A Complete System for Fingerprint Authentication using Delaunay Triangulation

Pedro Ribeiro Mendes Júnior, Antonio Carlos de Nazaré Júnior, David Menotti
Department of Computing
Federal University of Ouro Preto
pedrormjunior@gmail.com, acnazare@gmail.com, menottid@gmail.com

Abstract

The problem of recognizing a individual using fingerprints can be categorized into two types: authentication and identification. This paper presents an approach through Delaunay triangulation see if it uses two different fingerprints correspond to the same finger from the minutiae extracted from each fingerprint. The method showed an accuracy of 85%.

1. Introduction

A fingerprint is the design formed by papillae (elevations of the skin), present in the extremities of the fingers. The fingerprints are unique to each individual. This feature, called unity, causes the to be used as a means of identifying people for centuries [3].

Systems for Fingerprint Authentication are normally associated with criminal identification, however, currently also being used in commercial applications, such as access control and security in financial transactions. Another type of fingerprint application is confirmation of identity of candidate of selection process and concourses.

There are two types of applications for fingerprint recognition systems: authentication and identification. In an authentication system, the system input is a query to two fingerprints, which verifies that both are the same individual. In an identification system, the input is only a query with a fingerprint and the output is a short list of individuals who, according to a tolerance of error, possibly have that fingerprint.

A fingerprint is formed by a group of papillae (ridges). The most common characteristics of the ridge, called minutiae, are the end of lines, line ends, junctions and islands, as illustrated in Figure 1.

Usually the algorithms of fingerprints authentication have three steps:

1. **Preprocessing:** It is constituted by applying a series of image processing techniques in order to enhance the image of fingerprint, increasing the contrast between the ridges and background.

2. **Minutiae points extraction:** Extract the set of characteristics valid wich represents the fingerprint..

3. **Minutiae points matching:** Find the corresponding minutiae between the fingerprints in order to verify the similarity between them.

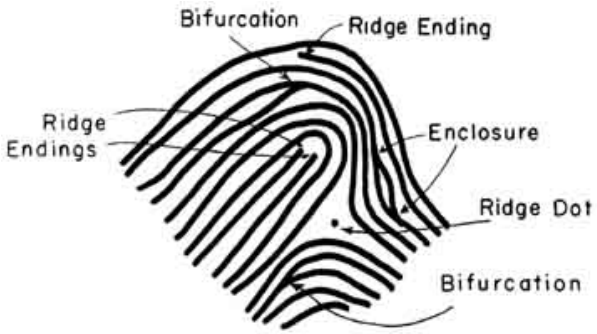


Figure 1. Examples of minutiaes

This paper proposes a method for fingerprints authentication that uses delauney triangulation, proposed in [7] to matching.

The remainder of this paper is divided as follows: Section 2 presents the techniques of image processing applied to the fingerprint image enhancement. Section 3 presents the extraction of minutiae. Section 4 presents the matching method used. The tests and results are discussed in Section 5. Finally Section 6 concludes.

2. Preprocessing

2.1. Image Enhancement

To have a enhancement on the image of fingerprint, is used a local algorithm of histogram equalization [4].

The image is divided into blocks of pixels of size $h \times w$ and for each block is counted the number of pixels at each

(a) Original Image. (b) Enhancement Image.

Figure 2. Example of Enhancement of Original Image for blocks with size 32×32.

intensity level (ie histogram) and calculated the new level of intensity for each of 256 levels, as Equation 1:

$$B_i = \left(\sum_{j=0}^{i} N_j \right) \times \frac{max\_intensity}{quant\_pixels}, i = 0, \ldots, 255 \quad (1)$$

where B_i is the new level of intensinty in the block, $max\_intensity$ is 255, $quant\_pixels$ equals to $h \times w$ and N_j is a pixel intensity level less than or equal to 1.

The image of Figure 2 shows a example this processing.

2.2. Segmentation

In this work, two segmentations are performed: the first (described in this Section) aims to find the "body"of the fingerprint and discard the background.

