UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

LOW COST FINGERPRINT MATCHING SYSTEM FOR BIOMETRIC IDENTIFICATION

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Ferran José Torra

In partial fulfilment

of the requirements for the degree in

Ciències I Tecnologies de Telecomunicació

Advisor: Josep Vidal Manzano

Barcelona, June 2015

Abstract

The objective of this project is to develop a system required by the Doctors Without Borders NGO that consists in managing the medical histories of a refugee camp in an easier way. Specifically, this system was already designed in another final degree project with the objective of storing the medical histories in a database, managing them by an application, and using the fingerprint for the patient's identification. Due to the poor results obtained in the fingerprint identification, this project is focused in developing a low complexity technique (that should run in an Android device) with low mismatching error. Two minutiae-based matching methods have been developed but only one of them reaches sufficiently good results. It is based in the vectors between the minutiae in the frequency domain and it has reached a correct recognition probability of 70% and a correct rejection probability of 71.4286% with a processing time of 5 seconds in a database compounded by 60 people.

Resum

L'objectiu d'aquest projecte és desenvolupar un sistema sol·licitat per la ONG Metges Sense Fronteres que consisteix en gestionar més fàcilment els historials mèdics d'un camp de refugiats. En concret, aquest sistema va ser dissenyat en un altre treball fi de grau i consisteix en guardar els historials mèdics en una base de dades, gestionar-els a través d'un aplicació i utilitzar l'empremta dactilar per la identificació dels pacients. Degut als pobres resultats obtinguts en la identificació amb l'empremta dactilar, aquest projecte està centrat en millorar aquesta part. S'han desenvolupat dos mètodes de matching basats en minúcies però només un d'ells assoleix resultats suficientment bons. Aquest mètode es basa en els vectors que uneixen les minúcies en el domini freqüencial i ha assolit una probabilitat de reconèixer correctament del 70% i una probabilitat de rebutjar correctament del 71.4286% amb un temps de processat de 5 segons en una base de dades formada per 60 persones.

Resumen

El objetivo de este proyecto es desarrollar un sistema solicitado para la ONG Médicos Sin Fronteras que consiste en gestionar más fácilmente los historiales médicos de un campo de refugiados. En concreto este sistema fue diseñado en otro trabajo fin de grado y consiste en guardar los historiales médicos en una base de datos, gestionarlos a través de una aplicación y utilizar la huella dactilar para la identificación de los pacientes. Debido a los pobres resultados obtenidos en la identificación con la huella dactilar, este proyecto está centrado en mejorar esta parte. Se han desarrollado dos métodos de matching basados en minucias pero solo una de ellos logra resultados suficientemente buenos. Este método se basa en los vectores que unen las minucias en el dominio frecuencial y ha logrado una probabilidad de reconocimiento correcto del 70% y una probabilidad de rechazar correctamente del 71.4286% con un tiempo de procesado de 5 segundos en una base de datos formada por 60 personas.

I dedicate this project to my parents, Isidre and Teresita, sisters, Anna and Gemma, brothers-in-law, Jordi and Ritxi, and my grandmother, Josefa, who have always trusted me and have helped me to reach this point providing both moral and economic support. I also dedicate it to my aunts, Josep and Maria Teresa, who have taken me to Barcelona during all the degree.

Thanks.

Acknowledgements

I specially acknowledge the advices and supervision of my project supervisor, without him the development of this project had not been possible.

I also acknowledge to all the people that have contributed in the process of the capturing fingerprints and Joan Morales, who helped me to develop the user interface of the final android application.

Revision history and approval record

Revision	Date	Purpose
0	16/06/2015	Document creation
1	28/06/2015	Document revision
2	05/07/2015	Document revision
3	08/07/2015	Document revision
4	10/07/2015	Document finalization

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Ferran José Torra	fjosetorra@gmail.com
Josep Vidal Manzano	josep.vidal@upc.edu

Written by:		Reviewed and approved by:	
Date	10/07/2015	Date	10/07/2015
Name	Ferran José	Name	Josep Vidal
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Statement of purpose

The project is carried out at the University Association Students for cooperation (AUCOOP) from Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB). This association launches projects from the technological field in developing countries both on local and international level.

The project has been required by an NGO, Doctors Without Borders (MSF) and its goal is to efficiently manage the medical histories of a refugee camp with the help of latest information and communication technologies. Specifically, this refugee camp is in Ethiopia and its refugees come from war zones in Somalia. This work was already started by Irene Baeza in her final grade project: SOLUCIONES TIC PARA LA CREACIÓN DE SISTEMAS DE SEGUIMIENTO DE HISTORIALES CLÍNICOS EN CAMPOS DE REFUGIADOS [1]. Our project is the continuation of [1] and, as a consequence, the proposed system complements her work.

The system is able to store the medical histories in a database and manage them by a specific application. As a result, the medical histories are always perfectly organized and, as the system is supported by an application running on mobile devices, the medical histories are not hampered by the problems associated to storage in paper (space, fragility, etc.). Regarding to patient's identification, the proposed system has to be very different from the current one. Nowadays, the system only requires the name of the patient, which is not a reliable solution as, for instance, some patients change their names and visit MSF more than once in order to get more rewards from doctors. As a solution, a biometric identification has to be implemented. Fingerprint identification guarantees high reliability and acceptance by users.

In order to set the scope of this project, it is important to analyse the milestones that were reached in [1]. Concerning the medical histories management, a mobile phone application called AplicaciónTest1 using the CommCare framework was designed. Although it is a demo version, and it does not incorporate communication with the server (where the database is stored) this application is very complete: apart from generic features like registration, it also has other features. For instance, it is possible to create groups of doctors that share information of patients, and allow different doctors visiting a patient to quickly access the patient medical history.

However, in the field of the fingerprint identification, the progress of Baeza's work was limited. The fingerprint identification was identified as the most appropriate identification system and this is the reason why it is used in this project. However, the matching methods were poor. Therefore, we aim at developing an enhanced matching method and maintain the rest of the system (android application, usage of a computer acting as a server in order to store the database, usage of tablets for the doctors, etc.).

In order to develop this matching method, some information has been extracted from another final grade project called "IDENTIFICACIÓ D'EMPREMTES DIGITALS" [2] authored by Núria Esquís. Specifically, what is borrowed from this project is the minutia extraction, a procedure well established in the literature that explained in section 3.2. Moreover, as [2] reaches quite good results in matching, her method is compared with the matching method developed here and, as a consequence, the details of the method are explained in the section 2 and it is evaluated in the section 4. Furthermore, as all the matching methods are implemented in Matlab but in the final system they have to be implemented in the Android platform, an android application plugged to a fingerprint reader is also developed to test the behaviour of the matching method in the Android platform. As its purpose is basically for testing the matching method, it does not include neither the part of communication with the server nor the part of the medical histories expounded in [1]. Its specific functioning is explained in the section 5.2 of this thesis. However, as it is described in the incidences, it has not been possible de develop a complete version of this application, only a basic user interface.

1.2. Requirements and specifications

Project requirements:

- Fast user identification and user verification by fingerprinting of one or more fingers.
- A clear user interface in order to facilitate the application management.
- Application integrated in an Android tablet.
- Adopt the minutiae extraction code already available.
- The cost of the final product should be low.

Project specifications:

- The fingerprint identification has to accomplish a certain maximum percentage of false recognition range (FRR) and false acceptance range (FAR). FRR is defined as the rate of incorrect rejections of registered users (their fingerprints are stored in the data base). FAR is defined as the rate of incorrect acceptance of unregistered users (their fingerprints are not stored in the data base).
- Evaluation requires a training/test database of size enough to obtain the required percentages mentioned above with a high accuracy.
- The fingerprint identification has to accomplish a certain maximum percentage of wrong identification. It occurs when a subject is mistaken by another one.

1.3. Work plan

1.3.1. Incidences

During the project lifetime some incidences has occurred and, as a consequence, the work plan has changed.

Firstly, the pruning of the database has finally been removed. In the beginning of the project, this database was the database used in [2] and in the critical review it was moved from the beginning of the time plan to the finalization of the minutiae matching methods development. However, in order to achieve results closer to those achieved in the refugee camp, the final database, called training database in this thesis, has been built using the sensor that will be used in the field. Therefore, as the fingerprint capturing can be repeated, all the stored fingerprints have enough quality. As a consequence, another task called "development of a training database" has been added in the evaluation field, which starts after the purchase of the fingerprint sensor.

Secondly, the development of the minutiae matching methods has taken more time than expected as the methods developed both in the initial planned date and in the critical's planned day had not reached good results. It entailed both a delay and a reduction of the evaluation and selection of the matching methods.

And thirdly, the rest of the incidences are related to the Android implementation. Concerning the Android search, it was carried out almost at the same time that the development of the interface of the final application. As Android is very extensive and only some areas were needed, the used method consisted in search for only certain information when it was needed. For instance, how to change the screen when the user presses a button. In reference to the fingerprint sensor's identification, it has been more difficult than it seemed because of three main reasons: 1) only few fingerprint sensors support the Android platform, 2) the Android device also has to accomplish some features, and 3) the drivers are not available in the Internet and it was necessary to contact with the companies to ensure the compatibility of the sensor with a specific Android device. Finally, regarding the development of a basic database in the Android platform, the development of both the minutiae matching extraction and the fingerprint matching technique in the android platform, and the testing of the final application, they have not been done due to the lack of time associated to both the minutiae matching methods and the identification of the fingerprint sensor.

The impact in the work plan of all these incidences related with the android implementation is big. On the one hand, there is the modification of the task "development of a user interface & basic database" into the task "development of a user interface", which has been advanced in order to show the parallelism between this task and the task "Android search and learn", which has been expanded. On the other hand, there are the elimination of the tasks "Develop minutiae extraction & fingerprint matching technique" and "Beta test application", and the expansion of the task "Identification of fingerprint sensor". Moreover, in reference of the project's requirements, the third, which is "Application integrated in an android tablet", has not been fulfilled.

