



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

FINAL THESIS

Design and Implementation of a Finger Vein Identification System

Studies: Telecommunications Engineering

Autor: Duque Vehils, Jose Miguel

Director: Hong Zheng

Any: 2011

General Index

Collaborations	6
Gratitude	7
Resumen del Proyecto	9
Resum del Projecte	10
Abstract	11
1.Introduction	13
2.Data Acquisition	15
2.1.The device for image acquisition.....	15
2.2.The Procedure.....	17
2.2.1.Image Acquisition	17
2.2.2.Finger Vein Segmentation.....	20
3.Features Extraction and Matching Algorithms	25
3.1.Hausdorff Distance (HD) method	26
3.2.FD Distance method	27
3.3.Maximum Minimum Distance (MMD) method.....	29
3.4.Score-Level Fusion Based on SVM	31
4.Experimental Results.....	33
4.1.Kernel Density Estimation with Gaussian Kernel to Generate the Probability Distribution Function	33
4.2.KDE Application for Method Modeling	35
4.3.Genuine Acceptance Rate and False Acceptance Rate calculation	38
4.4.Combinatory for N samples problem	41
5.The proposed Solution	46

5.1.The Server	46
5.2.The Client.....	51
5.2.1.SQL Database Management	51
5.2.2.Image Processing	54
5.2.3.Matching Functions	55
5.3.User Interface.....	57
5.3.1.The Main Window	58
5.3.2.New User.....	59
5.3.3.Settings	60
5.3.4.Users' Window	61
5.3.5.View User Window and View Sample Window	62
6. Conclusions	65
7. References	66

Collaborations



Intelligent Computing and Intelligent Systems Laboratory, Wuhan University.

Gratitude

I would like to have an special thanks for this work to all my collage during my stay in the ICIS Laboratory in Wuhan, China, for their patience and willing to help me while I was lost in the execution of this project.

I would like also to thanks the foreign relations department in the Polytechnyc University of Catalonia for giving me the chance to go China for the realization of this work.

Finally, I would also add an special thanks to professor Hong Zheng, lead researcher and teacher in the ICIS Laboratory, Wuhan University. I would like to thank him for helping me during the realization of this project and his patience to try to have a good understanding with each other, despite the cultural differences.

Resumen del Proyecto

El objetivo de este proyecto ha sido el desarrollo y diseño de un sistema de identificación basado en el patrón definido por las venas de los dedos que pudiese ser usado por un número limitado de usuarios en un entorno de red. El trabajo realizado se puede dividir en dos partes, el estudio y desarrollo del proceso de reconocimiento y el desarrollo de la solución en red.

En general, se reconoce que es una tarea de gran dificultad el diseño de un sistema de identificación basado en las venas de los dedos que consiga un elevado nivel de rendimiento. Para este propósito se han estudiado diferentes métodos y su combinación para definir una configuración final, la cual esta explicada en este proyecto.

Este proyecto también presenta un nuevo método para el reconocimiento de los patrones definidos por las venas de los dedos basado en propiedades geométricas, adquiriendo la distancia máxima y mínima entre dos líneas cualesquiera en un barrido transversal del patrón.

Estudiará también el uso de diferentes métodos y su combinación mediante las llamadas Maquinas de Soporte Vectorial (SVM) para proporcionar un valor representativo de la comparación entre dos patrones.

Presentará el modelado utilizado para la caracterización y determinación de las características del sistema de cara a realizar una configuración final del sistema basado en el uso de "*Kernel Density Estimation*" (KDE).

Finalmente se presentara la solución en red desarrollada, usando un modelo cliente-servidor mediante el uso de herramientas gratuitas para base de datos SQL (MySQL Server).

Resum del Projecte

L'objectiu d'aquest projecte ha sigut el desenvolupament i diseny d'un sistema d'identificació basat en el patró definit per las venes dels dits que pogues ser utilitzat per un número limitat d'usuaris en un entorn en xarxa. El treball realitzat es pot dividir en dos parts, l'estudi i desenvolument del process de reconeixement i el desenvolupament de la solució en xarxa.

En general, es reconegut que es una tarea de gran dificultad el diseny d'un sistema d'identificació basat en les venes dels dits que aconseguixi un elevat nivell de rendiment. Per aquesta finalitat s'han estudiat els diferents mètodes i combinacions per a definir una configuració final, la qual s'explica en aquest projecte.

Aquest projecte també presenta un nou metode de reconeixement de patrons definits per les venes dels dits basat en propietats geometriques, adquirin la distancia màxima i mínima trovada entre dos líneas qualsevols en un escaneig transversal del patró.

Estudiarem l'ús de diferents metodes i la combinació mitjançant l'us de les anomedades Maquines de Suport Vectorial (SVM) per a proporcionar un valor representatiu de la comparació entre dos patrons.

Presentarem també el modelatge utilitzat per a la caracterització i determinació de les caracteristiques del sistema de cara a realizar la cofngiraucó final del sistema basat en l'ús de "*Kernel Density Estimation*" (KDE).

Finalment, presentarem la solució en xarxa desenvolupada, fent servir un model de client-servidor mitjançant l'ús de eines gratuites per bases de dades SQL (MySQL Server).

Abstract

The target of this project was the design and development of a finger vein identification system that could be used by a limited number of users in a networked environment. The work realized was divided in two parts, the study and development of the finger vein recognition process and the networked solution.

In general, it is recognized that it is a great challenge to design a finger vein identification systems that achieves certain level of performance. For this propose different extraction methods and combinations were studied in order to define a final setup that is contained and explained in this thesis.

This thesis also proposes a new finger vein pattern recognition method based on geometrical parameters of the finger vein pattern, which is to acquire the values of the maximum and minimum distance between two lines of cross section scans of the finger vein pattern.

It will study also the use of different methods and the combination of results to generate a final matching score using a Support Vector Machine (SVM).

It will present the modeling used to characterize and determine the performance of different methods in order to decide a final system setup, based in Kernel Density Estimation (KDE).

Finally, this thesis presents the developed networked solution. It will present a server-client structure using free SQL Database servers tools (MySQL Server).

1. Introduction

Nowadays there is an increased interest in modern societies with the development and deployment of internet and web technologies for methods that can verify or identify the identity of a user that access from a remote location. Traditional security systems as key locks or identification cards are also target for a modernization that can upgrade the security of critical locations such as ATMs, banks, nuclear power plants, etc. Those and other different scenarios are pushing the development of more sophisticated systems based on biometrical information given the impossibility of a malicious individual to reproduce the information.

Those systems are usually known as biometrical identification systems. Systems through pattern recognition can identify an individual by a unique biometrical feature. Theoretically, the ideal biometrical feature for human identification should include: easy to be extracted from an individual, hard to be access by general public and hard to be reproduced by anyone else.

The acquisition of biometric parameters is a very hard procedure since it requires the conditions around the acquired parameter be as similar as possible. To achieve this target, it's necessary to make a combination between hardware design and software procedures. Through hardware design, the system may instruct the user to perform properly to make the pattern recognized easily. Through software procedures, the system may correct the problems related to the acquisition of the pattern, relying on algorithms that aid to solve irregularities. Furthermore, the algorithm is able to create the score of similarity in biometrical character from extracted fingerprints, the result of which is assumed to be accurate with infinite decimal.

Fingerprint identification is one of the most common biometric systems to identify individuals. There are three main advantages: firstly, it's easy to extract the fingerprint; secondly, the size of the device can be small; thirdly, the algorithms used for identification have been widely researched, analyzed and tested. However, there is one unsolved problem in this area - falsification, since it's easy to obtain the fingerprint of an individual from any object that he/she has touched in the past.

Thus, finger vein identification has become one of the main trends of biometrical identification systems in recent years. However, one of the main disadvantages of finger vein identification is that the result run by

the system is easily changed since the finger vein pattern extracted for the identification will be modified if the individual rotates his/her finger or revises the level curvature inside the device.

This thesis will study and propose an implementation of a finger vein identification system. For such task, this thesis will be defined in different chapters that contain the information and explanations related to the data acquisition, the features and extraction for identification, the study of results, the design of the proposed solution and finally a conclusion that will explain the challenges found around finger vein identification and tips for hardware design.

2. Data Acquisition

It's understood as data acquisition for finger identification the process that targets the process that starts from the camera input to the final extracted information that the system requires. This data has to be representative of the individual finger and needs to be able to deliver similar results between captures in order to be able to define a proper identification system.

The procedure for which the information is captured, transformed, extracted information about the user and finally compared with a database samples is the finger vein identification process.

2.1. The device for image acquisition

The main body property used to acquire the images required for finger vein identification is the fact that blood vessels are opaque to Near-Infrared light and at the same time, bones and flesh aren't, delivering a different degree of shadowing at a picture taken in this wave length.

A finger vein identification system will target such property through the use of a Near-Infrared (NIR) illumination system that targets to create the required lighting conditions to create an input frame for a Near-Infrared sensor from the light that pass through the user fingers and defines a pattern in the input image.

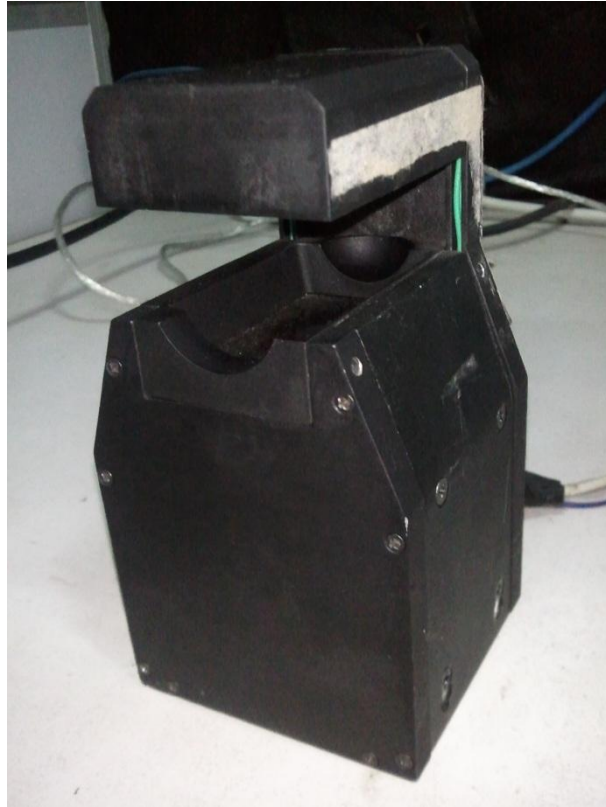


Fig 1: Device used in the acquisition.