The second segmentation is performed along with extraction of minutiae(Section 3) and aims to discard false minutiae that are located on the edges of the "body"of the fingerprint (Figure 8(c)).

In this first segmentation, a binarization is performed in the original fingerprint image, the threshold determined by Otsu method [6]. Then, is performed a closing operation, on the negative image, so that the crests (value 1 in the negative image) of the fingerprint unite among themselves and fingerprint becomes a single object no "holes". In the resulting image, is applied to an open operation for small objects (smaller than the structuring element of the operation) are excluded.

From the result of morphological operations, we identify the bounding box that encompasses the entire object fingerprint. Thus, the segmented image (fingerprint) will be the region inside the bounding box outlined in the original image.

2.3. Gabor Filter

The Gabor filter is defined by a plane sinusoidal wave enveloped by a Gaussian (Figure 3).

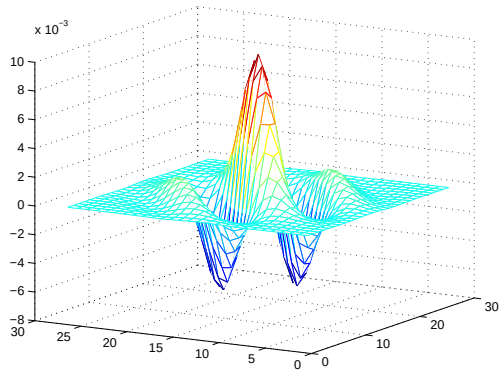


Figure 3. Representation of Gabor Filter with the parameters: $\theta = 3\pi/4$, $f = 1/8$, $\sigma_x = 4$ and $\sigma_y = 4$.

Since the orientations of ridges in the same fingerprint is variable, and the application of Gabor filter takes into account this guidance (in addition to the frequency of ridges), it is necessary to build a new filter for each region. This is accomplished by splitting the image into blocks and each block is calculated the orientation of the crests of the same frequency, the filter built and applied.

2.3.1 Calculation of Frequency and Orientation of Ridges

The method of calculating the orientation and frequency of the ridges described below is performed considering a block of size $h\ timesw$.

The orientation each block can be calculated by the method proposed by [8], shows in Equation 2:

A orientação de cada bloco pode ser calculada com

$$\Theta_B = \frac{1}{2} \text{atan} \left(\frac{\sum_{i=1}^{h} \sum_{j=1}^{w} 2 g_x(i,j) g_y(i,j)}{\sum_{i=1}^{h} \sum_{j=1}^{w} (g_x^2(i,j) - g_y^2(i,j))} \right) + \frac{\pi}{2} \quad (2)$$

onde g_x e g_y are the magnitudes of the gradient in the direction x and y and charge respectively may be calculated using the Sobel operator for example [4].

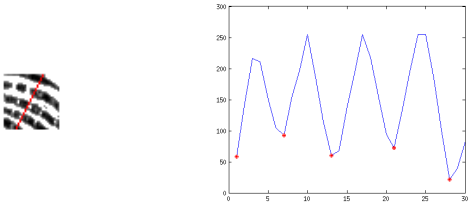
Figure 4 shows an example of acquisition with the application of the guidelines in blocks of size 16×16 and size 32×32. In these examples, the preprocessed images used in the acquisition of the guidelines were processed into blocks of same size as the blocks of the acquisition of the guidelines.

2.3.2 Frequencies of Crests

To calculate the frequency of ridges in a block, you need the information of the orientation of ridges in that block. It made a projection of the gray levels of the block in a

(a) Blocks of size 32×32 (b) Blocos of size 16×16

Figure 4. Examples of purchasing guidelines.



(a) Straight line orthogonal to the orientation of ridges. (b) Gray levels projection.

Figure 5. Acquisition of local frequency.

straight line through the center orthogonal to the orientation of ridges(Figure 5).

In this projection is applied an median filter with size 3 in order to eliminate false negative peaks, then, is desconsidered the value of the initial and final vector projection. From this projection is calculated the distance between negative peaks and then held the average distances achieved. The average achieved earlier is the wavelength, and thus the frequency is the inverse of this average (wavelength).