1.3.2. Work Breakdown structure

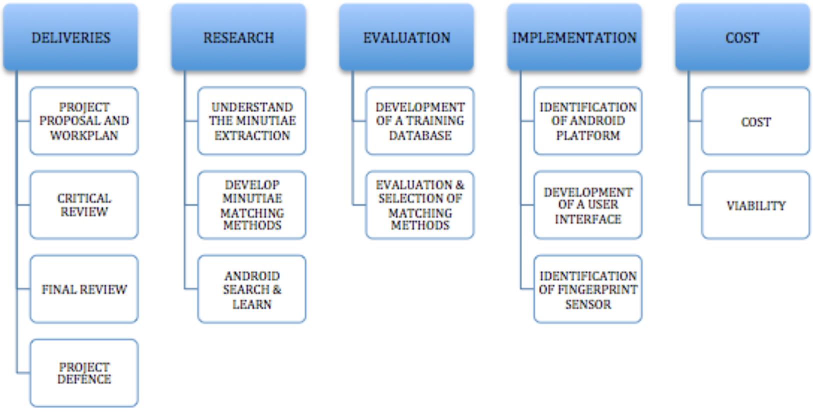

Figure 1 – Work breakdown structure

1.3.3. Work packages, tasks and milestones

Work Packages:

Project: DELIVERIES	WP ref: (WP#)	
Major constituent: Documentation	Sheet 1 of 5	
Short description: This package is compounded by all the information that I have to deliver during the entire project. This information can be both documents and oral presentation.	Planned start date: 23/02/2015 Planned end date: 24/07/2015 Start event: 23/02/2015 End event: 24/07/2015	
Internal task T1: Deliver a document about the Project proposal and the work plan. Internal task T2: Deliver a document in the middle of the term in order to see the progress and the delays. Internal task T3: Deliver the final document of the entire project. Internal task T4: Oral project defence with slides, which have to be delivered too.	Deliverables: Documents and slides.	Dates: T1: 23/02/2015 06/03/2015 T2: 13/04/2015 24/04/2015 T3: 15/06/2015 10/07/2015 T4: 13/07/2015 24/07/2015

Project: RESEARCH	WP ref: (WP#)	
Major constituent: Investigation	Sheet 2 of 5	
Short description: This package is constituted for all the things whose knowledge for my part is lower or null.	Planned start date: 02/03/2015 Planned end date: 01/05/2015 Start event: 02/03/2015 End event: 19/06/2015	
Internal task T1: Understand minutiae extraction code perfectly before start new code. Internal task T2: Detect and identify a fingerprint from a database with minutiae matching methods. This task will be carried out by different methods in order to find the best solution in front of the typical fingerprint identification problems (loss of information, translation, etc). Internal task T3: Get a basic knowledge of android language, as it is necessary to carry out the fingerprint identification as an android application.	Deliverables: Code	Dates: T1: 02/03/2015 13/03/2015 T2: 16/03/2015 19/06/2015 T3: 09/03/2015 27/03/2015

Project: EVALUATION	WP ref: (WP#)	
Major constituent: Software	Sheet 3 of 5	
Short description: This package contains all the parts of the project that requires an evaluation before their implementation.	Planned start date: 09/03/2015 Planned end date: 29/05/2015 Start event: 27/04/2015 End event: 03/07/2015	
Internal task T1: Build a training database with the sensor in order to achieve results closer to those achieved in the refugee camp. Internal task T2: Evaluate the different minutiae matching methods and find the best one	Deliverables: Code	Dates: T1: 27/04/2015 01/05/2015 T2: 22/06/2015 03/07/2015

Project: IMPLEMENTATION	WP ref: (WP#)	
Major constituent: Software	Sheet 4 of 5	
Short description: This package includes all the tasks related to the development of the final android application.	Planned start date: 09/03/2015 Planned end date: 03/07/2015 Start event: 09/03/2015 End event: 24/04/2015	
Internal task T1: Identify an Android platform to programme the final Android application. Internal task T2: Develop the user interface of the android application. Internal task T3: Search and buy an appropriate fingerprint sensor in order to identify a fingerprint in real time. Identify both its drivers and its user interface in Android application, and capture/analyse fingerprints.	Deliverables: Code	Dates: T1: 09/03/2015 13/03/2015 T2: 16/03/2015 27/03/2015 T3: 09/03/2015 24/04/2015

Project: COSTS AND VIABILITY	WP ref: (WP#)	
Major constituent: Research	Sheet 5 of 5	
Short description: This package contains the costs and viability of the entire infrastructure.	Planned start date: 22/06/2015 Planned end date: 03/07/2015 Start event: 22/06/2015 End event: 03/07/2015	
Internal task T1: Calculate the costs, which are the price and the maintenance both of a device that supports android applications and of a computer that stores de database. Internal task T2: Study the viability of the infrastructure in the refugee camp.	Deliverables: Documents	Dates: T1: 22/06/2015 03/07/2015 T2: 22/06/2015 03/07/2015

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
DL	T1	Project proposal and workplan	Document	2
DL	T2	Critical review	Document	9
DL	T3	Final review	Document	20
DL	T4	Project defence	Slides	22
RS	T1	Understand minutiae extraction	-	3
RS	T2	Develop minutiae matching methods	Code	17
RS	T3	Android search & learn	Knowledge	5
EV	T1	Development of a training database	-	10
EV	T2	Evaluation & selection of matching methods	Code	19
IM	T1	Identification of Android platform	-	3
IM	T2	Development of a user interface	-	5
IM	T3	Identification of fingerprint sensor	-	9
CV	T1	Costs	Document	19
CV	T2	Viability	Document	19

Table 1 – Milestones

1.3.4. Time plan

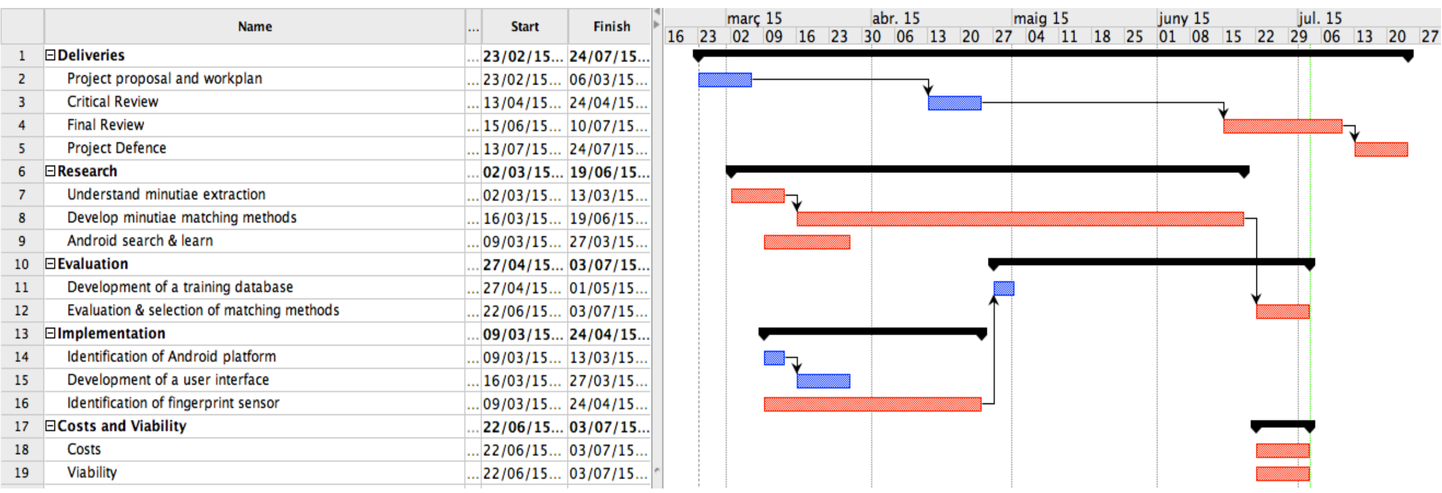

Figure 2 – Time plan

2. State of the art of the technology used or applied in this thesis

2.1. Existing matching methods

The existing matching methods can be divided in three families [4]:

- Correlation-based matching: two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments.
- Minutia-based methods: minutiae are extracted from the two fingerprints and stored as sets of points. Then, the number of minutiae pairings between both fingerprints is computed.
- Ridge feature-based methods: two fingerprints are compared in terms of features extracted from the ridges such as ridge shape, orientation and frequency.

The most popular and widely used technique is the minutiae-based. This is the technique implemented in this project and, therefore, this point is focused in this technique. In particular, two minutiae-based matching are explained. They are the matching method used in [2] mentioned in the first point of the thesis, and the on-line fingerprint verification made by Anil Jain, Lin Hong and Ruud Bolle [3]. Both are really related, as the first one is a simplification of the second one.

The reason of the presence of the first case in this point is because it could be compared in the point 4 of this thesis, with this project, as both use the same minutiae extraction (explained in the Methodology/ project development point). However, the matching method is totally different.

It is divided in two steps: minutia alignment and minutia comparison. The first step consists in find a suitable reference minutia both for the input fingerprint and the database's fingerprint. Then, the minutiae are aligned around their reference minutia taking the translation and rotation into account. The search of the suitable reference minutia for each fingerprint resides in the scalar product between the ridges of both fingerprints. This scalar product has to be higher than 0.8, and in this case the minutia of each selected ridge is designated as the suitable reference minutia of its fingerprint. Then, the rest of minutiae are aligned around the reference minutia with the following expression where θ_i is the ridge angle:

$$\begin{pmatrix} x_i^{alin} \\ y_i^{alin} \\ \theta_i^{alin} \end{pmatrix} = \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ -\sin\Delta\theta & \cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i - x^{ref} \\ y_i - y^{ref} \\ \theta_i - \theta^{ref} \end{pmatrix} \quad (1)$$

The second step consists in create a box around each aligned minutia of the database's fingerprint, and then, compute the percentage of aligned minutiae of the input fingerprint that are inside a box of the database's fingerprint.

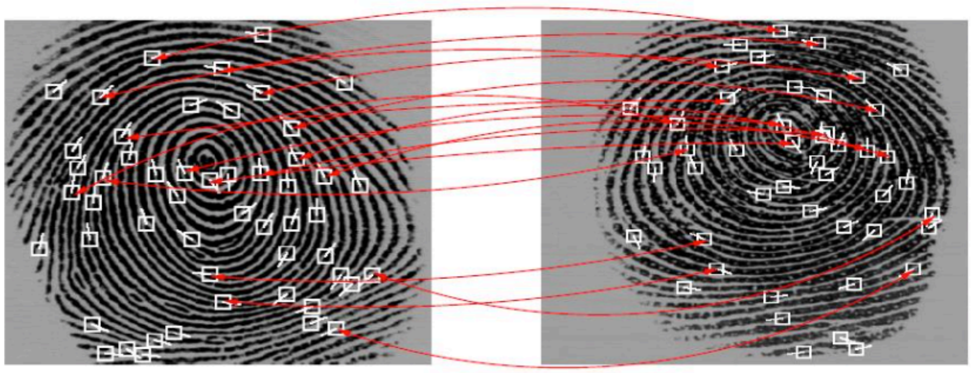

Figure 3 – Minutiae comparison example

The computed percentage is used as a parameter to define the similarity between both fingerprints, and when it reaches a value higher than 50, both fingerprints are identified as the same fingerprint.