The device used for this project, shown at Figure 1, consists into an array of Near-Infrared (NIR) light illuminators regulated by a COM interface that serves the propose as NIR light source, a NIR camera that serves the propose of capturing the light that passes through the user's finger and create an image that can be processed by a computer and some physical support for the users to place the finger.

The illuminators are located at the top-front part of the finger and the camera is located at the base of device heading up. The illuminators have a design fault and haven't been located homogeneously at the top of the device, given a stronger lighting level at the front of the device.

The camera acquisition resolution is setup at 320x240 pixels. However, given the distance between the camera and the finger, the region of interest for the system has a resolution of 210x131 pixels, this count as a 35.8% of the source picture resolution.

The device only has two supports for the front and back of the finger respectively without support for the hand. This have proven to provoke that different pictures are likely to have problems with rotation and different levels of curvature of the fingers given the high degree of freedom the users have to place the finger.

2.2. The Procedure

Figure 2 shows the procedure related to finger vein identification. This process can be divided in two blocks, the first one: *image acquisition* and the second one: *vein segmentation and feature extraction*. The target of the first block it's to acquire a clear picture from the finger and the second block target it's to generate a pattern as clear as possible and obtain representative features from it that can be used in the identification process. The identification or authentication process will require an stage of feature extraction and matching. The features studied and used in this thesis and their respective matching methods will be explained in the following chapter.

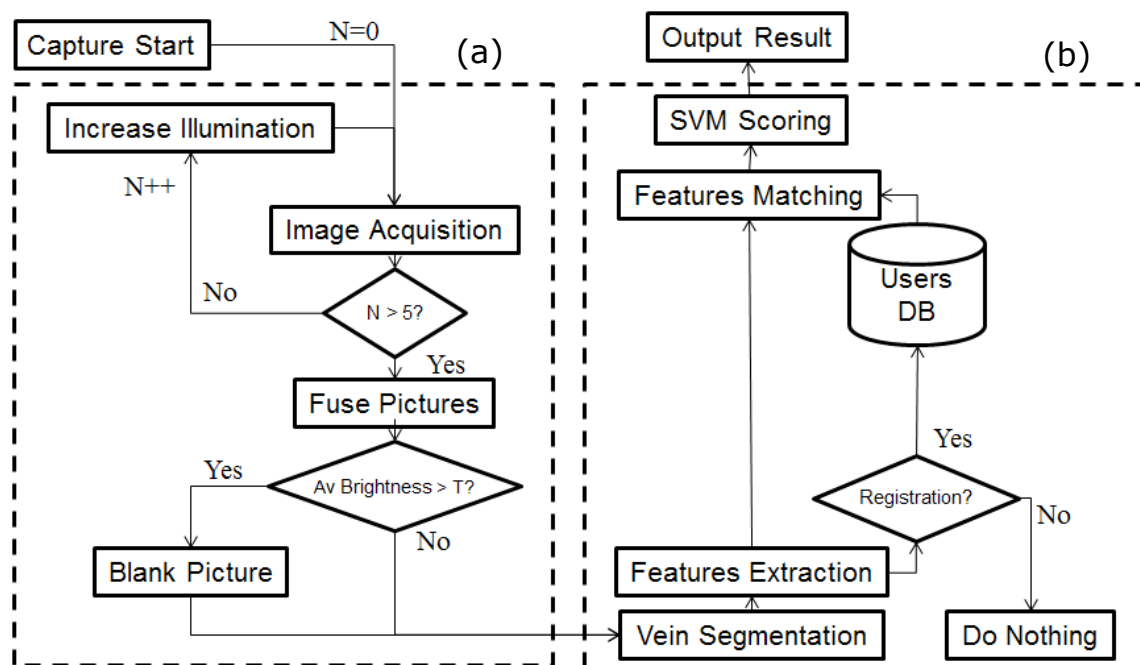


Fig 2: a) Image Acquisition Block b) Vein Segmentation, Feature Extraction and Matching Block

2.2.1. Image Acquisition

Acquiring a clear picture from the user can be more challenging than may be expected. We have to remember that we are using an infrared camera that captures the image that flows from a led array from the top of the device through the user's finger to the camera. The amount of light that will be delivered to the sensor will vary according to the user's finger thickness.

This problem requires modifying the camera exposure times to compensate the lack of light captured and at the same time to be adaptive

in order to avoid users calibration. There is also problems related to different degrees of lighting at different parts of the finger at the device used and the fact that some people has fingers with different degrees of thickness at the same finger. All this problems will be manifested as lack of clarity of the finger pattern, noise, over exposition at some areas, etc.

In order to obtain a good image for data acquisition at the first block it's required to capture several different pictures at different illumination levels and fuse the different pictures using a pixel per pixel averaging using the Equation (2.1).

$$A(x, y) = \left(\frac{I_1(x, y)^{\frac{3}{4}} + I_2(x, y)^{\frac{3}{4}} + \dots + I_{N-1}(x, y)^{\frac{3}{4}}}{N} \right)^{\frac{4}{3}}_{\forall x, y} \quad (2.1)$$

Where x and y , are the coordinates of the pixel and N the number of images used for fusing. During this thesis, the results and the calculations were realized using 5 different pictures, improving the visibility of the veins and the stability of the extracted vein pattern given the noise reduction that is obtained after the averaging of multiple samples from the same source.

It's good to remember the device has some design faults that provoke that the region of the picture taken by the camera near the illuminators position has saturation problems and at the same time, given a picture that avoids, this picture has the side effect that at the back part of the picture appears a darkness problem. Different pictures at different illumination levels were required.

To achieve the desired result it's required to choose a non-balanced average that tries to reduce the score of highly illuminated areas, since the most common problem is to achieve some level of vein visibility into the area surrounding the illuminators. Using a non-balanced average is possible to keep the stronger shadow provoked by the finger veins detected in low exposition times.

In order to solve the problem related to the fact that different users will have fingers with different thickness, the global illumination level of the picture is checked and used to adjust the exposition time of the camera in order to try to get pictures that have similar levels of luminosity and improve the vein visibility, once more, trying to avoid the saturation of the camera near the illuminators area. The different illumination levels can be observed in the Figure 3 as well as the fusion result.

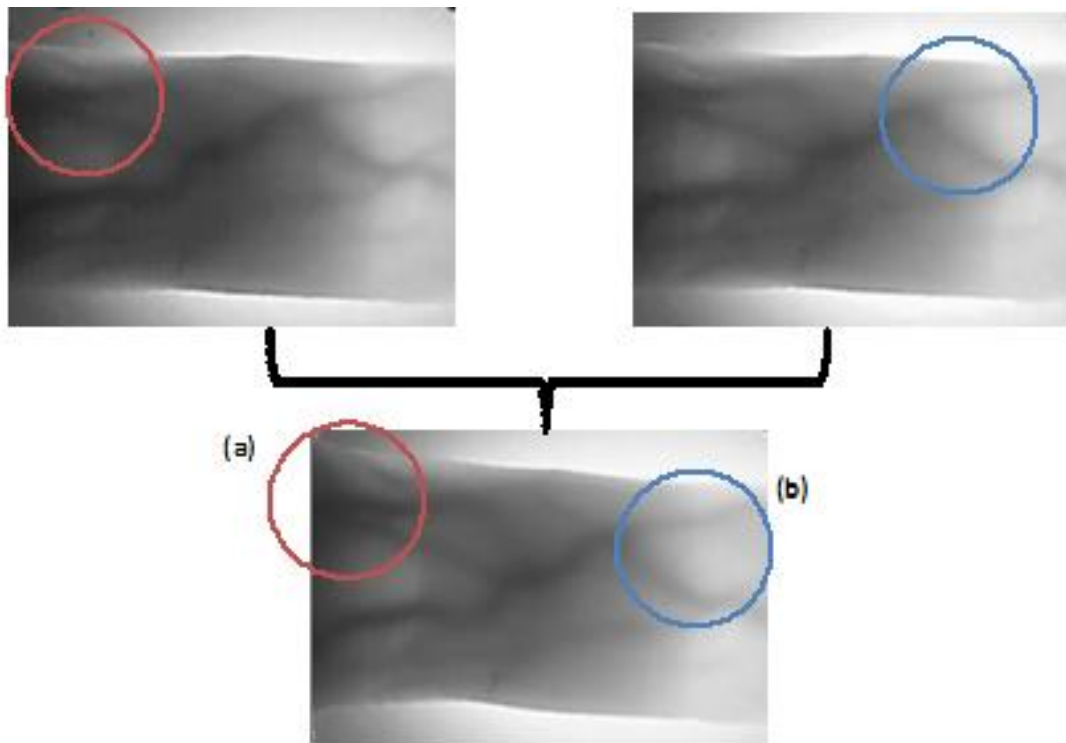


Fig 3: a) Smoother vein pattern near the darkest part of the picture. b) Smoother transition from dark part of the finger to a more illuminated region.

With this procedure, it's is possible to overcome some of the device design faults and at the same time as shown in the Fig 3, the final picture has a more recognizable finger vein pattern and less noise than neither the lowest or highest exposition time.

2.2.2. Finger Vein Segmentation

Usually the acquisition of the vein pattern is limited to an area inside the finger. However, the experimentation during this project showed that including the finger shape as one more finger vein, fusing it with the vein closer to the border of the finger, improved the end user experience, increasing the probability that a genuine user was identified.

