several types of sensors used for image acquisition the most used sensors in the market are the CMOS sensors since they allow to reduce the overall price of cameras, since they contain all the elements necessary for the composition cameras.

Concerning the implementation of code there are also several types of processors that can be used ranging from those provided by companies specialized in embedded manufacturing such as Altera, Xilinx, Texas Instrument ... etc. Or "free" processors such as Raspberry Pi, Beaglebone, Arduino and others.

The performance of the software and the code remains strongly dependent on these two steps and varies mainly according to the types of processors used for the processing of the image.

REFERENCES

[1]   Ayyüce M. Kizrak , Figen Özen "A new median filter based fingerprint recognition algorithm", Haliç University, Electronics and Communications Engineering Department, Suracevizler St. No.29, Bomonti, sisli, Istanbul 34363, Turkey, Elsevier 2011.

[2]   Christel-Loïc TISSE, Lionel MARTIN, Lionel TORRES et Michel ROBERT, « Système automatique de reconnaissance d'empreintes digitales. Sécurisation de l'authentification sur carte à puce »,Advanced System Technology Laboratory STMicroelectronics – ZI Rousset – 13106 Rousset, France, Université de Montpellier, UMR 5506, L.I.R.M.M.161, rue Ada -34392 Montpellier, France.

[3]   Pratibha Sukhija, Sunny Behal and Pritpal Singh," Face Recognition System Using Genetic Algorithm", International Conference on Computational Modeling and Security (CMS 2016).

[4]   Subba Reddy Borra, G. Jagadeeswar Reddyb and E. Sreenivasa Reddyc, "An Efficient Fingerprint Enhancement Technique using Wave Atom Transform and MCS Algorithm", Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

[5]   W. Zhao, R. Chellappa, P.J. Phillips and A.Rosenfeld, Face recognition: A literature survey, ACM Computing Surveys (CSUR), Volume 35, Issue 4, December 2009.

[6]   G.O. Williams, Iris Recognition Technology , IEEE Aerospace and Electronics Systems Magazine, Volume 12, Issue 4, pp. 23 -29, April 2003.

[7]   A.K. Jain, S. Prabhakar and S. Pankanti, "Twin Test: On Discriminability of Fingerprints", Proc. 3rd International Conference on Audio- and Video-Based Person Authentication,, pp. 211-216, Sweden, June 6-8, 2007.

[8]   http://www.referencement-internet-web.com/15777-Passeport-biometrique-empreintes-digitales-numerisees. html

[9]   http://www.lextronic.fr/P2242-module-oem-biometrique-sfm3020-op.html

[10]   http://www.agent-de-securite.be/biometrie/securite_biometrique.html

[11]   A.M. Bazen and S.H. Gerez, "Directional Field Computation for Fingerprints Based on the Principal Component Analysis of Local Gradients", in Proceedings of ProRISC2000, 11th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, Netherland, November 2009..

[12]   T. Aach, I. Stuke, C. Mota and E. Barth, "Estimation of Multiple Local Orientations in Image Signals", IEEE International Conference on Acoustics, Speech and Signal Processing, Quebec, 2010.

[13]   http://focus.ti.com/docs/toolsw/folders/print/tmdsdsk6416.html

AUTHOR

Farah Dhib Tatar

Specialty: Electrical Engineering (Embedded Systems)

Education:

o 2010: Engineering Degree from sfax College of Engineering, ENIS, Tunisia

o 2012: Master degree from Sfax College of engineering, ENIS, Tunisia

o In progress: Ph.D. degree from Tunis College of engineering, ENIT, Tunisia

Experience:

o 2010-2016: College of Higher Studies and Technologies ISET, Tunisia