2.3.3 Filter

The construction of the Gabor filter is given by the Equation 3:

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right)\right) \cos\left(2\pi x_\theta f\right) \tag{3}$$

where,

$$x_\theta = x\cos(\theta) + y\sin(\theta)$$

and

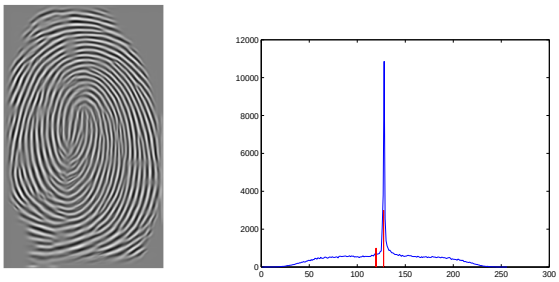
$$y_\theta = -x\sin(\theta) + y\cos(\theta)$$

The values of σ_x and σ_y are chosen based on the frequency [5]:

$$\sigma_x = \frac{k_x}{f}, \sigma_y = \frac{k_y}{f} \ ,$$

and the filter size is chosen based on the σ_x e σ_y values [5]:

$$w_x = 4.5\sigma_x, w_y = 4.5\sigma_y \ .$$



(a) Image after Gabor FilterApós o filtro de Gabor (b) Image Histogram

Figure 6. Threshold detection

2.4. Binarization

The filtered image using the Gabor filter contains two kinds of values differentiate themselves. The first region, darkest, refers to the ridge and the lightest region refers to the valleys (between ridges) as seen in Figure 6(a). In this image there is a large concentration of values that fall between the values of these two classes mentioned, which are refers to the background and other regions of indecision, as singularity areas (delta and loops for example) or flaws in the design of the fingerprint (Figure 6(b)).

Thus, the average is calculated (largest vertical straight of Figure 6(b)) and the variance, as the image after applying the Gabor filter is normalized between 0 and 1. The threshold T is calculated by the Equation 4

$$T = \mu - \sigma^2 \ , \tag{4}$$

where μ is the average and σ^2 is the variance of the pixels intensity of imagem after applying the Gabor Filter. In the histogram of Figure 6(b), the smallest vertical straight represents this threshold.

2.5. Thinning

After binarization (Figure 7(a)), is conducted a thinning in the crests of the image fingerprint so that the minutiae extraction step can better detect endings and bifurcations. Thus, the ridges have 1 pixel of width. The thinning (Figure 7(b)) is accomplished using morphological operations [6].

3. Minutiaes Extraction

After enhancement, the image of the fingerprint is used for the extraction of minutiae points. There are several characteristics that can be used to authenticate the fingerprint, but most of the minutiae are restricted to only two types: The bifurcations and the line ends, call terminations.

A termination is the point where a line (ridge) ends and junctions are points where the ridge is divided in a simple way to join a 'Y'.

(a) Binarization. (b) Thinning.

Figure 7. Binarization and thinning.

3.1. Extraction of All Minutiae

The points of termination and bifurcation are extracted from the thinned image (Figure 7) of fingerprint with the aid of the concept of Condition Number (c_n) [1]. The Equation 5 calculates the condition number for a pixel belonging to the ridge of the fingerprint.

$$C_N = \frac{\sum_{k=1}^{8} |\Gamma(k+1) - \Gamma(k)|}{2} \text{ onde } \Gamma(9) = \Gamma(1). \quad (5)$$

where,

$$\Gamma(p) = \begin{cases} 1 \text{ if } p \text{ belongs to ridge} \\ 0 \text{ else} \end{cases}$$