The utilization of a fuzzy segmentation method avoids identifying the finger vein contour to isolate it from the finger vein pattern during the extraction. This project proposes for such a task an algorithm similar to the one found at the study of *Miura (2007)*, with a lower level of complexity and similar performance as shown in equation (2.2) that relies in the fact that veins are represented as a decrease of the brightness level in the picture and tries to detect those valleys into the brightness cross-section profile to define the finger vein pattern detected. Figure 4 shows an example of the application of equation (2.2) to the brightness profile. As pointed by *Miura (2007)*, the position of a finger vein near the border of the finger is hidden by the increase of luminosity derivate from the decrease of the finger height. In our case, this phenomenon is used to avoid the separation of the last finger vein detected and the finger side detected. This can be observed comparing Fig 6a and Fig 6b.

It would be possible to get the same result of the original formula proposed by *Miura (2007)* applying two times the equation (2.2). As shown in Figure 5, both valleys can be detected. However, the experimentation proved that the vein pattern from some people that has thin fingers, as girls or some women is hard to detect. Adding the finger side information increases the pattern complexity and also the system properties.

$$C(y) = P(y) \frac{\sum_{q=-W/2}^{W/2} P(q) T_r(H-1 > y+q > 0)}{\sum_{q=-W/2}^{W/2} T_r(H-1 > y+q > 0)} \quad (2.2)$$

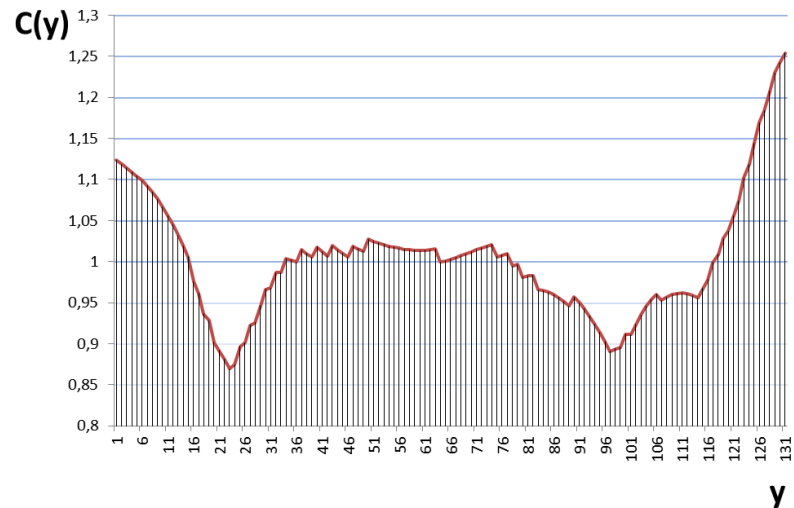


Fig 4: $C(y)$, Cross-sectional profile of finger-vein image

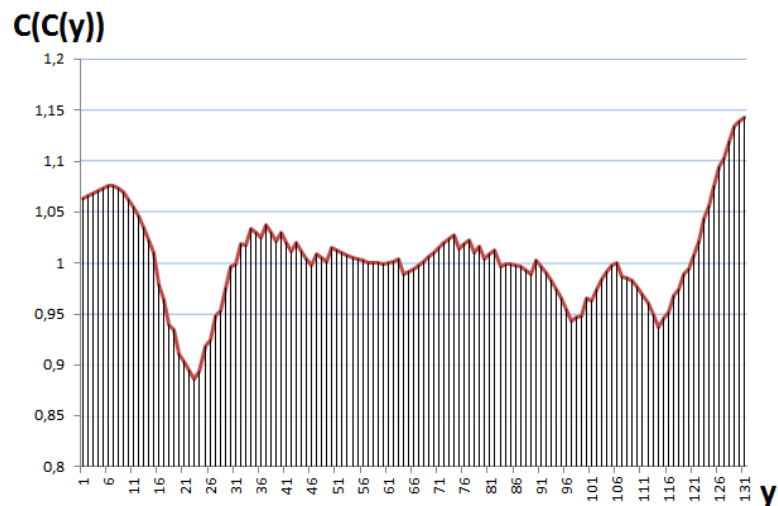


Fig 5: $C(C(y))$, Cross-sectional profile of finger-vein image.

In equation (2.2), H is the height of the picture, W it's an odd value to use as window size (35 was used for the above graphs and study) for a local average filtering, $T_r(b)$, is a logical function that given the expression b is true the value it's 1, 0 otherwise and $P(y)$ it's the function of the profile value of the image.

In order to detect the finger veins and the sides a simple threshold to Equation 2.2 can be used, since the background of our pictures it's bright and the start of the finger it's characterized by a drop in the average brightness level as can be observed in the Figure 4. However, the finger veins closer to the finger side won't be detected and will be fused

with the finger side itself, that's especially useful for fingers with almost undetectable finger veins.

There are some issues related to the acquisition of the final image by this method, such as single isolated points around the main detected pattern. In order to avoid such irregularities a smooth Gaussian filter is applied to the finger vein pattern to reduce the segmentation noise.

Once obtained the finger vein segmentation (Figure 7b), it's required to apply a thinning algorithm to reduce the width of the pattern to a single pixel width line. This project uses an 8-kernel matrix algorithm for such a task. Fig 6 shows the 8 kernels used.

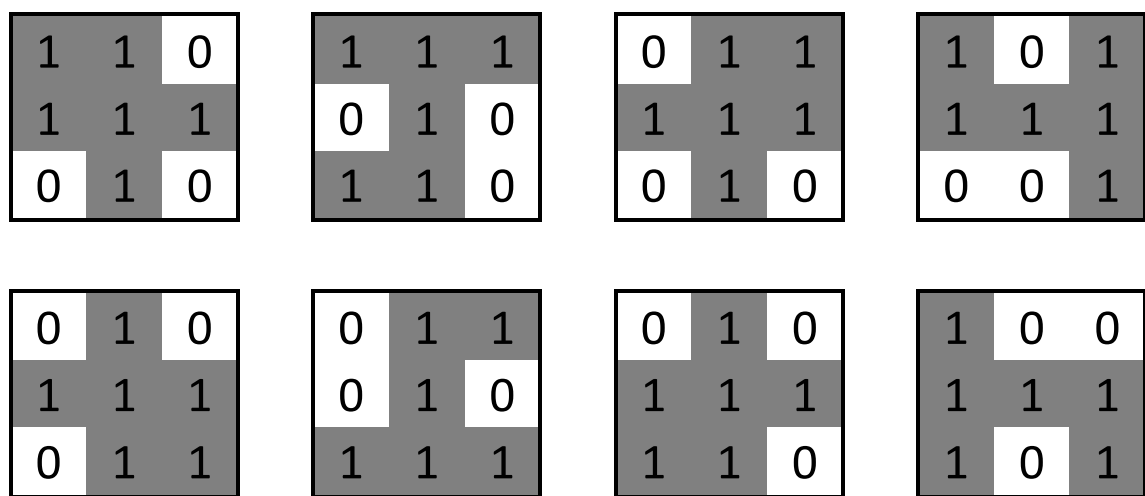


Fig 6: The 8 Kernels used in the thinning algorithm.

The thinning using those 8 kernels is an iterative algorithm that consists into the following steps:

- Create a matrix with the white pixels as 1, blacks as 0
- Create a matrix with the black pixels as 1, whites as 0
- Apply each kernel at both matrix
- Each kernel should have created a matrix as result for both, white and black matrixes.
- Apply a binary threshold at 2.99 both matrixes
- Delete the pixel from the source if both matrices are true.
- Restart the process until no pixel is deleted

After thinning the input image, the target will be the irregularities like small branches and holes found at the finger vein pattern that usually add noise to the pattern since they are randomly detected. In order to delete them is required to follow the next procedure:

- Apply a Hough Transform to detect the pattern lines
- Filter the lines that are shorter than a threshold distance.
- Join the lines that are separated less than a given distance
- Apply an Gaussian smooth the final result
- Apply a binary threshold to the image
- Apply the thinning algorithm again

After those steps, the final vein pattern that will be used for feature extraction is obtained, as shown in the Fig 6d.

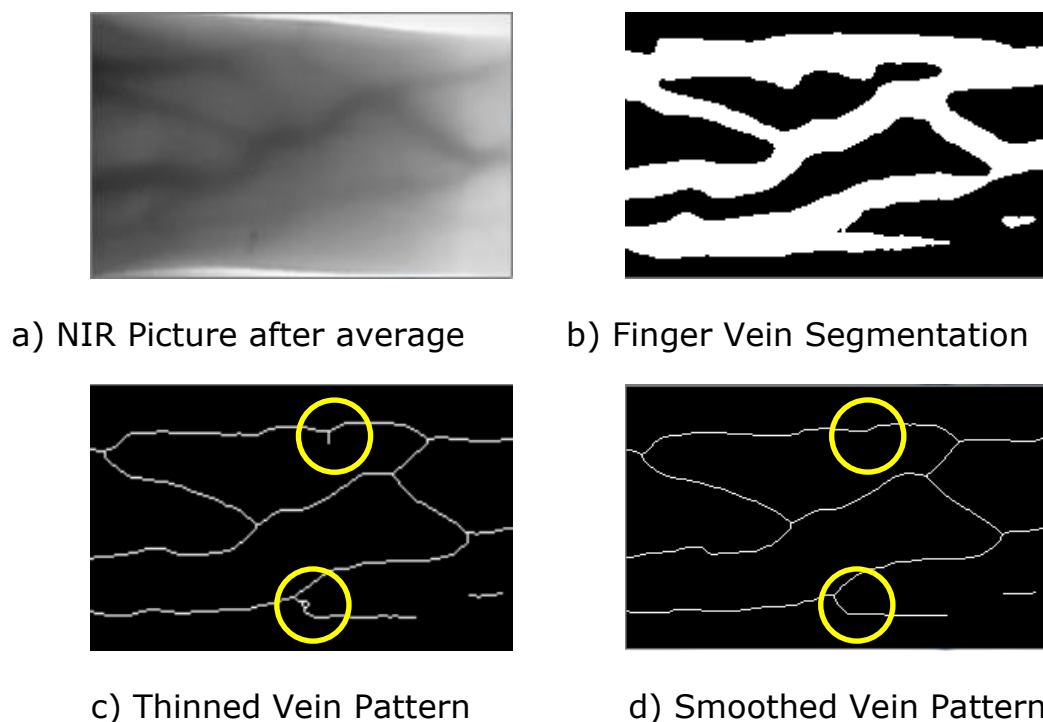


Fig 7: Example of the finger vein segmentation and thinning process. Circles mark the location of irregularities before and after the smoothing of the vein pattern.