In reference of the other matching, the only differences with the matching explained above, are the shape of the boxes and the minutiae coordinate system. In this matching the boxes are adaptive and the minutiae are in the polar coordinate system in order to compensate the nonlinear deformations. This type of deformation starts from a certain point and nonlinearly radiates outward producing a radial property in the polar coordinate. Moreover, the formulas used to adapt the boxes are not trivial, thus producing a complex but robust method.

Nevertheless, all the techniques used in the second case to cover the problem of the nonlinear deformations are not needed in this project due to its context. The on-line fingerprint verification made by Anil Jain, Lin Hong and Ruud Bolle was designed to could be implemented in the forensic field, where the fingerprint captures can not be repeated as the individual is not present. However, this project has to be implemented in a refugee camp, where the captures of the fingerprints can be repeated in the case of problems such as nonlinear deformations. Moreover, as the population of the refugee camp is big, the implemented system has to be very fast, which is easier to reach with a more simply matching method.

Therefore, the matching method used in [2], can be implemented in the context of this project. Hence, it will be compared in the point 4 with the matching developed in this project using the same database and the same input fingerprints.

3. Methodology / project development

Although some methods and models have been tested during the project, only two of them have reached quite good results. Nevertheless, all of them follow the general matching structure that can be seen in the following section.

3.1. Functional diagram

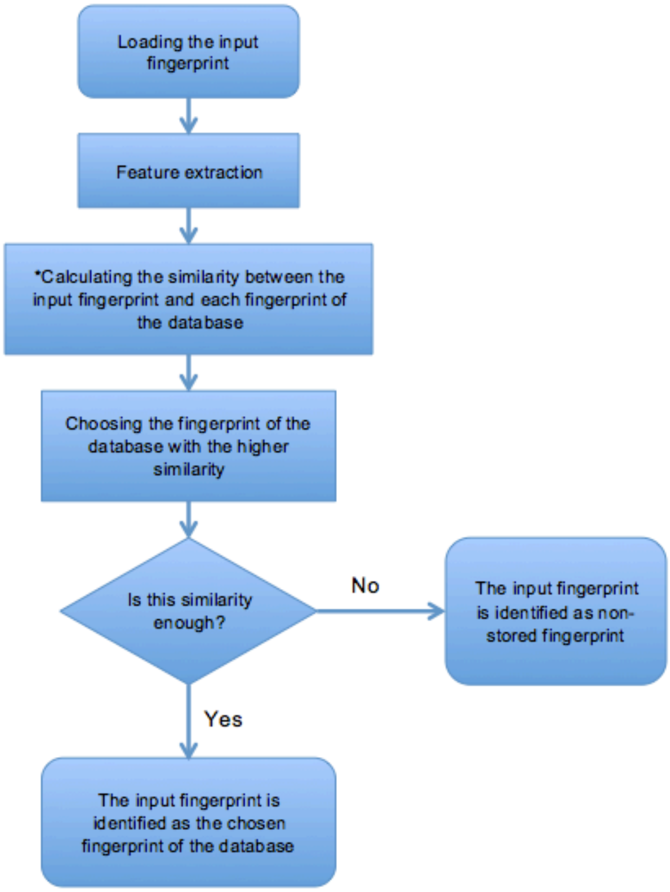

Figure 4 – Flow diagram of matching recognition procedure

*The parameters that model the similarity mentioned in this task, depends on the method and model used, and they are explained in the section 3.3.

3.2. Feature extraction

The fingerprint matching is performed using minutiae, which are the features used for recognition.

3.2.1. Minutiae definition

Minutiae are points related with particular shapes of the fingerprint ridges. They are divided in two types: junctions and ends. Junctions are the points where two ridges are joined, and ends are the points where the ridges end.

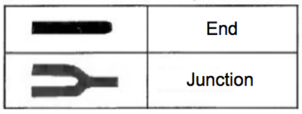

Figure 5 – End and junction shapes

As it has been told in the first section of this thesis, the extraction of the minutiae is a well-established procedure, which is described below. This extraction consists in a previous step where the fingerprint is enhanced, and a final step where the minutiae are extracted as ends and junctions. The first step uses the Hong et algorithm combined both with other techniques and with information extracted from http://www.csse.uwa.edu.au/~pk/research/matlabfns/. The second step is based in the Jain and Ratha et algorithm.

The results from each step can bee seen in the following figures:

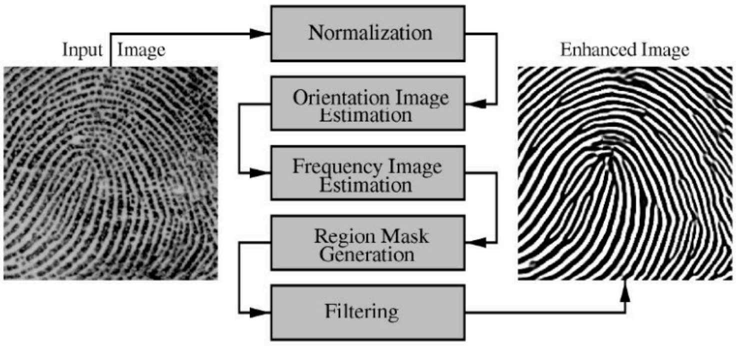

Figure 6 – Fingerprint enhancement step

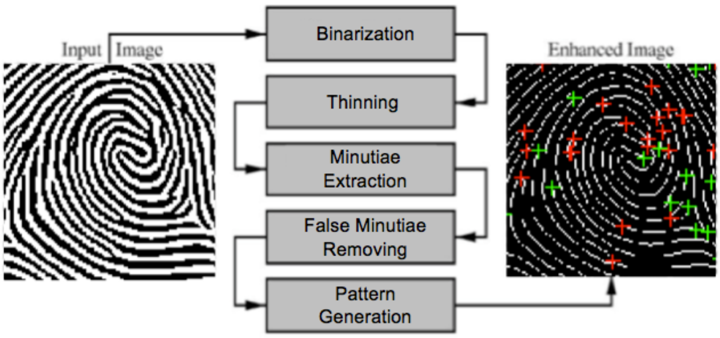

Figure 7 – Minutiae extraction step

However, some modifications have been done in order to fit better with the matching methods proposed in this project. On the one hand, the way used to store the minutiae in the task "Patron Generation". In [2], the minutiae are stored in a matrix, where a column is for the X coordinate and another column is for the Y coordinate. Moreover, there is no distinction among junctions and ends. In this project, the minutiae, called X_i, are stored in a vector, where in each position the minutia's X coordinate and the minutia's Y coordinate are stored as real and imaginary components of a complex value, respectively. Furthermore, there is no distinction among junctions and ends either. On the other hand, the task "False Minutiae Removing" has been expanded as due to the small screen of the sensor some minutiae appear in the limit in the case of big fingers. These minutiae have also to be removed as they are not intrinsic from the fingerprint and they can mislead in the matching process.

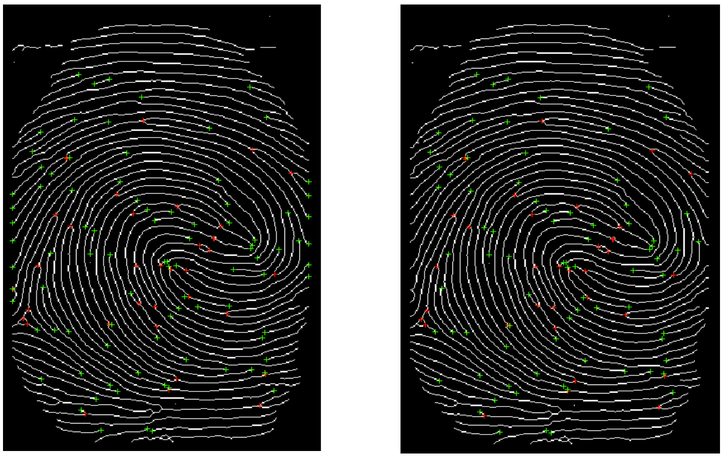

Figure 8 – Minutiae before and after the limit minutiae removing

3.2.2. Distortion in the minutiae patterns

Although using good minutiae extraction algorithms and the fact that every fingerprint is unique, the process of capturing the fingerprint image creates some differences between fingerprints of the same finger. Therefore, it is important to take these problems into account and delete or reduce their impact, or incorporate them in the matching process in order to identify each fingerprint correctly.

3.2.2.1. Translation and rotation

These problems are due to finger position in the moment of the capture. Doubtless, it is impossible to put the finger always in the same position and as a consequence, the fingerprint is translated and some times rotated. A finger displacement of just 2mm (imperceptible to the user) results in a translation of about 40 pixels in a fingerprint image scanned at a resolution of 500 dpi, which is the chosen sensor resolution. Nevertheless, intrinsic features of the fingerprint such as distance between minutiae remain equal. In this project they will be called T and $\Delta\theta$ (or delta), respectively.

3.2.2.2. Lack of minutiae

This problem would not exist if the sensors were ideal. It consists in the loss of minutiae due to the loss of ridges originated by some finger features like its dryness. This loss of ridges could also change an end into a junction and vice versa. For this reason, the minutiae are stored without distinction among ends and junctions. The impact of loss of minutiae is more complex than it seems, as depending on the method, not only the number of lost minutiae is important, also its position. For instance, those methods that require ordered minutiae. In this project the problem of the lack of minutiae produces an error in the mean and this error will be called m.

3.2.2.3. Pixel precision

It is also due to the finger position in the moment of the capture. However, in this case even intrinsic features of the fingerprint could be changed.

3.3. Matching

Once the minutiae are extracted, the next step is to identify the input fingerprint with the corresponding fingerprint of the database. To implement this, it is necessary to define a model for the minutiae set and a method, or matching criterion, to compare the similarity between two fingerprints. In this project, as has been told in the beginning of this section, two methods with its corresponding models have been implemented.

3.3.1. Full fingerprint matching

3.3.1.1. Signal model

The model of this matching consists in the application of two transforms in the minutiae set. These two transforms are the angle transform and the Fourier transform.

Angle transform

This transform solves the problem of translation, prepares the minutiae set for the Fourier transform and generates a model that take into account both the problem of rotation and the problem of lack of minutiae. Moreover, if the lack of minutiae is null (m=0), which will is called ideal case, this transform becomes really useful to solve the rotation in the matching criteria easily. For this reason, the impact of this transform will be explained both for the case without lack of minutiae (ideal case) and the case with lack of minutiae (non-ideal case).

Concerning its implementation, it is the same for both cases and it has two steps. The first one consists in computing the mean ($\bar{x}$) from all the minutiae. It guarantees, in the next step, a non-translation model both in the ideal and non-ideal case. However, as it will be explained later, in the non-ideal case the mean is not exactly the ideal case's mean.