and k represents the eight neighbors of p ordered in a clockwise direction

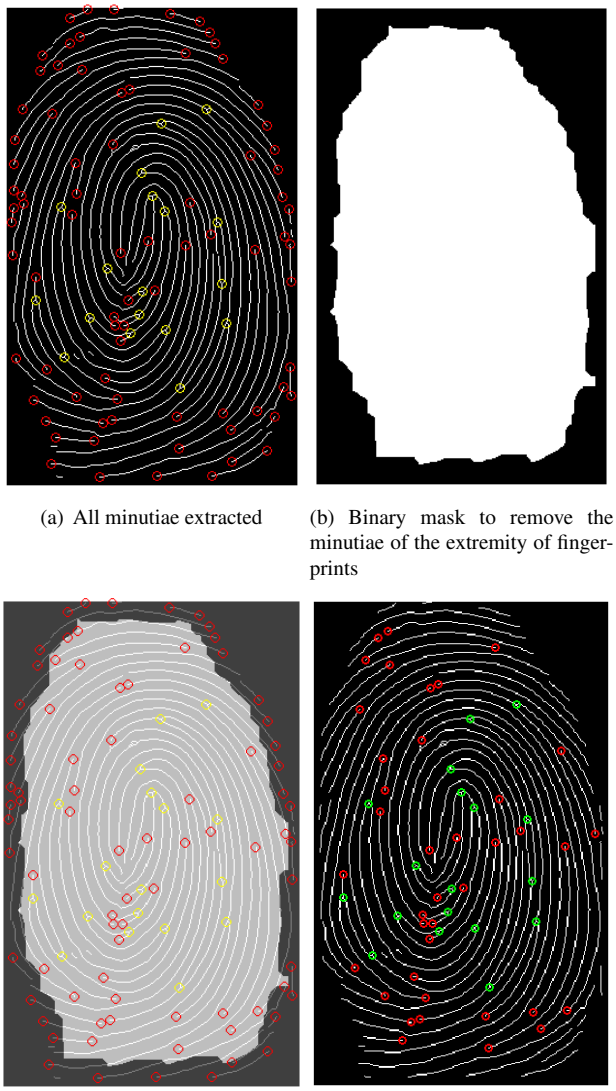
If $c_n(p)$ is equal to 1, p is an termination point. For p be a bifurcation, $c_n(p)$ should be equal to 3. All other values of c_n are ignored. At the end have been a two-dimensional array ($MIN_{i,j}$ with the same dimensions i and j of the thinned image containing the values 1 and 3 for points minutiae and the remaining values equal to 0.

The Figure 8(a) shows all minutiae extracted from the fingerprint of Figure 7.

3.2. Spurious Minutiae Filtering

After extraction of the minutiae, they pass through a filter in order to detect spurious minutiae and remove them. These minutiae are often found in samples of fingerprint due to the presence of noise in the other stages of image processing and decrease the accuracy and performance of the authentication of fingerprints.

The first part of this filtering is deleting all minutiae belonging to the extremity of fingerprint. To do so creates a binary mask (MB) from the thinned image by performing morphological operations on the same [2], this mask



(a) All minutiae extracted (b) Binary mask to remove the minutiae of the extremity of fingerprints

(c) Mask applied to the array of minutiae points (d) Minutiae image after the spurious minutiae filtering

Figure 8. Example of spurious minutiae filtering

is shown in Figure 8(b) will be considered true only those minutiae that intercede with the mask. This can be expressed by the following Equation 6:

$$Min'(i,j) = Min(i,j) \times MB(i,j) \quad (6)$$

where $Min'(i,j)$ is the new matrix of minutiae points. The figure 8(c) shows the result of this operation.

The second part of spurious minutiae filtering consists in the analisys of the flow of ridges, well as the distance and connectivity of the minutiae. The algorithm used is described in [2].

The final result of spurious minutiae filtering is show in Figure 8(d).

4. Fingerprint Matching Using Delaunay Triangulation

For fingerprint matching, was used the method proposed in [7]. The following sections explain the method, wich uses the Delaunay Triangulation.