As you can notice in the smoothed vein pattern, the introduction of the finger side as one more vein fused with the top and bottom finger vein, increases the number of lines detected and the complexity of the final vein pattern. This aid in the identification of a given vein pattern and at the same time decreases the correlation between finger vein patterns from different fingers. It also helps to avoid the problem related with fingers with almost un-detectable vein patterns. In this case, the finger geometry becomes the finger vein pattern used for identification proposes. It also has the advantage that the rotation of the finger commonly hides the vein closer to the finger side. Fusing the side and the closest finger vein helps

to decrease the differentiability between both finger vein patterns. The idea to use the finger shape in combination with finger vein was pointed by at the study realized by *Kang (2009)*.

During the smooth of the vein pattern the final picture size is normalized to a resolution of 280x180 pixels. This allows the possibility to modify the image size during the image acquisition without requiring the user to register again and isolate the image acquisition from the user matching process. However, maintenance operations will be required and user may be required to be registered again in case that the image acquisition system is changed.

3. Features Extraction and Matching Algorithms

Once the finger vein pattern is acquired the next step in an identification system is to choose, design the implementation of the pattern recognition algorithm and features that will be used for identification. Multiple kinds of data and pattern recognition algorithms can be fused in a single final score. Those systems are usually known as multimodal systems since they use different features to perform a single identification.

In this project three different kinds of data from the finger vein pattern were selected as feature points and the fusion between them in order to obtain a final result, such features are:

- The end and cross points of that can be found in the final picture after the image acquisition procedure.
- The finger vein pattern as it is will be recorded as a feature.
- The maximum and minimum distance found between two finger veins in a cross-section scan of the finger vein pattern as geometric feature.

An example of the data set acquired from a finger vein is shown at Figure 8.

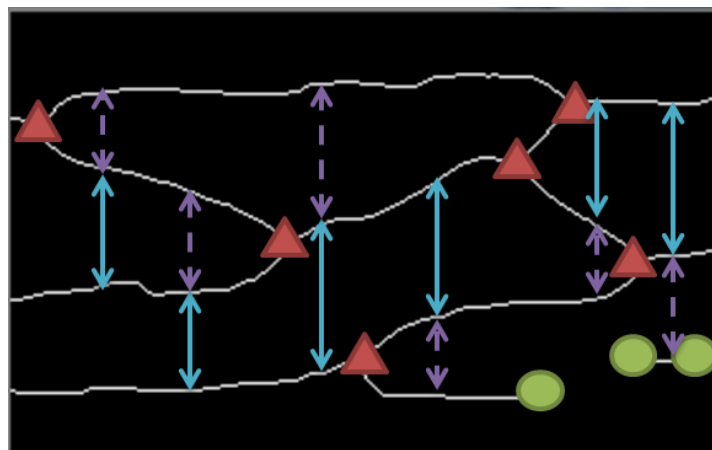


Fig 8: Example of feature points extracted: a) Triangle: Cross point b) Circle: End Point c) Continuous Arrow: maximum distance value d) Discontinued Arrow: minimum distance value

3.1. Hausdorff Distance (HD) method

A traditional pattern recognition method used for finger vein identification is the acquisition of the cross and end points of the finger vein pattern as feature points to define a constellation that characterizes a finger vein pattern. Then, the distance between two different constellations of cross and end points can be measured using the Hausdorff distance (HD). The MHD algorithm is shown at equation (3.1). Given that different data sets can have different number of cross and end points and can't be assumed that all the fingers scanned will contain end and cross points at its pattern, the modified Housdorff distance (MHD) shown at equation (3.1) can't be used as simplification of the HD. One of the main problems about using this procedure is the noise of the segmentation, thinning and smoothing of the vein pattern process plus the rotation translations of the finger between captures can create and hide features points and consequently, the distance measure may widely vary.

$$MHD(X, Y) = \frac{1}{n_x} \sum_{x_j \in X} \min_{y_i \in Y} \|x_j - y_i\| \quad (3.1)$$

Where X and Y are the collections of points; x_j and y_i are single points of such collections.

$$HD(X, Y) = \max(MHD(X, Y), MHD(Y, X)) \quad (3.2)$$

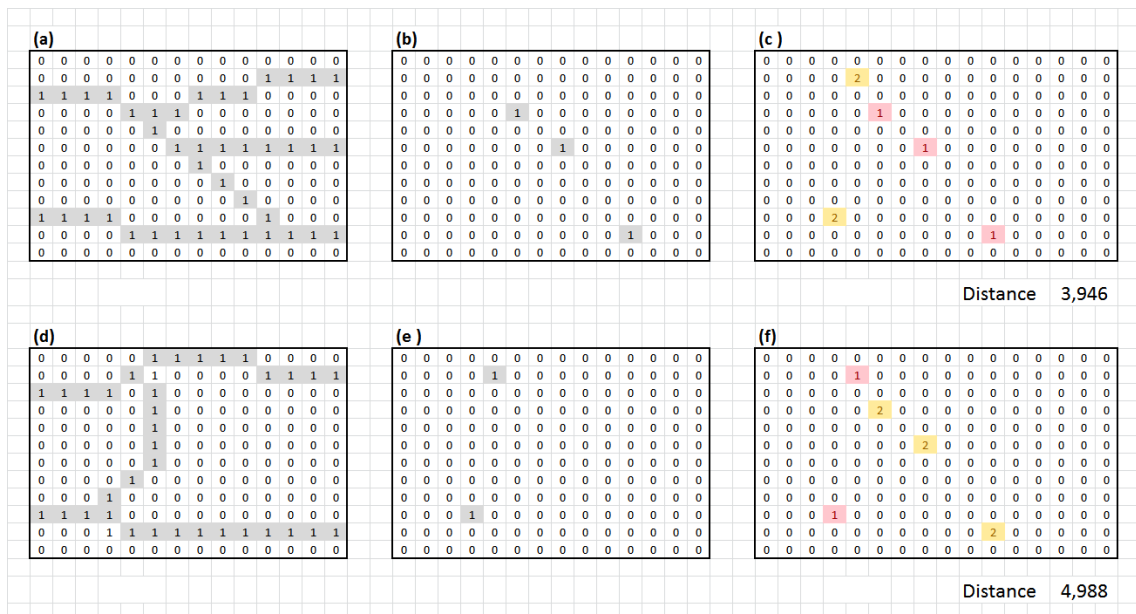


Fig 9: Simplified example of MHD applied to cross points.

a,d) Simplified thin pattern of the samples

b,e) Generated matrix of cross points

c,d) Matrix containing the end points of both samples used to calculate the distance. The distance value it's calculated from the red set (2's) against the yellow set (1's) using the MHD.

Figure 9 shows the distance calculated using the MHD algorithm between two simplified vein patterns (Figure 9a, 9d). It is possible to see that the distance between both patterns varies depending which is the reference set of points. This serves as an example also about why it is required to use the HD algorithm.

3.2. FD Distance method

A commonly used method to compare different shapes, in this case the created used from the finger vein pattern is the use of templates. This is another traditional approach to the finger vein pattern recognition problem. This method usually relies directly to the segmented pattern found and tries to apply a two dimensional hamming distance for the score calculation.

However, in this project a variance of such method is used to decrease the storage requirements of the system, the implementation has been designed to work with the final thinned pattern. The implementation tries to recover the original segmentation from the thinned pattern, for this reason, in this work this measure has been redefined as "*Fat Distance*", since it works with non-thinned version of the pattern.

In order to recover an approximation of the original segmentation result the next steps can be applied to the thinned pattern:

- Gaussian Smooth of the thinned pattern
- Binary Threshold

Selecting the wide (9 in this thesis) of the Gaussian smooth it is possible to define the target wide of the input finger vein pattern. This will define the minimum distance in pixels from the input sample to the reference pattern sample that marks an input pixel as found.

The matching calculation is performed in a similar way to a 2-D binary Hamming distance with some degree of toleration with the position in the matrix. Figure 10 shows a graphical example of the calculation using a simplified version of two samples and displays the numerical distance result. It also shows that the distance between two samples is not reciprocal.

$$FD(X, \bar{Y}) = \frac{\sum_{i,j} (x_{ij} - x_{ij} y_{ij})}{(\sum_{i,j} x_{ij}) + 1} \quad (3.3)$$

$$EFD(X, Y) = \max(FD(X, \bar{Y}), FD(Y, \bar{X})) \quad (3.4)$$

Equations (3.3) and (3.4) define the mathematical calculation for such a distance. In (3.3) X is the collection of points in the thinned version of the input vein pattern and \bar{Y} are the collection of points in the fat version of the reference vein pattern. x_{ij} and y_{ij} are the points of the image containing the finger vein pattern (X), and the reference template (\bar{Y}), and (i,j) are the row and column position of the point. Those points have value 1 at the pattern positions and 0 otherwise.

Just as was happening with the HD method, the distance between an input pattern and the reference is not symmetric. This is the reason about why the final distance is calculated using equation (3.4), as the maximum using both patterns as reference and template alternatively.