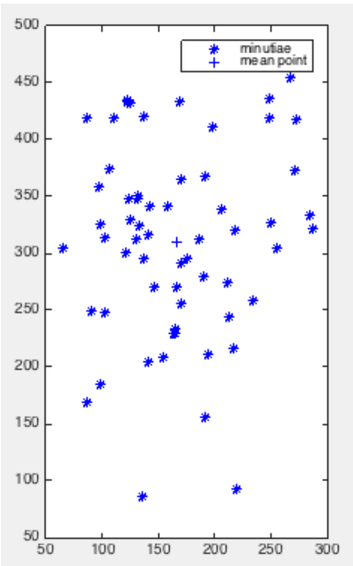

Figure 9 – Minutiae and mean point of a real fingerprint

The other step consists in change the reference of each minutia, which is its coordinates in an imaginary number, for two other references very different: (1) the vector that joint the mean point ($\bar{x}$) and the minutia, which is called x_i, removes the translation and is defined as a complex number, (2) the angle between the vector and a horizontal line crossing the mean point. This angle is called θ_i and is useful to prepare the minutiae for the Fourier transforms, as all the minutiae are ordered depending on its angle θ_i from 0 to 2π radians.

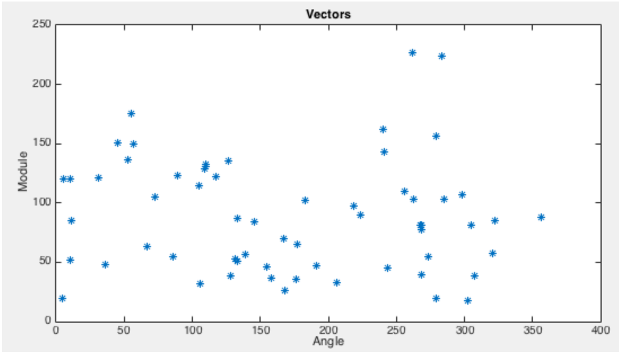

Figure 10 – Vectors obtained with the angle transform from the same fingerprint of the Figure 9

In the Figure 10, each point corresponds to a vector's angle in degrees in the coordinate axis and the corresponding vector's module, or distance between the minutia and the mean point, in the ordinate axis. The presence of higher modules around 90º and 270º is due to the fact that fingerprints are long, that is to say, the fingerprints' vertical distance is higher than the fingerprints' horizontal distance. This particular shape can be seen in the Figure 9.

Regarding the impact of this transform, as translation has just been deleted, it is concentrated in how this transform models both the problem of rotation and the problem of lack of minutiae. To see this, the impact is focused on the rotation and it is differentiated between the ideal case and the non-ideal case.

In the ideal case, a rotation in the coordinates field is translated into the same rotation in the vectors field and a circular shift in the angles field. Consequently, making the result 2π periodic, a rotation is translated into a simply shift in the angles field as can be seen in the Figure 11, where x's are the vectors. This procedure of making the signal 2π periodic is implemented in the Fourier transform but is evaluated here in order to simplify the model.

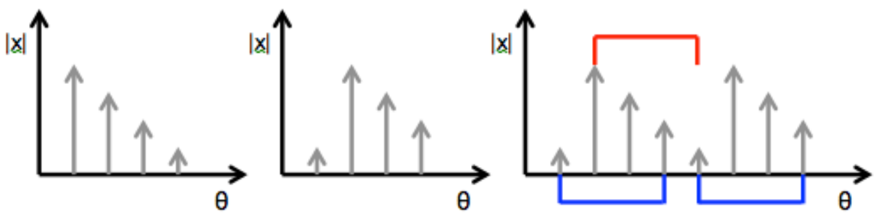

Figure 11 – Obtaining of a periodic signal: A) Original signal, B) Signal with a circular shift, C) periodic signal with a normal shift created with B)

Therefore, the corresponding ideal model is as can be seen in the (1), where i is the index of each minutia:

$$\hat{y}_i(\theta) = \hat{y}_i \cdot \delta(\theta - \hat{\gamma}_i) = x_i \cdot e^{j\Delta\theta} \cdot \delta(\theta - \theta_i - \Delta\theta)$$

where (2)

$$\hat{y}_i = x_i \cdot e^{j\Delta\theta}; \ \hat{\gamma}_i = \theta_i + \Delta\theta$$

In the non-ideal case, the resulting final model is more complex. The reason is that a lack of minutiae produces a modification of the mean point. Therefore, a constant (m) will be added in each vector and a particular phase will be added in each angle θ_i. It can be seen in the following figure, where the angle θ_i of upper minutiae increases with the lack of minutiae and the angle θ_i of lower minutiae decreases with the lack of minutiae.

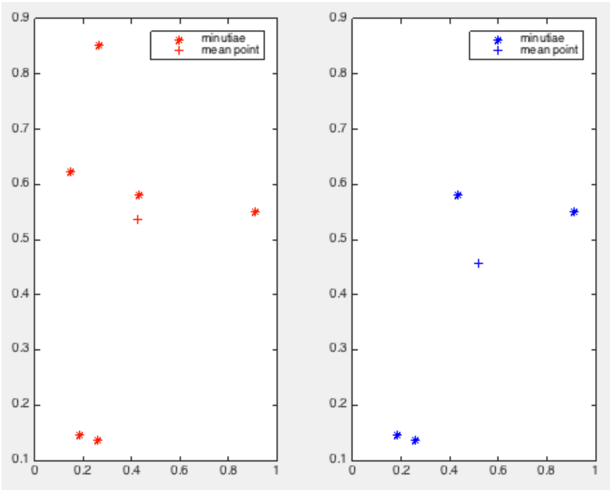

Figure 12 – Modification of the mean point in a synthetic fingerprint due to the lack of two minutiae.

In order to capture these effects in the model, the expression of the (2) has been used:

$$\gamma_i = -j\frac{1}{2}\cdot\ln\left(e^{2j\cdot\gamma_i}\right) = -j\frac{1}{2}\cdot\ln\left(\frac{|y_i|\cdot e^{j\cdot\gamma_i}}{|y_i|\cdot e^{-j\cdot\gamma_i}}\right) = -j\frac{1}{2}\cdot\ln\left(\frac{y_i}{{y_i}^*}\right) \quad (3)$$

Appling this expression, the resulting non-ideal model is:

$$\hat{y}_i(\theta) = \hat{y}_i\cdot\delta(\theta-\hat{\gamma}_i) = (x_i\cdot e^{j\Delta\theta}+m)\cdot\delta\left(\theta + j\frac{1}{2}\cdot\ln\left(\frac{x_i\cdot e^{j\Delta\theta}+m}{{x_i}^*\cdot e^{-j\Delta\theta}+m^*}\right)\right)$$

where (4)

$$\hat{y}_i = x_i\cdot e^{j\Delta\theta}+m;\ \hat{\gamma}_i = -j\frac{1}{2}\cdot\ln\left(\frac{x_i\cdot e^{j\Delta\theta}+m}{{x_i}^*\cdot e^{-j\Delta\theta}+m^*}\right)$$

It is important to realize that with m=0 the model is the same as the ideal model.

Fourier transform

This transform is used to reduce the impact of the problem of lack of minutiae. The principle is that this transform uses a summation of all the ordered minutiae and, as a consequence, the final result does not depend of the position of the missing minutiae. This fact is really important as a method that needs the position of the lost minutiae requires a time-consuming combinatorial evaluation to find theses positions. Furthermore, the number of lost minutiae almost does not affect the resulting transformed signal because of its little amount compared with the total number of minutiae.

Additionally, making the signal N-periodic, mentioned in the previous point as N=2π, is also applied in this transform. It entails computing the Fourier transform and then discretizing the result with f=k/N. The result is a vector y of length L where each value is expressed as $y(k)$. This length L is arbitrary. The higher is L, the higher are the quality of the results and the process time. Hence, it has been computed to achieve sufficiently good results with the minimum possible time.

$$y(f) = \int y(\theta) \cdot e^{-j2\pi f\theta} d\theta = \sum_i y_i e^{-j2\pi f\gamma_i}$$

where (5)

$$y_i(\theta) = y_i \cdot \delta(\theta - \gamma_i)$$

$$y(k) = \sum_i y_i e^{-j2\pi\frac{k}{2\pi}\gamma_i} = \sum_i y_i e^{-jk\gamma_i}$$

where

$$f = \frac{k}{2\pi}$$

In fact, the functions used in this Fourier transform are not basis functions as the different θ are not consecutive integers. As a consequence, the inverse transform cannot be computed, which is not a problem as the inverse transform is not needed.

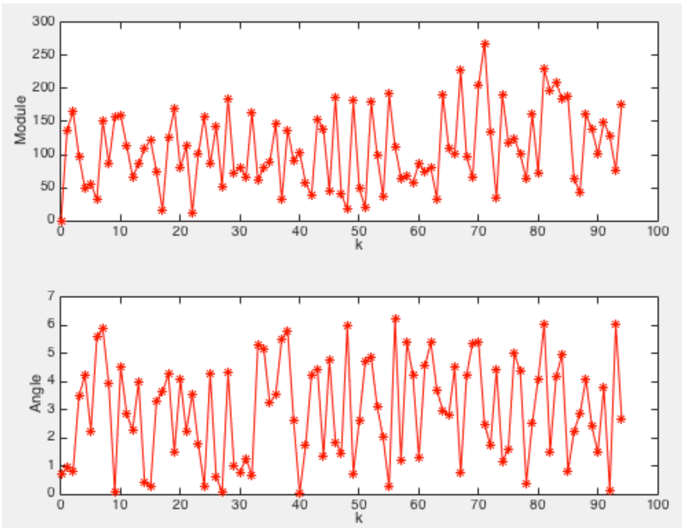

Figure 13 – Fourier transform of the fingerprint in Figure 9 with L=95

Therefore, once the Fourier transform is applied, the final ideal model taking into account the expression of (2) is:

$$\hat{y}(k) = \sum_i \hat{y}_i e^{-jk\hat{\gamma}_i} =$$

$$= \sum_i x_i \cdot e^{j\Delta\theta} \cdot e^{-jk(\theta_i+\Delta\theta)} = e^{-j(k-1)\Delta\theta} \sum_i x_i \cdot e^{-jk\theta_i} = e^{-j(k-1)\Delta\theta} \cdot x(k)$$

where (6)

$$\hat{y}_i = x_i \cdot e^{j\Delta\theta};\ \hat{\gamma}_i = \theta_i + \Delta\theta$$

And the final non-ideal model taking into account the expression of (4) is:

$$\hat{y}(k) = \sum_i \hat{y}_i e^{-jk\hat{\gamma}_i} =$$

$$= \sum_i (x_i \cdot e^{j\Delta\theta} + m) e^{-\frac{k}{2}\cdot \ln\left(\frac{x_i \cdot e^{j\Delta\theta}+m}{x_i^* \cdot e^{-j\Delta\theta}+m^*}\right)} = \sum_i (x_i \cdot e^{j\Delta\theta} + m) \cdot \left(\frac{x_i \cdot e^{j\Delta\theta}+m}{x_i^* \cdot e^{-j\Delta\theta}+m^*}\right)^{-\frac{k}{2}} =$$

$$= \sum_i (x_i \cdot e^{j\Delta\theta} + m)^{-\frac{k}{2}+1} \cdot (x_i^* \cdot e^{-j\Delta\theta} + m^*)^{\frac{k}{2}}$$

where (7)

$$\hat{y}_i = x_i \cdot e^{j\Delta\theta} + m; \ \hat{\gamma}_i = -j\frac{1}{2} \cdot \ln\left(\frac{x_i \cdot e^{j\Delta\theta}+m}{x_i^* \cdot e^{-j\Delta\theta}+m^*}\right)$$

3.3.1.2. MSE minimization as matching criterion

Once the model is defined, the next and final step is to define a matching criterion, to compare the similarity between two fingerprints. In practice, a slight displacement of the finger on the sensor provokes that some of the minutiae are lost. Therefore in this section the ideal model is not analysed and taking into account the non-ideal model established previously, the used matching criterion is the MSE minimization, which consists in finding the optimum values of Δθ and m that minimizes the MSE, where MSE is defined as:

$$MSE = \frac{1}{L}\sum_k |y(k) - \hat{y}(k)|^2 \quad (8)$$

In order to reach this minimum quickly and avoid testing all the Δθ and m possible combinations, two methods have been studied: Gradient method and Newton-Raphson method.