4.1. Fingerprint matching using minutiae triangles

The comparison of minutiae triangles is a common approach to matching a pair of minutiaes sets. For this task it is necessary are forming triangles from minutiae triples and matching them using invariant features. The Figure 9 shows an example. In general, a pair of minutiae triangles provides enough information to compute a transformation that potentially aligns the minutiae sets. To compute good alignments, voting technic is applied in the transformation space to find transformation that are supported by many minutiae triangles. A number of hypothetical transformations is then found by considering transformations that have received high number of votes. Each hypothetical transformation is explicitly verified by using it to align the minutiae sets and counting the number of overlapping minutiae. The best alignment is the one maximizing the number of overlapping minutiae.

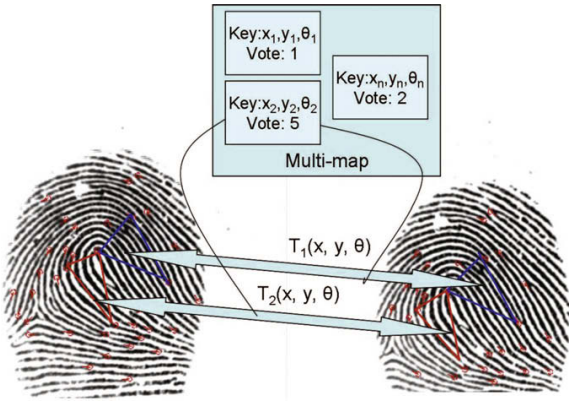
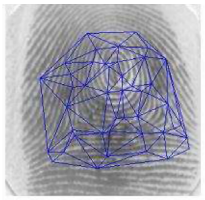


Figure 9. Matching by comparing minutiae triangles. Good alignment are computed by using voting in the transformation space to find transformations that are supported by many matching triangles.

The computational complexity to form the minutia triangles is $O(n^3)$. To keep complexity low, [7] proposes associating a unique topological structure with the minutiae using Delaunay triangulation and using the Delaunay triangles for matching. This reduces the number of minutiae triangles to $O(n)$, speeding up matching considerably without affecting accuracy significantly. The Figure 10 shows the Delaunay triangulation of a set of minutiae, overlaid on the corresponding fingerprint image.



Minutiae

Id	*x*	*y*	*D (degree)*
1	200	47	42
2	125	49	117
3	173	69	160
4	219	71	60
5	139	74	309
...	...	...	...

Triangular Features

l_1/l_3	l_2/l_3	*Cos(A)*	m_1	m_2	m_3
0.47	0.69	-0.47	2	1	3
0.55	0.66	-0.36	3	2	5
0.66	0.76	0.01	3	4	1
0.32	0.96	0.03	3	4	6
0.76	0.98	0.36	3	5	11
...	...	...	...	...	...

Figure 10. An example of minutiae features extracted by algorithm proposed by [7].

4.2. Invariant Features

Once the Delaunay triangulation of a set of minutiae has been computed, is considered two groups of features from each triangle based on the sides and angles of the triangle. Specifically, given a minutiae triangle (Figure 11), the first group of features, denoted as V_t , includes three attributes which are invariant to rigid transformations:

$$V_t = \left[\frac{l1}{l3}, \frac{l2}{l3}, cos(A)\right] \tag{7}$$

where $l1 \leq l2 \leq l3$ and A is the largest angle of the triangle (*i.e.*, the angle across from the largest side). In [7] was used the cosine of A instead of A itself because the cosine is less sensitive to noise. The second group of features, denoted as V_m, involves the angles of the minutiae. Specifically, we order the minutiae forming the triangle with respect to the length of the side across them and take their angles as the second set of features:

$$V_m = [\angle m1, \angle m2, \angle m3] \tag{8}$$

Very large angles produces triangles whose points are almost collinear. Such triangles are undesirable since small errors in minutiae locations can lead to large errors in the computation of the parameters of the alignment transformation. Although the Delaunay triangulation tends to avoid skinny triangles, it is not always guaranteed unless inserting extra points. To deal with this issue, [7] reject triangles whose largest angle is greater than a threshold (e.g., 168 deg). Figure 10 shows an example of minutiae features extracted by algorithm proposed by [7].