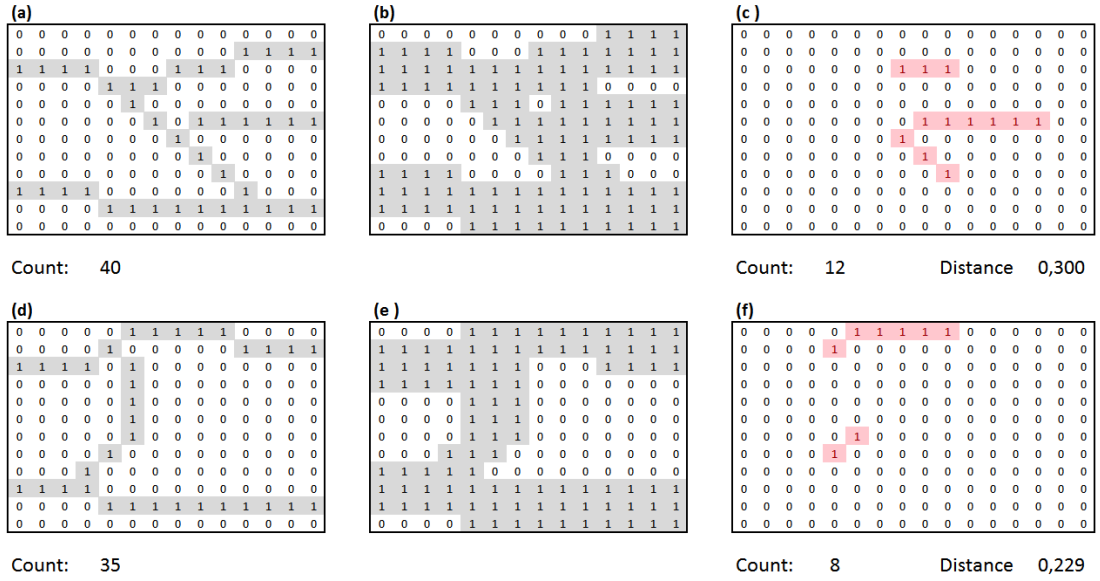


Fig 10: Simplified example of FD method.

a,d) Thinned pattern of different samples

b,e) Generated matrix that applies to the criteria

c) Matching result between a and e

f) Matching result between d and b.

3.3. Maximum Minimum Distance (MMD) method

This procedure is a novelty proposed in this thesis. It consist into realize a cross section scanning of a finger vein pattern from left to right recording the maximum and minimum distance found into two feature vectors, like can be observed in the Figure 11.

The number of scanned lines can be lower than the number of columns of the original picture to speed up the process with similar results, since the distance parameters are defined by a low speed functions. If there are less than two lines, and a maximum and minimum distance can't be defined, this is expressed like 0 in the vector, in case that there are only two lines both values are equal, and otherwise they vary from each other.

The similarity between the two vectors is defined by a custom method as it's shown in equation (3.5), reflecting the Average Vector

Similarity (AVS), that it's equivalence is the average similarity of every component of the two vectors. It proved to be more reliable and stable than applying a Euclidian or Hamming distance algorithms.

Applying this method to the vector of maximum distance and the vector of minimum distance it is possible to obtain two different results. Since the noise that the thinning process adds to the maximum and minimum distance vector it's not highly correlated both vectors add information to the system. A simplified example of this calculation for both vectors is shown in the Figure 11. It's possible to observe, that both values can differ.

$$AVS(\vec{x}, \vec{y}) = \frac{\sum_{\forall i} T_r((x_i, y_i) \neq (0, 0)) (\min(x_i, y_i) / \max(x_i, y_i))}{\sum_{\forall i} \{T_r((x_i, y_i) \neq (0, 0))\}} \quad (3.5)$$

Where $T_r(b)$, is a function such that if logical expression b is true the value it's 1, 0 otherwise; \vec{x} and \vec{y} are two vectors of the same length and x_i and y_i the values. The AVS function proposed ranges between 0 and 1, where 1 it's a full match and 0 it's a total mismatch.

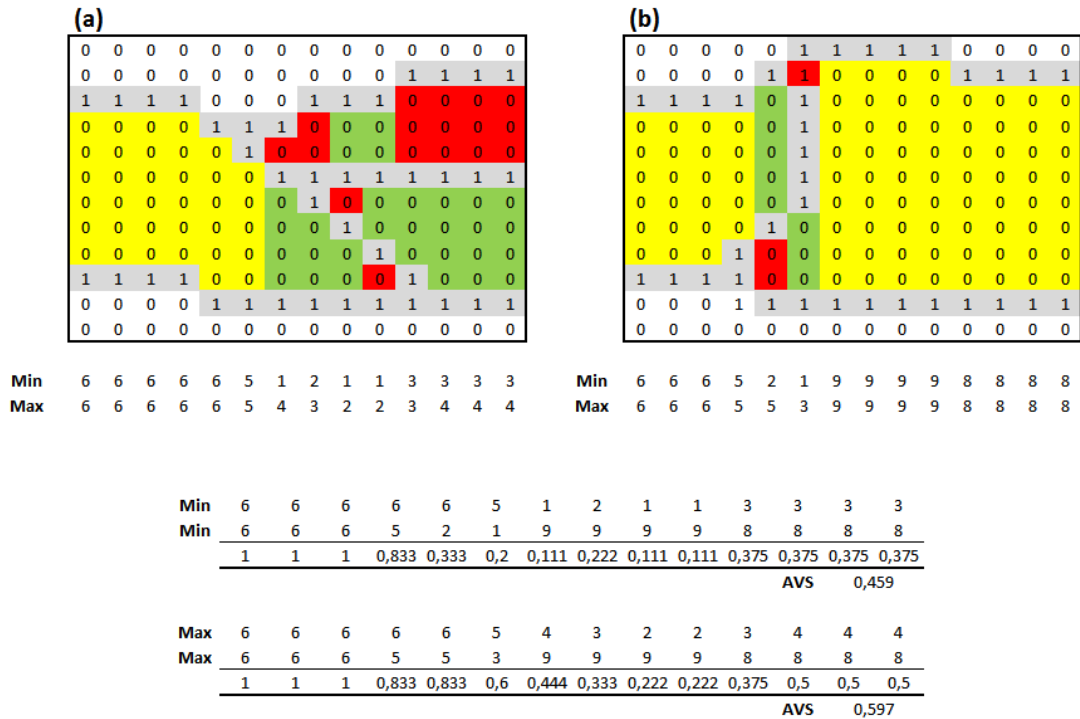


Fig 11: Example of minimum and maximum distance vector and the calculation of the AVS at each. Yellow marked the value used for both maximum and minimum, green the value used for the maximum and red the value for the minimum.

In this paper, the experimentation results were obtained using 140 transversal scans to setup the maximum and minimum distance vectors from an original image of 280 wide pixels, counting as 1 scan every 2 columns.

3.4. Score-Level Fusion Based on SVM

As pointed before, it is possible to use different values from different methods in a single identification using a fusing score system. For such propose, this thesis proposes the use of a Support Vector Machine (SVM) classifier to realize the final scoring of the identification.

An SVM classifier requires the acquisition of data for training proposes of the system and cross validation data. The training data is used as reference by the classifier to define the boundaries of the classification problem. Given that the classifier will use the training data as reference, it is not possible to test the classification performance using the training data. The data used to test the classification setup is called cross validation data, and it's propose is to define the final performance of the system.

Five different scores can be achieved using the above defined algorithms. First, the HD distances for the end and cross points, ranging between 0 and an undefined value (2 scores). Second, the FD distance between two finger vein patterns that range between 0 and 1 (1 score). And third, the AVS distance for the maximum and minimum distance vector that ranges between 0 and 1 (2 scores).

From all the possibilities, this thesis studied the next five different SVM machines setup to realize the training and testing over the same acquired samples for comparison proposes. The input data for each SVM were setup as follow:

- HD SVM uses the calculated distance for end and cross points (2 values).
- FD SVM uses the calculated distance of the FD method (1 value).
- MMD SVM uses the calculated distance by applying the AVS to minimum and maximum distance vectors (2 values).
- Multimodal FD+MMD SVM uses the values used for FD and MMD SVM machines (3 values).

- Multimodal HD+FD+MMD uses the values defined for all the other SVM (5 values).

The SVM used in this thesis can be found into *openCV* distributions. The training was realized by the *auto_training* function with the next parameters:

- **Type:** Support Vector Regression with Nu kernel.
- **Kernel Type:** Gaussian Radial Basis Function.
- **Gamma:** 0.5
- **Nu:** 0.5
- **C:** 8
- **Epsilon:** 0.001
- **Iterations:** 1000

4. Experimental Results

Once the procedure for image acquisition, the features that will be used and the matching process, it's time to define and characterize the performance of the proposed system and matching methods. The study of the results of each method will be performed from statistic point of view.

The testing was performed using 342 finger samples from 114 different fingers (3 samples per finger). 117 samples were used to train the SVM and the other were reserved for cross-validation. The 117 for training and testing define 13.338 impostor results (matching between samples from different users) and 234 genuine results (matching between samples from the same user). Consequently, the impostor case will have better representation than the genuine case.

All the samples were taken by an automatic system that requires the users to remove the finger from the device between each sample with the consequent changes in position, rotation and level of curvature of the finger and their consequent change in the finger vein pattern, the common issues found under normal use.

The system for experimentation consist of an AMD Phenom II N830 (2.1Ghz), Windows 7 SP1, 4Gb of memory and the mentioned module device for image acquisition. The data was extracted using the image acquisition described at point 2 and the features extracted were the mentioned at point 3. The environment and the results were analyzed using Matlab R2010a.

4.1. Kernel Density Estimation with Gaussian Kernel to Generate the Probability Distribution Function

The study of the statistic properties of each system requires the use of a mathematical modeling that tries to synthetize the behavior of each defined one. Such behavior in statistical systems is defined by their Probability Distribution Function (PDF). In order to synthetize it from raw data, this thesis uses the Kernel Density Estimation (KDE).

Kernel Density Estimation (KDE) is a non-parametric method that allows to synthesize an approximated Probability Distribution Function (PDF) from an output samples collection of the process.