<u>Gradient method</u>

The minimum of a given function f is reached with the following iteration:

$$S_i = S_{i-1} - \mu \cdot \frac{\partial(f)}{\partial s}\bigg|_{S=S_{i-1}} \quad (9)$$

Where μ is a parameter that controls the convergence speed: the higher is μ, the higher is the convergence speed. However, it has an upper bound in order to avoid instability because of too big steps. In the full fingerprint matching case:

$$S = \begin{bmatrix}\Delta\theta \\ m\end{bmatrix}; \ \frac{\partial(f)}{\partial s} = \begin{bmatrix}\frac{\partial MSE}{\partial \Delta\theta} \\ \frac{\partial MSE}{\partial m}\end{bmatrix} \quad (10)$$

<u>Newton-Raphson method</u>

The minimum of a given function is reached with the following iteration:

$$S_i = S_{i-1} - H^{-1} \cdot \frac{\partial(f)}{\partial s}\bigg|_{S=S_{i-1}}; \quad H = \frac{\partial^2(f)}{\partial s^2} \quad (11)$$

In the full fingerprint matching case:

$$S = \begin{bmatrix} \Delta\theta \\ m \end{bmatrix}; \quad \frac{\partial(f)}{\partial s} = \begin{bmatrix} \frac{\partial MSE}{\partial \Delta\theta} \\ \frac{\partial MSE}{\partial m} \end{bmatrix}; \quad H = \begin{bmatrix} \frac{\partial}{\partial \Delta\theta}\frac{\partial MSE}{\partial \Delta\theta} & \frac{\partial}{\partial m}\frac{\partial MSE}{\partial \Delta\theta} \\ \frac{\partial}{\partial \Delta\theta}\frac{\partial MSE}{\partial m} & \frac{\partial}{\partial m}\frac{\partial MSE}{\partial m} \end{bmatrix} \tag{12}$$

Both methods, however have to face an important restriction: the MSE has local minimums and they are so close to the absolute minimum that is impossible to initialize neither Gradient method nor Newton-Rapshon method in a precise way. It can be seen in the following figure:

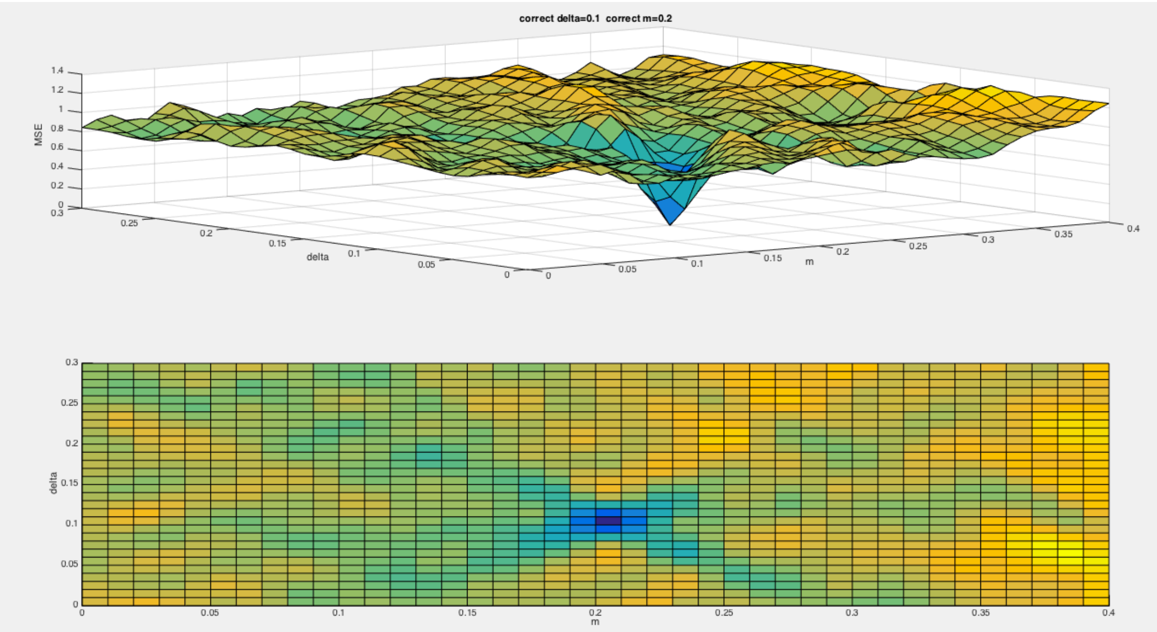

Figure 14 – MSE obtained from synthetics fingerprints with 50 minutiae, and optimal values of Δθ equal to 0.1 and m equal to 0.2. In this point the MSE presents an absolute minimum but many local minima are also present

Therefore, an exhaustive research calculating all the Δθ and m combinations taking into account that the m has real and imaginary part is needed. As a consequence, the processing time is high, which was tested in Matlab. These tests consisted in a recognition in a training database of 12 different fingerprints and although this method recognized the two input fingerprint correctly, the processing time for each input fingerprint was 15 minutes and 18 minutes, values that disqualifies the MSE minimization criterion.

3.3.2. Minutiae constellation matching

This matching is based in the previous full fingerprint matching but solving the error m in the calculation of the mean point due to the lack of minutiae that produced a very complex non-ideal model.

3.3.2.1. Signal model

The model also consists in the application of both the Angle transform and the Fourier transform but changing the vectors' definition in the Angle transform.

Angle transform

The main problem of this transform in the full fingerprint matching is the modification of the estimated mean point of a fingerprint in the presence of lack of minutiae. To solve the issue, in this matching the vectors' definition is changed from the vector that joins the mean point and the minutia into the vector that joins a minutia that plays as a centre of a circle of radius R and another minutia that is inside this circle. Therefore, the lack of minutiae does not affect the vectors. This is applied to all the minutiae, thus achieving a set of circular regions for each fingerprint, as can be seen in the Figure 15.

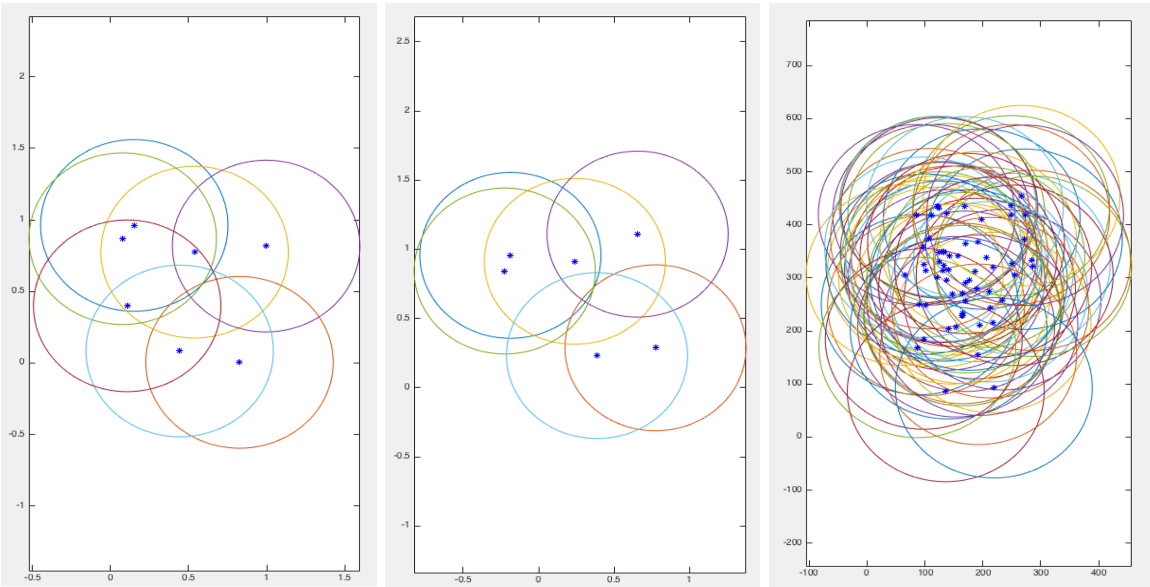

Figure 15 – A) Circular regions of a synthetic fingerprint, B) Same case but with lack of a minutia, C) Circular regions of the fingerprint of the Figure 9

The amount of vectors has been increased, because as can be seen in the Figure 15 each minutia is inside at least two circular regions: the circular region where it is the centre and the circular region where the centre is its neighbour minutia. However, the problem of the lack of minutiae has been solved, which is more important than the amount of vectors. Furthermore, the amount of vectors can be controlled with the length of the radius, which is arbitrary, it is the minimum to achieve sufficiently good results and it has been computed testing with different values. Moreover, the rotation affects each circular region in the same way that affects the fingerprint. Therefore, the corresponding model for each circular region is the ideal model seen in the full fingerprint matching (2), where "c" is the index of the circular region inside the fingerprint.

$$\hat{y}_{ci}(\theta) = \hat{y}_{ci} \cdot \delta(\theta - \hat{\gamma}_{ci}) = x_{ci} \cdot e^{j\Delta\theta} \cdot \delta(\theta - \theta_{ci} - \Delta\theta)$$

where (13)

$$\hat{y}_{ci} = x_{ci} \cdot e^{j\Delta\theta}; \ \hat{\gamma}_{ci} = \theta_{ci} + \Delta\theta$$

Fourier transform

The Fourier transform is exactly the same applied in the full fingerprint matching. Hence, the corresponding model for each circular region is the ideal model seen in the full fingerprint matching (6):

$$\hat{y}_c(k) = \sum_i \hat{y}_{ci} e^{-jk\hat{\gamma}_{ci}} = \sum_i x_{ci} \cdot e^{j\Delta\theta} \cdot e^{-jk(\theta_{ci}+\Delta\theta)} =$$

$$= e^{-j(k-1)\Delta\theta} \sum_i x_{ci} \cdot e^{-jk\theta_{ci}} = e^{-j(k-1)\Delta\theta} \cdot x_c(k)$$

where (14)

$$\hat{y}_{ci} = x_{ci} \cdot e^{j\Delta\theta};\ \hat{\gamma}_{ci} = \theta_{ci} + \Delta\theta$$

Actually, although the lack of minutiae does not affect the model, as the Fourier transform is a minutiae summation, there are differences between $\hat{y}_c(k)$ and $y_c(k)$ than in the ideal case of the full fingerprint matching. Nevertheless, these differences are still very low despite of the fact that the impact of a lost minutia is higher in a circular region, where there are fewer minutiae, than in the whole fingerprint.