4.3. Hypothesis generation

This step aims generating a number of hypothetical alignments between minutiae sets. This is performed by finding corresponding minutiae triangles using the invariant attributes. Specifically, let T and Q correspond to two different minutiae templates. Then, triangles in T are compared to triangles in Q using the following three constraints:

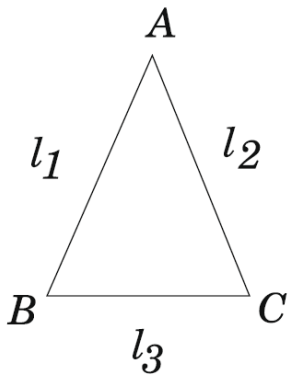


Figure 11. A triangle defined by a triplet of minutiae.

1. *Similarity consistency:* Tests the similarity between twon minutiae triangles. As show in Figure 10, each triangle is represented by six invariatns. If the differences between corresponding pairs of invariants are all below a threshold, then is considered that the triangles match. In the experiments performed by [7], the thresholds used for the spatial and angular features were 0.3 and 0.5, respectively.

2. *Planarity consistency:* This constraint tests whether matching minutiae triangles can be brought into alignment using inplane transformations only. The Figure 12 illustrates this criterion with an example. If we order the minutiae in Figure 12 starting from the first one in each triangle and going clockwise, the ordering in the left triangle would be $m_{11}m_{12}m_{13}$ while the ordering in the right triangle would be $m_{21}m_{23}m_{22}$. Obviously, an out-of-plane transformation is required in order to align the triangles in this case. Such inconsistencies can be fixed by changing the order of the minutiae in the left triangle (*e.g.*, starting from m_{12}). Then, the similarity consistency between the two triangles can be tested using the new ordering. We consider all three possible orders.

3. *Minutiae orientation consistency*This constraint test whether corresponding minutiae have similar orientations. This is performed by estimating the rigide transformation that aligns corresponding triangles and computing the orientation differences of the corresponding minutiae.

If all constraints are satisfied for a given pair of minutiae triangles, then we concluded that the minutiae triangles match. Given a pair of matching triangles, a rigid transformation can be computed which aligns them. These, "locally optimum", transformations are used to substantiate a number of hypothetical alignment transformations between the minutiae sets which are supported by many corresponding triangles. To find these, "globally consistent", transformations we employ a voting scheme where each matching pair of triangles casts a weighted vote in the transformation space. The weight of a vote is inversely proportional to the average minutiae orientation differences. To compensate for quantization errors in the transformation space, we also cast votes to the immediate neighbors of the estimated transformation using lower weights. Alignment transformations that receive high votes are considered for further verification.

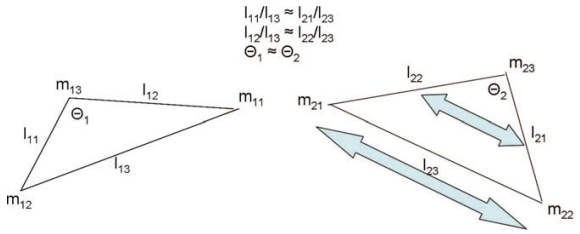


Figure 12. An example illustrating an inconsistency in the ordering of the minutiae between triangles.

The Figure 13 shows the structure of the entries in the transformation space. Each entry holds (1) the transformation parameters, (2) the number of votes, and (3) a list of corresponding minutiae that have voted for this transformation. It should be mentioned that the transformation space needs to be quantized coarsely enough to let the entries receive enough votes and build reasonable histograms. Since we align the minutiae triangles using rigid transformations, the transformation space is three-dimensional (*i.e.*, x, y, and h). However, since we use $cos(\theta)$ and $sin(\theta)$ in our calculations frequently, we decided to store this information instead of the actual angle for efficiency.

X	Y	cos θ	sin θ	Vote	p_{11}	p_{12}	...	p_{1n}
					p_{21}	p_{22}		p_{2n}

Figure 13. Structure of entries in the transformation space.

After all the votes have been accumulated, local maxima in the transformation space are detected and considered as possible candidates for aligning the minutiae sets. The resulting set of transformations yields a set of hypotheses which are verified in the next stage.