The method is based in the use of a Kernel function that must have be symmetric (Equation 8) and have unitary area (Equation 9).

$$1 = \int_{-\infty}^{\infty} K(u) du \quad (4.1)$$

$$K(u) = K(-u) \forall u \quad (4.2)$$

Given $x(0), \dots, x(N-1)$ the known output sample collection from a determined random variable X , and $K(u)$ the kernel function used, it's possible to get a synthesized version of the PDF of X as the average addition of all the kernel functions localized each one at each known sample.

$$\hat{f}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \left(\frac{K(x-x_i)}{h} \right) \quad (4.3)$$

Equation 10 shows how it's calculated the value the value for each point of the PDF ($f(x)$) based on the collection of samples using the KDE method, where h is the wide of the selected kernel function.

In this thesis, the selected kernel function has been a Gaussian function in coherency with the SVM. The final expression used to synthesize the PDF is shown in the Equation 11.

$$\hat{f}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_j - u_{ij})^2}{2\sigma_j^2}} \quad (4.4)$$

4.2. KDE Application for Method Modeling

Each method can be modeled as two different Probability Density Functions (PDF) that define the results obtained between samples of the same user (Genuine Distribution) and samples of different users (Falsification Distribution). Given the genuine and falsification distributions it's possible to analyze the behavior and characteristics of each proposed method. An ideal system would have disjunctive distributions in order to be able to easily setup a threshold that differentiates the area associated with Genuine and Falsification cases. The Figure 12 shows the drawing of such distribution for each proposed method.

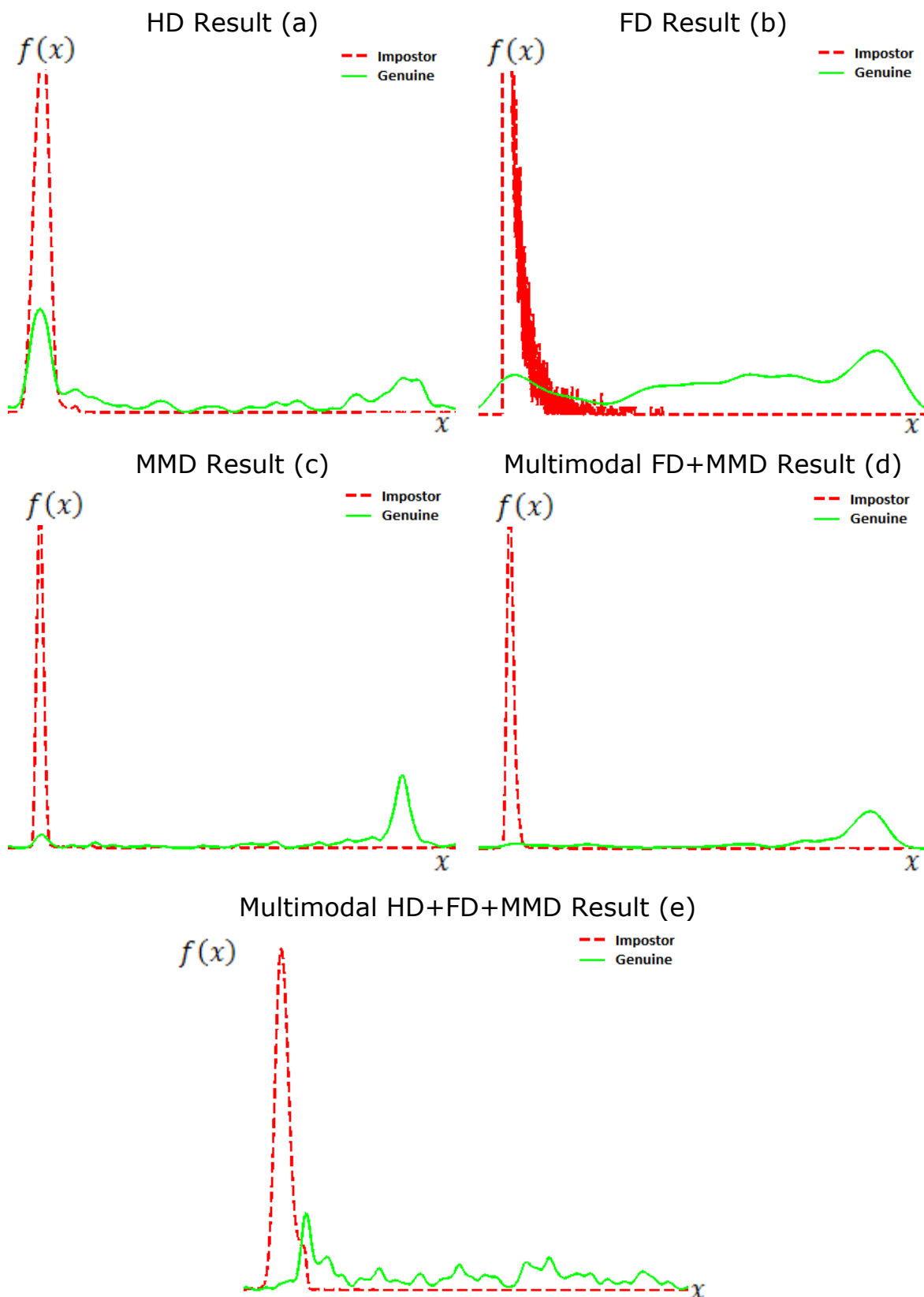


Figure 12: Probability density functions of each proposed SVM

In the Figure 12 has been marked the top-left corner (the training data) to differentiate how the SVM scoring performs with the training set and the cross validation data.

Figure 12 (a), the HD method shows exactly what it's expected to not be seen, there is an important overlap between the Genuine and Impostor distributions that implies the impossibility to achieve high degrees of accuracy. The system fails to score differently for a large number of genuine cases and impostor cases. This behavior can be assumed to be related to the instability of the thinning process that randomly deletes or adds ends and cross points from the finger vein pattern and point to that as the cause for the low performance previously observed at the ROC curves. However, the impostors distribution it's clearly defined.

Figure 12 (b), the FD method also shows problems to completely avoid the overlap between the impostors and genuine distribution, however in this case the peak observed under the main lobule of impostors PDF has less importance, pointing the possibility to achieve higher degrees of accuracy. In this case, the noise of the thinning process can be observed as a wide distribution of the genuine scores, failing to perform a defined Gaussian form separated of the impostor distribution.

Figure 12 (c), the MMD method shows a minimal peak under the impostors distribution. This method achieves a high degree of isolation between the impostors and genuine distributions showing clearly two different centers for two different Gaussian distributions. That points to the possibility to successfully avoid the false identification of the users given the high margin of distance between both distribution samples. Once more it is possible to observe some overlap, with some samples being unable to be differentiating between genuine and impostor case. This graph proves that the MMD method has a higher degree of resistance to the thinning noise than the FD and HD method.

Figure 12 (d), Multimodal MMD+FD method shows that the combination of two methods can significantly decrease the peak shown by the genuine PDF down the main lobule of impostors PDF. However, this is achieved sacrificing the elegant peak found in the MMD PDF for a wider genuine distribution. However, the average results as shown a significant boost of accuracy and identification rates under the same conditions.

Figure 12 (e), Multimodal HD+FD+MMD seems to show that to include every possible method into a multimodal system doesn't implies

that the end result will be improved. As shown in the graph, the peak under the impostor PDF from the genuine PDF has been greatly decreased as with the last multimodal method, however, in this case the tradeoff have been a greatly increase in the spread of the distribution and the translation of the main peak of the genuine PDF near the main peak of the impostor PDF.

4.3. Genuine Acceptance Rate and False Acceptance Rate calculation

In order to calculate the Genuine Acceptance Rate (GAR) and False Acceptance Rate (FAR), it's possible to use the PDF calculated using the KDE method. The final GAR and FAR for a given decision threshold can be calculated as shown in Equations (4.6) and (4.7), applying the common formula for calculating the probability at a given point of a density function, Equation (4.5).

$$F(x) = \int_{-\infty}^x f(x)dx \quad (4.5)$$

Where x is the decision threshold and $f(x)$ is the PDF of the random variable. The FAR and GAR will be calculated as shown in Equation (4.6) and Equation (4.7).

$$FAR_M(x) = \int_{-\infty}^x far_M(x)dx \quad (4.6)$$

$$GAR_M(x) = 1 - \int_{-\infty}^x gar_M(x)dx \quad (4.7)$$

Where M is the chosen method and gar_M and far_M are the PDF of GAR and FAR of such method. However, given that the threshold defines non valid matches, the final probability for GAR needs to be subtracted from 1.

Using a support vector regression type allow us to setup a threshold for the final independently from the SVM. This kind of setup only works as a scoring machine that tries to output a numerical score. The parameter "C" marks the threshold selected that the automatic setup will try to achieve as optimal point, however it's possible to use any desired threshold value. Once the training data and parameters of the SVM have been choose, it is possible to use it as a method to score the similarity. It's required for the methods that have more than one value and especially useful for the multimodal systems. The next step will be to

compare the output data from the matching of the 225 samples reserved for cross-validation.

In order to compare different identification systems it's common to draw the Receiver Operating Characteristics (ROC) curves. Those curves are defined using the Genuine Acceptance Rate (GAR) and False Acceptation Rate (FAR) at each axis of a two dimensional graph for a range of different threshold values in the identification score. These curves are especially useful as a graphical representation of the trade-off between GAR and FAR. A desired system would be able to have a 100% GAR with a 0% of FAR. In the following graphs, that would be represented as a curve that it's closer to the top-left corner.