3.3.2.2. Matching criteria

The matching criterion can borrow some from the MSE minimization criterion described in the previous sections.

In order to implement the matching criteria for the minutia constellation matching, the first step is to define the error $e_c(k)$ between the circular region $\hat{y}_c(k)$ and the circular region $y_c(k)$:

$$\hat{y}_c(k) = e^{-j(k-1)\Delta\theta} \cdot x_c(k) \ \rightarrow\ e_c(k) = e^{jk\Delta\theta} \cdot \hat{y}_c(k) - e^{j\Delta\theta} \cdot x_c(k) \quad (15)$$

Expressed in vectors:

$$\underline{e_c} = \underline{\underline{A}} \cdot \underline{\hat{y}_c} - e^{j\Delta\theta} \cdot \underline{x_c} \ ; \quad (16)$$

where $\underline{\underline{A}}$ is a diagonal (kxk) matrix that contain the $e^{jk\Delta\theta}$ values in the diagonal and zeros elsewhere.

Therefore, the following MSE_c expression for each pair of circular region is:

$$MSE_c = \frac{1}{L} \cdot \underline{e_c}^H \cdot \underline{e_c} = \frac{1}{L}\left(\underline{\hat{y}_c}^H \cdot \underline{\hat{y}_c} + \underline{x_c}^H \cdot \underline{x_c} - 2 \cdot \mathbb{R}\left\{ \underline{x_c}^H \cdot \underline{\underline{A'}} \cdot \underline{\hat{y}_c} \right\} \right) \quad (17)$$

where $\underline{\underline{A'}}$ is a diagonal (kxk) matrix with the $e^{j(k-1)\Delta\theta}$ values in the diagonal and zeros elsewhere.

Once obtained the MSE_c expression, three criteria are proposed to evaluate the similarity between two fingerprints. Both consist in comparing all the circular regions $\underline{x_c}$ from a fingerprint $\underline{x}$ with all the circular regions $\underline{\hat{y}_c}$ from another fingerprint $\underline{y}$, for a set of possible values of Δθ.

The first criterion compares the Δθ that minimize MSE_c in (17). It builds a matrix, called errors matrix, which contains the optimum value of Δθ for all the possible combination between each $\underline{x_c}$ and each $\underline{\hat{y}_c}$. Then, the lowest MSE_c is extracted from the matrix and its corresponding row and column are removed. This procedure is repeated until the matrix is null, and then all the extracted MSE_c are added up. In this way, for each $\underline{x_c}$, the most similar $\underline{\hat{y}_c}$ is selected. The resulting value of the summation is the MSE that defines the similarity between both fingerprints. Hence, the recognized fingerprint is the one reaching the lowest MSE among all in the database.

The second criterion follows the same strategy but it compares only the term $2 \cdot \mathbb{R}\left\{\underline{x_c}^H \cdot \underline{\underline{A'}} \cdot \underline{\hat{y}_c}\right\}$.

A third criterion is proposed that take advantage of another feature of the used model: the rotation described in the model in section 3.3.1.1 affects each circular region $\underline{x_c}$ in the same way. In other words, the variance of the optimum rotation values Δθ obtained for the different regions is ideally zero for the best matching fingerprint. Therefore, a matrix, containing the optimum value of Δθ for all the possible combination between each $\underline{x_c}$ and each $\underline{\hat{y}_c}$ is built and in each position. Then, the Δθ that are in the same positions of the MSE_c extracted from the error matrix (the lowest) are also extracted from the deltas matrix and they are stored in a vector. From this vector, the variance of the deltas of the $\underline{x_c}$,$\underline{\hat{y}_c}$ is computed. The obtained Δθ variance divides the MSE or the $2 \cdot \mathbb{R}\left\{\underline{x_c}^H \cdot \underline{\underline{A'}} \cdot \underline{\hat{y}_c}\right\}$ by the Δθ variance plus a lower value to avoid the infinite in the case of Δθ variance equal to zero as this variance, in some cases, does not differentiate some fingerprints sufficiently to be used by itself. Consequently, it is possible to obtain good results with much less time, as the MSE_c calculation does not need to be so precise (lower radius, which implies less vectors, lower length L of the Fourier transform and bigger steps in the research of each Δθ, which implies less steps).

Finally, due to its better results, the final chosen method is the maximization of $2 \cdot \mathbb{R}\left\{\underline{x_c}^H \cdot \underline{\underline{A'}} \cdot \underline{\hat{y}_c}\right\}$.

3.3.2.3. Identification of artefacts

Nevertheless, once the minutiae constellation matching was tested in real fingerprints it did not reach the expected results. Evaluating the failed cases, a common fingerprint feature was observed: an injury or a trace on the skin that produced a high concentration of minutia around it. To solve it, the number of really closed minutiae is computed for each minutia and if this number is higher than a certain established bound, some really closed minutiae are removed, thus reducing the local density. It is also applied in each circular region with the same purpose. An example of an injured fingerprint can be seen in the Figure 22 C).

4. Evaluation

In order to consolidate the minutiae constellation matching as a suitable method that accomplish all the requirements and specifications of the project an exhaustive evaluation is needed. To carry out it, a training database is needed.

4.1. Training Data Base

Although some tests have been done with synthetic fingerprints, for instance the test of Figure 12, the final chosen training database is made by fingerprints captured with the sensor for the sake of obtaining realistic results, as this is the sensor that will be used in the refugee camp. Moreover, in order to obtain a big database involving few people, five different fingers have been captured from each person and they are stored as different people. These fingers are the left index and the thumb, the index, the middle finger and the ring finger of the right hand. In total, the training database is formed by 60 fingerprints, which correspond to the 5 mentioned fingers of 12 people. Moreover, in order to achieve lower processing times the fingerprints are stored as a circular regions set in the frequency domain and with different Δθ. The place where this training database is stored is a laptop computer, as it needs to be loaded in the Matlab, the platform used to execute the matching tests. All the captures of the fingerprints of the training database have been done in a laptop computer as well. Moreover, in order to evaluate the minutiae constellation matching, 61 more fingerprints have been captured to play as input fingerprints in the evaluation. These fingerprints are both from stored fingers and non-stored fingers to also evaluate the rejection process of the matching.

4.2. Scoring

All the parameters used to evaluate the minutiae constellation matching performance are probabilities estimations ($\hat{p}$). Specifically, the maximum likelihood estimation expression is obtained in (18), where p_0 is the real probability, *n* is the number of tried recognitions and *k* depends of the estimated probability. For instance, if the estimated probability is the probability of recognize correctly, *k* will be the number of correct recognitions.

$$\hat{p} = argmax\ f(k) \rightarrow \frac{\partial f(k)}{\partial p_0} = 0 \rightarrow \hat{p} = \frac{k}{n}$$

where *f(k)* corresponds to a binomial density function (18)

$$f(k) = \binom{n}{k} \cdot {p_0}^k \cdot (1 - p_0)^{n-k}$$

However, the reliability of the estimated probabilities depends of the variance of the estimate. That is to say, given a certain $\hat{p}$, the actual value of p_0 will be in an interval around the $\hat{p}$, with a certain probability ∝ that in this project is fixed in 95%. Concerning the intervals of each possible $\hat{p}$, they are computed for each p_0 and then they are related with the $\hat{p}$ joining them for all the possible p_0's in a graphic. Specifically, the expression for each bound $(E_{\propto/2}', E_{\propto/2})$ of the interval is:

$$p\left(\frac{k}{n} < E_{\propto/2}'\right) = \sum_{k=0}^{E_{\propto/2}' \cdot n} \binom{n}{k} \cdot {p_0}^k \cdot (1 - p_0)^{n-k} = \frac{\alpha}{2} \quad (19)$$

$$p\left(\frac{k}{n} > E_{\propto/2}\right) = 1 - \sum_{k=0}^{E_{\propto/2} \cdot n} \binom{n}{k} \cdot {p_0}^k \cdot (1 - p_0)^{n-k} = \frac{\alpha}{2}$$

Figure 16 shows the confidence intervals for n=40:

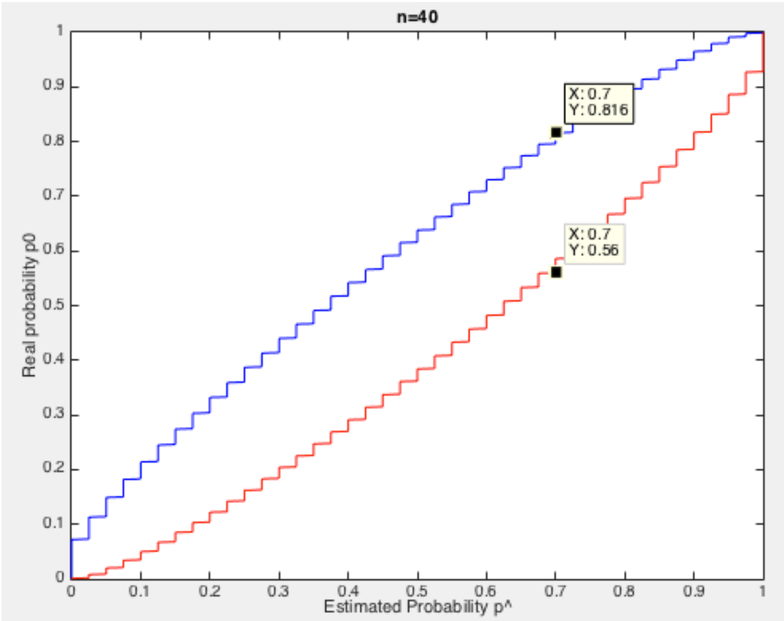

Figure 16 – Confidence intervals on p_0 given an estimated value of the probability

The interpretation of Figure 16 for a trail with 28 correct recognitions would be the following: the estimated probability of correct recognition is 70% (28/40) and hence, the actual value of the probability is inside the interval [56% , 81.6%] with 95% reliability. Following this methodology, the probabilities of the minutiae constellation matching shown in the Table 2. They are 1) Correct recognition probability: the user is correctly identified as a stored user and he is correctly recognized. 2) False recognition probability: the user is correctly identified as a stored user but he is wrongly recognized. 3) False rejection probability: the user is wrongly identified as non-stored user. 4) Correct rejection probability: the user is correctly identified as non-stored user. 5) False acceptance probability: the user is wrongly identified as stored user.