4.4. Hypothesis verification

In this stage, the minutiae templates are aligned using the hypothetical transformations in order to determine the highest number of overlapping minutiae. First, each candidate transformation is refined (*i.e.*, re-estimated) using all matching triangles that have voted for that transformation. Unlike local transformation computations, which are based only on a pair of minutiae triangles, the refinement process computes globally consistent transformations by consider-

ing minutiae correspondences scattered over a diverse region of the fingerprint. Then, the quality of each hypothesis is evaluated by aligning the minutiae sets and computing the number of overlapping minutiae.

The overlap between the minutiae sets is determined by considering the differences between corresponding minutiae locations and orientation angles. The number of overlapping minutiae is normalized to calculate a similarity score between the minutiae sets. Specifically, let t and q be the number of minutiae in the two sets, respectively, if m is the number of matching minutiae, then the similarity score s is calculated as in Equation 9:

$$s = \frac{2m}{t+q} \times 100 \tag{9}$$

The hypothesis yielding the highest number of overlapping minutiae is taken as the best hypothesis. It should be mentioned that we tried several other ways to compute similarity scores, however, the formula above gave the best results.

5. Validation of The Method and Results

The implementation of the algorithm was done in MATLAB. In the experiments, it was used a image database of 408 images. This database has images of 51 different individuals (8 imagem from each individual).

In the first set to validation, it was selected randomly 2 images (from 2 different individuals) to compare each one to each other 50 fingers. The objetive of this set is to obtain the false positive (FP) rate. In the second, it was selected randomly 14 images (from 14 different individuals) to compare each one with all 7 of the same individual. The objetive of this set is to obtain the false negative (FN) rate.

Thus, it was created a confusion matrix where it is presented the following rates (see Table 1).

- **TP - true positive:** Correct authentication of two fingerprints of the same individual.
- **TN - true negative:** Correct rejection of two fingerprints of different individuals.
- **FP - false positive:** Incorrect rejection of two fingerprints of the same individual.
- **FN - false negative:** Incorrect authentication of two fingerprints of different individuals.

	P	N
T	69 (70%)	100 (100%)
F	29 (30%)	0 (0%)

Table 1. Confusion matrix denoting the TP, TN, FP, and FN

From the confusion matrix can calculate the accuracy (AC) and precision (P) of the method by Equations 10 and 11, respectively.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{169}{198} = 85\% \tag{10}$$

$$AC = \frac{TP}{TP + FP} = \frac{69}{69 + 29} = 70\% \tag{11}$$

6. Conclusion and Future Works

It can be concluded that the use of Delaunay triangulation in stage of matching for fingerprint authentication presented satisfactory results. As future work we propose an empirical study to define parameters, such as threshold for authentication decision. It also proposes to reduce the complexity of using the hierarchy of Delaunay.

References

[1] J. C. Amengual, J. C. Prez, F. Prat, and J. M. Vilar. Realtime minutiae extraction in fingerprint images. In *Proceedings of the 6th International Conference on image processing and its application*, pages 871–875, 1997.

[2] S. Kim, D. Lee, and J. Kim. Algorithm for detection and elimination of false minutiae in fingerprint images. In J. Bigun and F. Smeraldi, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 235–240. Springer Berlin / Heidelberg, 2001.

[3] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, 2 edition, 2009.

[4] R. Miron and T. Letia. Fuzzy logic decision in partial fingerprint recognition. In *IEEE International Conference on Automation, Quality and Testing, Robotics*, volume 3, pages 1–6, 2010.

[5] R. Thai. Fingerprint image enhancement and minutiae extractions. Technical report, 2003.

[6] The MathWorks. Image processing toolbox user's guide, 2010.

[7] T. Uz, G. Bebis, A. Erol, and S. Prabhakar. Minutiae-based template synthesis and matching for fingerprint authentication. *Computer Vision and Image Understanding*, 113(9):979–992, 2009.

[8] Y. Wang, J. Hu, and F. Han. Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields. *Applied Mathematics and Computations*, 185:823–833, 2007.