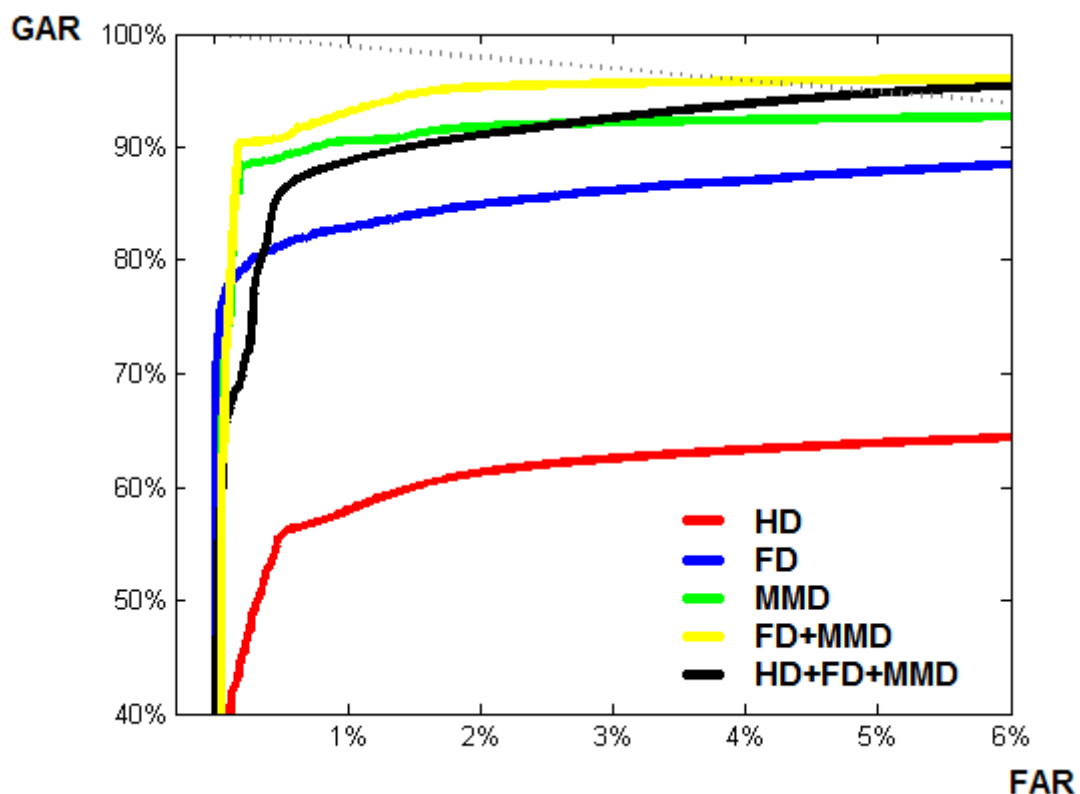


Fig 13: ROC curves for the different methods analyzed

- a) Red: HD
- b) Green: FD
- c) Blue: MMD
- d) Black: Multimodal HD, FD, MMD
- e) Yellow: Multimodal FD, MMD
- f) Dotted line: EER

Figure 13 (a), HD method shows to be a low performer, low GAR probability and require accepting some level of FAR in order to be able to

work. Other studies got much better results with this method. We assumed that this is a problem related to the acquisition of the pictures and instabilities in the finger vein pattern extraction proved to be a challenge for this method under normal conditions.

Figure 13 (b), FD method has a level of performance but experience difficulties to achieve high levels of GAR with a saturation point near the 90%. However, this method has the highest GAR probability at very low levels of FAR <0.1%.

Figure 13 (c), MMD method has balanced performance. It's possible to setup this method for different proposes. The GAR ranges between 85% to 95% and the FAR does between 0.25% and 5%. It's possible to setup the system for authentication proposes (High GAR at acceptable levels of FAR) and identification (Very low FAR, with acceptable levels of GAR).

Figure 13 (d), Multimodal FD+MMD method behaves in a similar way to the MMD. However, the saturation point of the GAR it's near the 95%, similar to MMD but for lower levels of FAR.

Figure 13 (e), Multimodal HD+FD+MMD method is also a balanced performer with worse behavior than the multimodal FD+MMD method for low levels of FAR but it's able to achieve higher GAR after the 5% FAR mark.

Table (4.1) compares the approximated values of performance focusing to setup a system with an acceptable level of FAR (0.5%), and noted the level of Equal Error Rate (EER), in other words, the point where FAR equals to False Rejection Rate (FRR), the complementary value of GAR.

Table 4.1: Comparative values of the different methods studied.

Method	GAR	FAR	EER
HD	65%	0.5%	28%
FD	87%	0.5%	9.7%
MMD	93%	0.5%	7.1%
Multimodal FD+MMD	96%	0.5%	5.1%
Multimodal HD+FD+MMD	84%	0.5%	5.3%

4.4. Combinatory for N samples problem

During the user registration process it's possible to require the user to input more than one sample for each finger that the user would like to register. The number of validations required within the total of acquired samples of the user to perform identification can be defined.

The use of the different samples in order to enhance the system final results will vary the improvement obtained by taking several samples of each user's finger. This may increase the performance of the system compared with the plain probability of each method.

In order to calculate the new False Acceptation Rate (FAR) and Genuine Acceptation Rate (GAR), it's required to rely into combinatory systems study.

Both FAR and GAR are a binary probability systems just like the problem of the probability to calculate the probability to obtain N faces after Y coin flips. The equation that define the output probability of the combinatory system it's defined by the Equation (4.8).

$$P_N = 1 - \sum_{i=0}^{N-1} (1 - P_0)^{Y-i} P_0^i \quad (4.8)$$

In this thesis the testing and developing has been realized using 3 samples of each finger. The final probability for FAR and GAR can be calculated as shown in Equation (4.9) and (4.10) respectively.

$$FAR_{MN} = \int_{-\infty}^x (1 - \sum_{i=0}^{N-1} (1 - far_{M0}(x))^{3-i} far_{M0}(x)^i) dx \quad (4.9)$$

$$GAR_{MN} = \int_{-\infty}^x (1 - \sum_{i=0}^{N-1} (gar_{M0}(x))^{3-i} (1 - gar_{M0}(x))^i) dx \quad (4.10)$$

Where M is the chosen method, N is the required number of positive matches required to perform identification. Given 3 samples per user, N ranges from 1 to 3. Fig 13 shows the different Receiver Operating Characteristics (ROC) for the FD+MMD multimodal method for one, two and three matches requested.

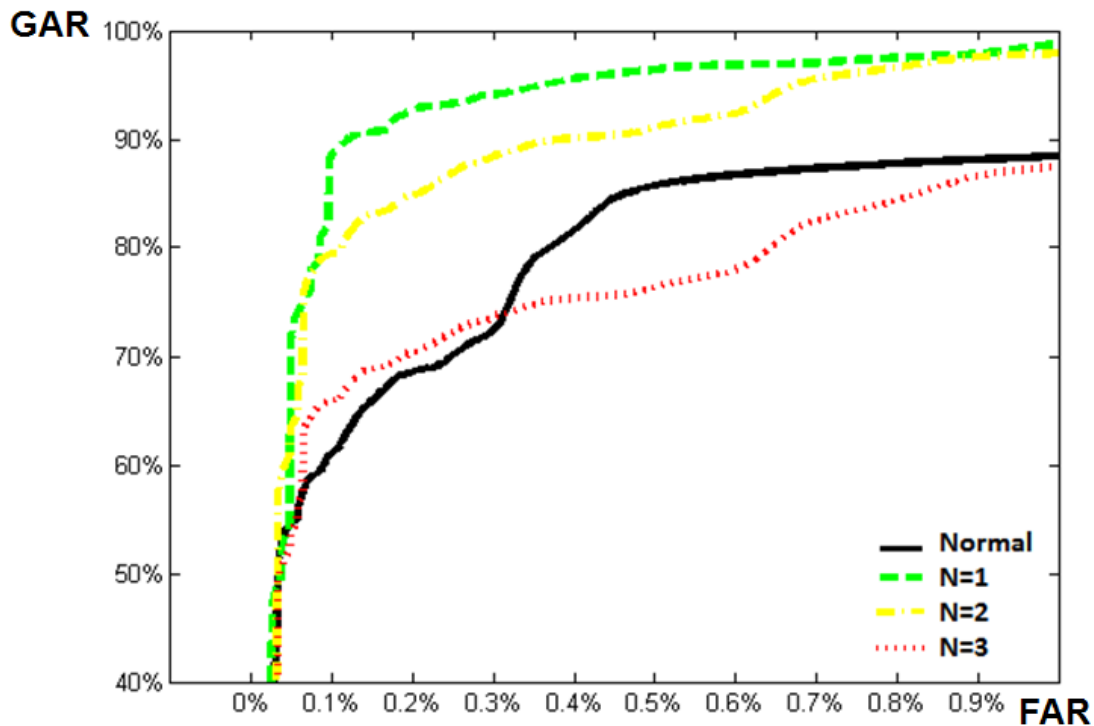


Fig 14: ROC curves for the different possible values of N with FD+MMD multimodal method.

As shown in Figure 14 requiring only 1 match from 3 samples in order to perform identification seems to draw the best ROC curve. It is also possible to observe the dramatic improvements of GAR that can be obtained by requiring multiple samples per user after analyzing the combinatory system properties.

Table 4.2: Equal Error Rate comparison table for each possible value of N for each studied method.

Method	N=1	N=2	N=3
HD	5,96%	12,5%	25,24%
FD	0,85%	2,72%	10,7%
MMD	0,43%	1,16 %	7,01%
Multimodal FD+MMD	0,43%	0,88%	4,48%
Multimodal HD+FD+MMD	1,02%	1,51%	6,72%

The Table (4.2) shows that using different values of N may incur into different levels of performance. Not every method has the same behavior in front of the number of requested matches for identification as should be expected given the differences found between each method PDF. However, all the studied methods proved to have better properties for N equal to 1.