Probability	Value (%)	Interval of the corresponding p_0 (%)	Reliability (1- α)	n
Correct recognition (CRC)	70	[56 – 81.6]	0.95	40
False recognition (FRC)	17.5	[8.6 – 30.3]	0.95	40
False rejection (FRJ)	12.5	[5.1 – 24.5]	0.95	40
Correct rejection (CRJ)	71.4286	[51.3 – 86.7]	0.95	21
False acceptance (FA)	28.5715	[13.3 – 48.7]	0.95	21

Table 2 – Minutiae constellation matching's probabilities

4.2.1. User identification: confusion matrix

In order to analyse deeper the false recognition cases, in addition to the false recognition probability, a matrix is built containing the probabilities of wrong identifications. This is helpful to see the behaviour of the matching method and be able to identify the problems that guide to a false recognition. As example of its structure, a confusion matrix computed with the training database of 60 users (60x60) is printed as image in the Figure 17.

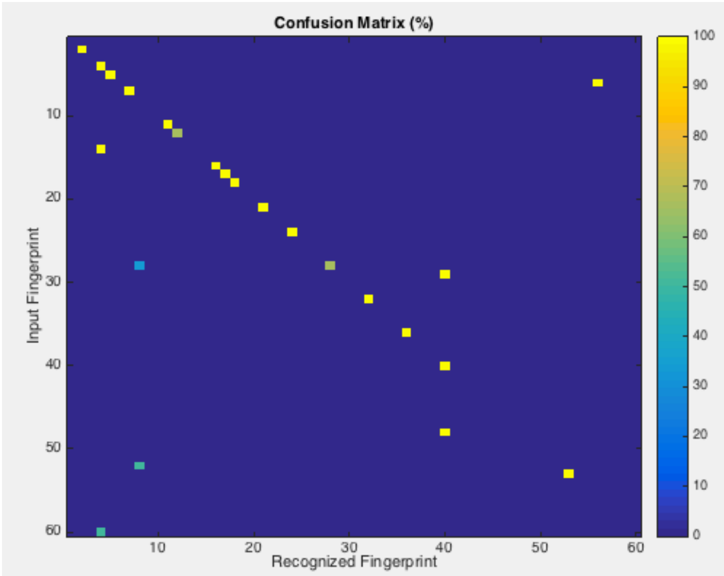

Figure 17 – Confusion matrix example

Due to the high correct recognition probability, the most non-zero values are in the diagonal of the confusion matrix, which has been computed with the selected parameter β explained in the following section.

4.2.2. User verification: sensitivity and specificity

In order to decide if a user is not stored in the database and as a consequence it has to be rejected, a parameter (β) is implemented. Its implementation consists in reject the recognized user if the value of its maximum metric $2 \cdot \mathbb{R}\left\{\underline{x_c}^H \cdot \underline{\underline{A'}} \cdot \underline{\hat{y}_c}\right\}$ with the input fingerprint, is lower than β. The value of β affects both the false acceptance probability and the false rejection probability: the higher is β, the higher is the false rejection probability and the lower is the false acceptance probability. Hence, its value is chosen analysing these two probabilities for different β. It can be seen in the Figure 18, where the values of beta are $6.1 \cdot 10^6$, $8.1 \cdot 10^6$, $1.01 \cdot 10^7$, $1.21 \cdot 10^7$ and $1.41 \cdot 10^7$ and the chosen value is $1.01 \cdot 10^7$ as achieve lower values both in the false acceptance probability and false rejection probability.

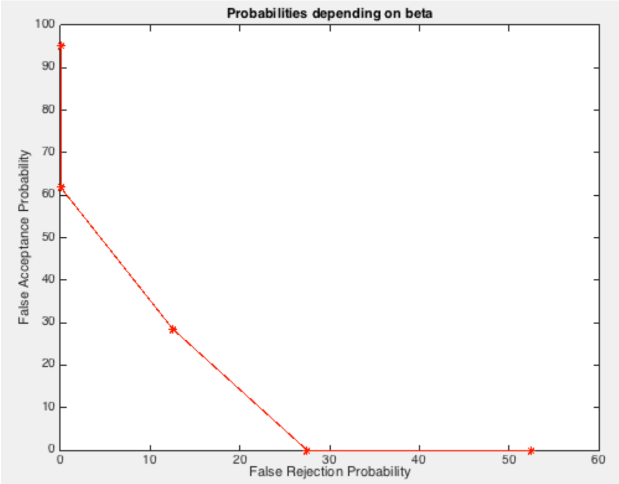

Figure 18 – False acceptance vs False rejection (%)

4.3. Processing time

Apart from good statistics results, another important feature that the minutiae constellation matching has to accomplish is a lower processing time. The refugee camp has 10 thousand people and if the processing time is not taken into account, the project implementation could be unfeasible.

The achieved processing time, since the input fingerprint is captured until the fingerprint is identified within the training database, is around 5 seconds using a MacBook Pro 15" 2.3GHz Intel Core i7 from 2012. Translated into a 10 thousand fingerprints database, the processing time is around 14 minutes.

As a final step, the minutiae constellation matching is compared to the matching method developed in [2] in terms of the resulting probabilities. Table 3 shows the probabilities obtained with the method [2], which should be compared to the probabilities in

Table 2.

Probability	Value (%)	Interval of the corresponding p_0	Reliability (1- ∝)	n
Correct recognition (CRC)	57.5	[43.4 – 70.8]	0.95	40
False recognition (FRC)	42.5	[29.2 – 56.6]	0.95	40
False rejection (FRJ)	0	[0 – 7.2]	0.95	40
Correct rejection (CRJ)	0	[0 – 13.2]	0.95	21
False acceptance (FA)	100	[86.8 – 100]	0.95	21

Table 3 – Probabilities of Nuria's matching

The proposed method is clearly superior: probabilities associated to the correct (wrong) decisions are higher (lower). Moreover, the superiority is also observed in terms of the processing time. Specifically, in the Nuria's matching the total achieved processing time since the input fingerprint is loaded until the fingerprint is identified in the training database is around 1 minute, while only 5 seconds are needed for the minutiae constellation matching.

5. Implementation

5.1. Fingerprint sensor

The chosen sensor is the futronic fs88:

Figure 19 – Futronic fs88

This sensor supports Android platform, a feature not common in fingerprint sensors, and a feature that involves two requirements for the android devices: USB on-the-go (OTG) and an external SD card.

On the one hand, USB OTG is a feature that allows USB devices such us mobile phones to act as a host, allowing other USB devices such as USB flash drives, keyboards, or fingerprint sensors in this case, to be attached to them. This feature is quite new in the mobile phone market and only some new mobiles or tablets have it. On the other hand, the reason for the external SD card is simply to storage all the fingerprint images captured by the sensor, that is to say, the place where the database is stored in the android application. In the final version of the system, which is not implemented in this project, the database will be stored in an external hard disk connected in a computer acting as a server of all the android devices used in the refugee camp.

Therefore, two android devices that fulfil all these features have been selected for this project. One of them is the Sony Xperia z tablet. This tablet was available in the department and has been used in this project to implement the android application. The other tablet is the Primux zonda 2 tablet 7”. This tablet has not been used in this project but is the tablet selected for the refugee camp, as is one of the cheapest tablets that fulfil all the requirements mentioned above.

In reference of the sensor’s API, which is needed to capture the fingerprint image, it is free. In order to acquire it, it is necessary to contact Futronic, as it is not available in the Internet, and then, they send you the address and password of their ftp server. However, other products such as SDK for fingerprint recognition are not free.

5.2. Android application

Before the start of the project, the knowledge of Android was null. Therefore, a previous learning phase and a search of a development platform were needed. Once they were achieved, a simple user interface with the Android studio platform could be developed. However, as it has been told in the incidences section of this thesis, the development of the final Android application planned for this project has not been possible. Its flow diagram can be seen in Figure 20. As its purpose is testing the matching in the android platform, both the part of communication with the server and the part of the medical histories, which have been analysed in [1] is not shown. Therefore, the database of this testing application would be stored in the external SD card mentioned in the previous point. Furthermore, as the database in this testing application would not be big, to remove a patient, the user should select a patient from a list with all the stored patient's names (Figure 21 B). However, in the final system it could not be possible as the list would contain 10 thousand people and to find a specific name would be very difficult:

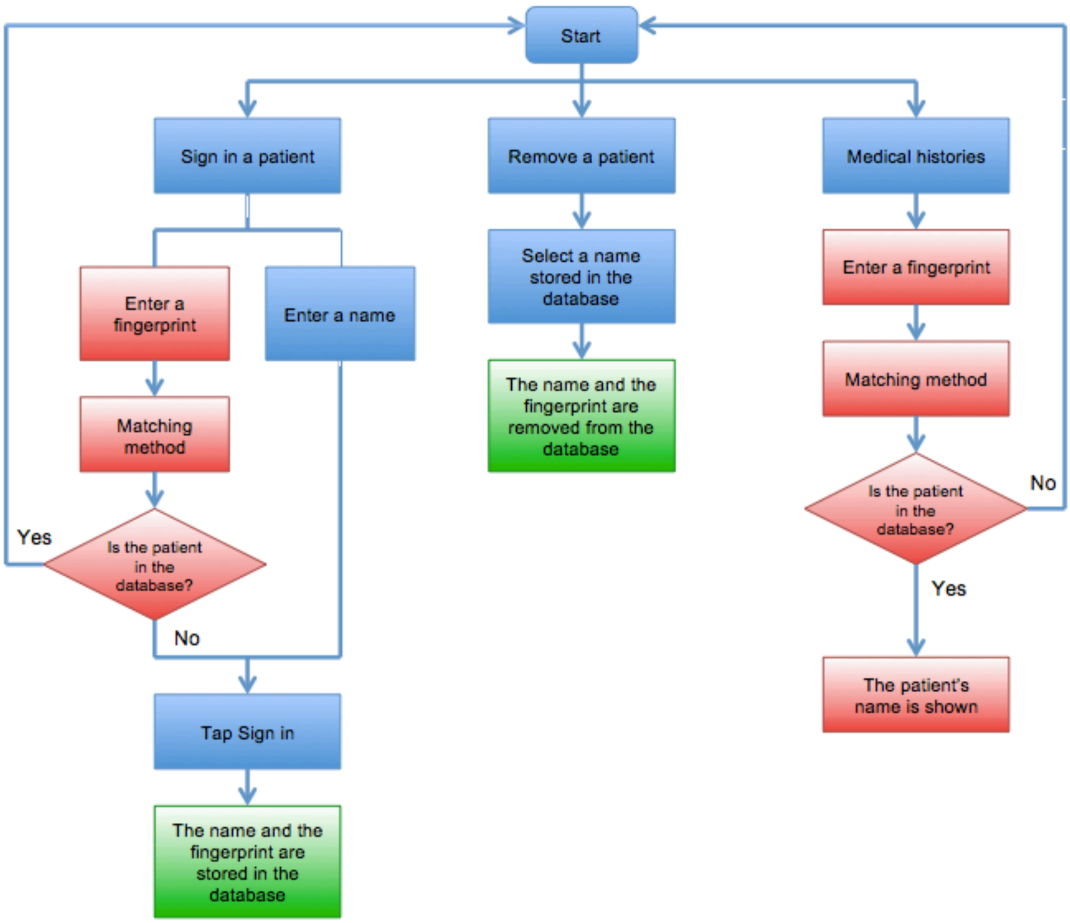

Figure 20 – Android application's functional diagram

When the patient was not identified, the application would go to the start to allow the user to choose between sign in a patient or try to enter its fingerprint again in the case of a wrong rejection.

In reference of the colours, the red tasks are those not included in the simply user interface and the green tasks are those that are implemented in the simply user interface with the modification that the fingerprint is not stored/removed as it is not captured. Therefore, the simply user interface only operates with the patient's names and, as a consequence, an other task is added after the "Enter a name" task that consist in check that the name is not already stored in the database. Finally, some screen example of the simply user interface are shown in the following figure:

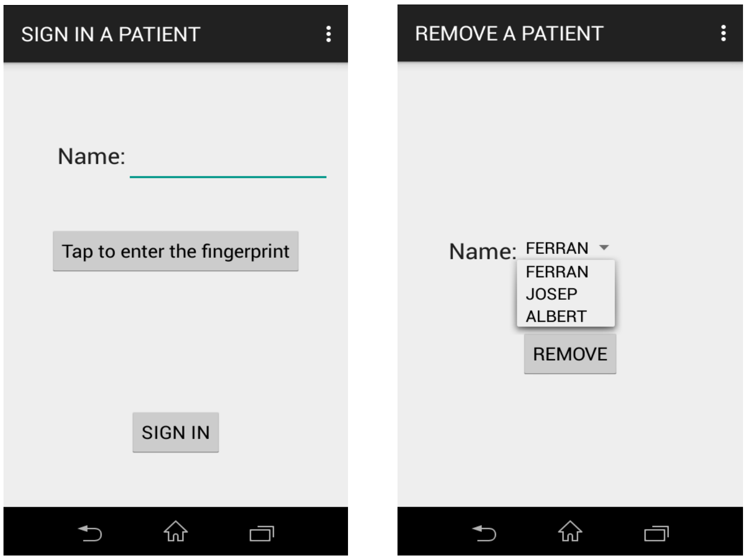

Figure 21 – A) Sign in screen, B) Remove screen

As can be seen in the Figure 21, A), the "Tap to enter a fingerprint" buttons are already added in the simply user interface. However, their function is not implemented (nothing occurs when they are tapped).

6. Budget

In order to compute the total cost of the project, three different categories have been done. They are the development cost, the installation cost and the maintenance cost.

6.1. Development cost

This cost involves the dedicated hours to develop the project and the licenses and the amortization of the software used. In the first case, the project has been developed in 20 weekly hours during 20 weeks. Therefore, the corresponding cost evaluated as a cost of a junior engineer (27€/hour) is:

$$20\ \frac{hours}{week} \cdot 22\ weeks \cdot 27\ \frac{€}{hour} = 11880\ € \qquad (20)$$

In the second case, the software used to develop the project has been Matlab and Android studio. The standard Matlab license costs 2000€[1]. However, the UPC students can use the license of the university freely. The other software, Android studio, has a free license. Therefore, the development cost is the cost calculated in (19): 11880€

6.2. Installation cost

In order to compute the installation cost of the project, the first necessary thing is listing the cost of each component. Although this project is focused in only one part of the system expounded in [1], which is the matching method, in the cost is necessary to evaluate the whole system as this is what the NGO requires. Therefore, the components of the final system that will be implemented in the refugee camp are 20 sensors, 20 android devices, a server computer and an external hard disk. The amount of sensors and android devices is the amount calculated in [1] as in that project a deep analysis of the refugee camp was done. For the same reason, the server, the external hard disk and the android devices have the same features of those selected in that project. However, the sensor is the one mentioned in section 5.1 of this thesis, as the sensor used to develop this project. Therefore, the cost of the project is:

- Android devices: Primux zonda 2 tablet 7", 20x69€[2] = 1380€
- Sensors: Futronic fs88, 20x64€[3] = 1280€
- Server computer: HP 250 G3 Intel N2840/4GB/500GB/15.6" = 283.30€[4]
- External hard disk: Toshiba Canvio Basics 2,5" 1 TB USB 3.0 = 54.99€[5]
- TOTAL installation cost= 2998.29€ + shipping

[1] Visited page: http://es.mathworks.com/pricing-licensing/ date 30/06/2015
[2] Visited page: http://tablets-baratas.es/products/primux-zonda-2-tablet-7-wifi-4-gb-android/ date 30/06/2015
[3] Visited page: http://shop.innovatrics.com/products/futronic-fs88 date 30/06/2015
[4] Visited page: http://www.pccomponentes.com/hp_250_g3_intel_n2840_4gb_500gb_15_6_.html date 30/6/2015
[5] Visited page: http://www.fnac.es/Toshiba-Canvio-Basics-2-5-1-TB-Disco-duro-externo-negro-Almacenamiento-Disco-duro-externo/a1063370 date 30/6/2015

6.3. Maintenance cost

This cost has also to be computed in order to know the real impact of the project for the NGO. However, this cost is null as the refugee camp has solar panels that produce enough energy to cover the energy consumption of the system. The energy consumption, which comprises the server computer and the charge of the tablets, has been computed:

Device	Power consumption (W)	Number of devices	*Working hours per day	Energy consumption per day (W/day)	Total (W/day)
Server computer	45	1	24	1080	1260
Tablet's battery	3	20	3	180	

Table 4 – Total energy consumption calculation

*The values of this column are also extracted from [1].

Moreover, as it is explained in this other project, a technician is needed to revise or correct the functioning of the system. It comprises the checking of the tablets, the modification of the parameters of the application in order to include any necessity that could appear, the checking of the server computer, the removing of the unnecessary data of the database and finally the updating of the exportations from the server computer to the external hard disk. As these tasks are not difficult, they could be done by a technician of the refugee camp, and, as a consequence, it does not implies an extra cost for the NGO.

Therefore, the final maintenance cost is zero, thus obtaining a total cost of the system of 14878.29 €.

7. Conclusions and future development

The objective of this project is to develop a system required by the Doctors Without Borders NGO that consists in managing the medical histories of a refugee camp in an easier way. Specifically, the system was already designed in final degree project "SOLUCIONES TIC PARA LA CREACIÓN DE SISTEMAS DE SEGUIMIENTO DE HISTORIALES CLÍNICOS EN CAMPOS DE REFUGIADOS" [1] by Irene Baeza and deals with the manage of medical histories in a database by an application, and using fingerprint matching for the patient's identification. Due to the poor results obtained in the fingerprint identification, this project is focused in enhancing the performance. Two minutiae-based matching methods have been developed using the minutiae extraction from the final degree project "IDENTIFICACIÓ D'EMPREMTES DIGITALS" [2] by Núria Esquís but only one of them has reached sufficiently good results. It is based in the vectors between the minutiae in the frequency domain and it has been compared with the matching method developed in [2] because of the good results also obtained in that project.

The results have been satisfactory. The munitae constellation matching has reached a correct recognition probability of 70% and a correct rejection probability of 71.4286% with a processing time of 5 seconds in a database compounded by 60 people. Probabilities much higher than in [2] and with 55 seconds less. In order to compute these values with enough reliability, a deep statistic analysis has been done, which comprises both a computing of the intervals of reliability and a huge work in the fingerprint capturing building a training database.

Moreover, the matching method is implemented in Matlab but in the final system it has to be implemented in an Android platform. To that end, an Android application plugged to a fingerprint reader had to be developed to test the behaviour of the matching method. As its purpose was basically for testing the matching method, it did not include neither the part of communication with the server nor the part of the medical histories expounded in [1]. However, as it has been explained, this part has not been finished, and only a basic user interface has been done.

As future work, one objective is to complete the Android application incorporating both the fingerprint capturing, the identification of the minutiae, and the matching method to the basic user interface. Moreover, in order to finish the whole project required by Doctors Without Borders, once checked the correct operation of the matching method in the Android platform with the chosen sensor, the last pass would be to incorporate the matching method in the application. Furthermore, it will be necessary to contact with MSE in order to know if the matching features are enough or they have also to be enhanced as a future work. In this case, some possible alternatives could be the incorporation of the distinction between ends and junction or the incorporation of the ridge's orientation.

Bibliography

[1] Irene Baeza Rojo. "Soluciones TIC para la creación de sistemas de seguimiento de historiales clínicos en campos de refugiados". TFG, Escola Tècnica Superior d'Enginyeria de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Spain, 2014.

[2] Núria Esquís Flores. "Identificació d'empremtes digitals". TFG, Teoria del senyal I comunicacions department, Escola Tècnica Superior d'Enginyeria de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Spain, 2007.

[3] Anil Jain, Lin Hong, Ruud Bolle. "On-Line Fingerprint Verification". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 302-314, April 1997.

[4] Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar. *Handbook of Fingerprint Recognition*, 1st ed. New York, USA: Springer-Verlag, 2003.

Appendices

Appendix A:

Examples of fingerprint images captured with the sensor:

Figure 22 – A) Perfect fingerprint, B) Dry fingerprint and C) Injured fingerprint

Glossary

AUCOOP	Associació d'Universitaris per a la Cooperació
ETSETB	Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
NGO	Non-governmental organization
MSF	Doctors Without Borders
TIC	Tecnologías de la Información i la Comunicación
FRR	False Recognition Range
FAR	False Acceptance Range
MSE	Mean Square Error
USB OTG	Universal Serial Bus On-The Go
SD card	Secure Digital Card
API	Application Programming Inteface
SDK	Software Development Kit