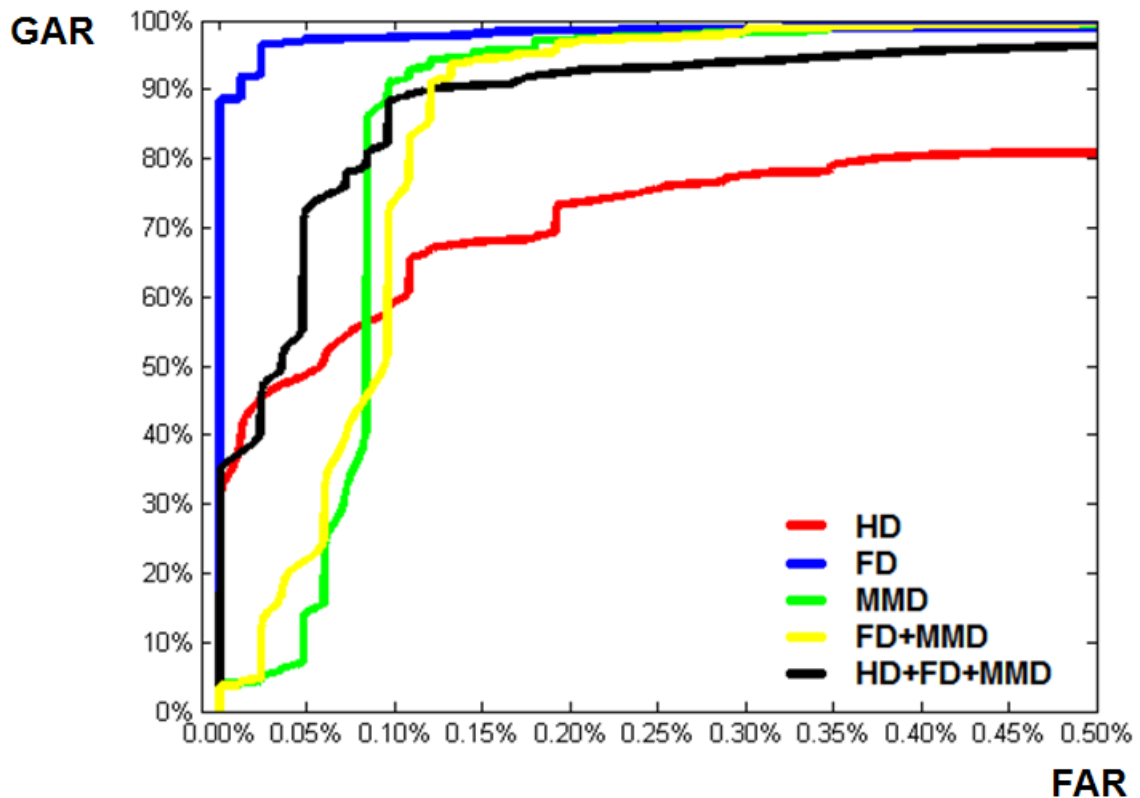


Fig 15: ROC curves for each method with N=1.

Figure 15 shows something that couldn't be observed in the previous ROC curves. Even if the MMD and Multimodal FD+MMD had higher levels of performance as one-to-one matching systems than FD or HD methods, this graph shows that even if not differentiable in the MMD method PDF graph seems that we have some probability for MMD method located down the main lobule of the genuine PDF just as we had a peak down the impostors PDF main lobule that expands its presence to the multimodal method. That's something that couldn't be observed before the GAR enhancement that implies the combinatory system for identification, showing that for extremely low FAR requirements, FD, HD or Multimodal HD+FD+MMD methods may perform better.

This behavior opens the door to the application priorities to select the method. Different applications may have different levels of FAR and GAR requirements. Applications where the user information is stored into a local support with its own biometric parameters may accept higher FAR given the one-to-one identification process. However, when the information is stored into a locally stored or networked database and the user tries to perform an identification that should output the user's identity, very low levels of FAR are required since the probability of a random

person being identified as someone else in the database increases with the database size as can be calculated using the Equation (4.11).

$$P_N = 1 - (1 - P_0)^N \quad (4.11)$$

Where N , is the desired maximum number of users in the system size and P_0 is the One-To-One FAR.

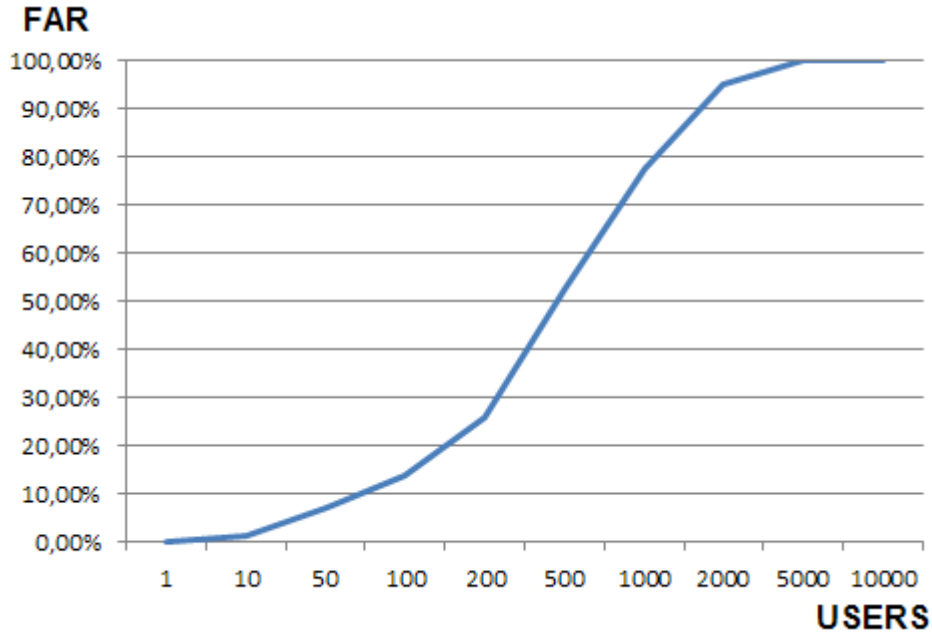


Fig 16: FAR of an Identification System that has method setup with a 0.15% One-To-One FAR.

Figure 15 shows that despite very low identification rates, it's still a challenge the possibility to use such methods for large databases. Systems designed to allocate more than 500 users require very low levels of FAR. A linear approximation (Equation (4.12)) may be used to calculate the required One-To-One FAR given a target FAR level for the Identification System.

$$FAR_0 = FAR_T / N \quad (4.12)$$

Where N is the number design target database size and FAR_T is the target FAR. This approximation remains valid for lower than a 5% FAR_T .

Table 4.3: Comparative values of the different methods studied under the combinatory system for N=1.

Method	GAR	FAR	EER
HD	65%	0.15%	5,96%
FD	97%	0.15%	0,85%
MMD	95%	0.15%	0,43%
Multimodal FD+MMD	94%	0.15%	0,43%
Multimodal HD+FD+MMD	90%	0.15%	1,02%

Table (4.3) compares the different methods under the combinatory system for N=1 and shows that the GAR and FAR levels have been significantly improved, and EER has been greatly reduced.

5. The proposed Solution

The solution proposed has been developed as a distributable system with a common database that may be updated and accessed by different clients from local or non-local networked locations. The solution relies in a data server that can be remotely accessed to acquire the registered users of the system and a GUI Client program that manages the user interactions: identification and registration; and operations: delete/modify users, benchmarking and setup.

In this thesis has been designed a server-client system were the computing tasks are realized in the client.

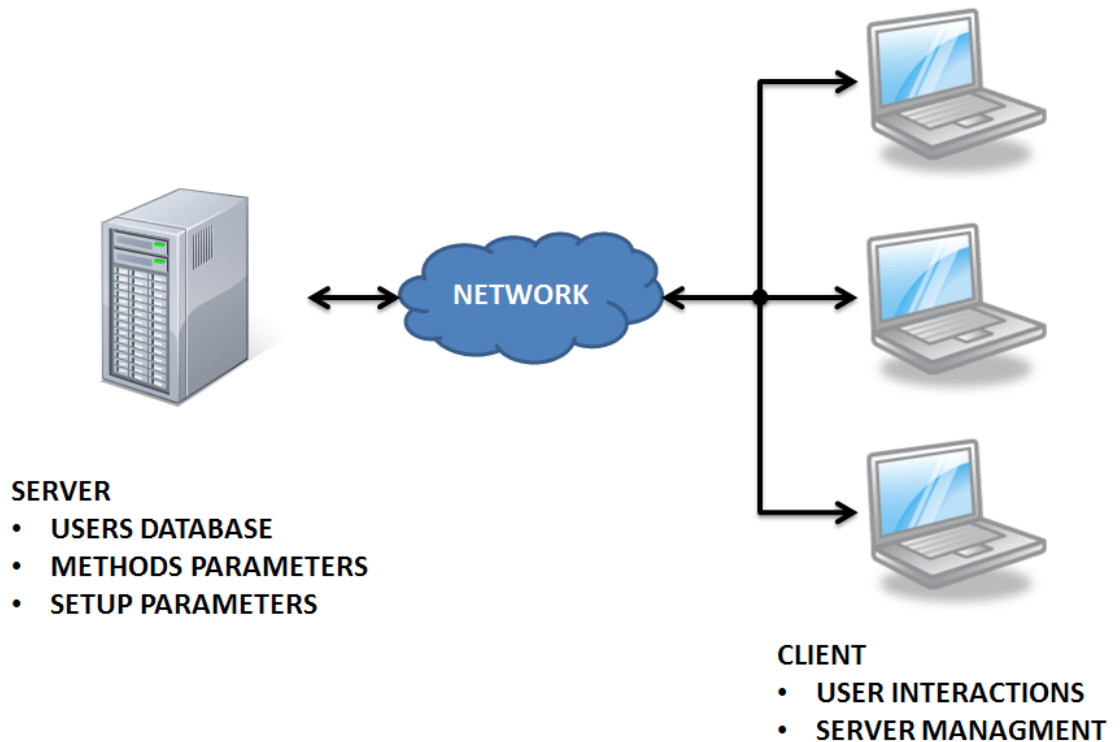


Fig 17: Server-Client system architecture and functionality.

Figure 17 shows the distributed functionality and responsibilities of the proposed system.

5.1. The Server

The server should contain the users' database, methods parameters and setup parameters. In order to store such information the proposed system will rely into an SQL Database Server. The required information

will be stored in the database. Figure 18 shows the SQL Database diagram used

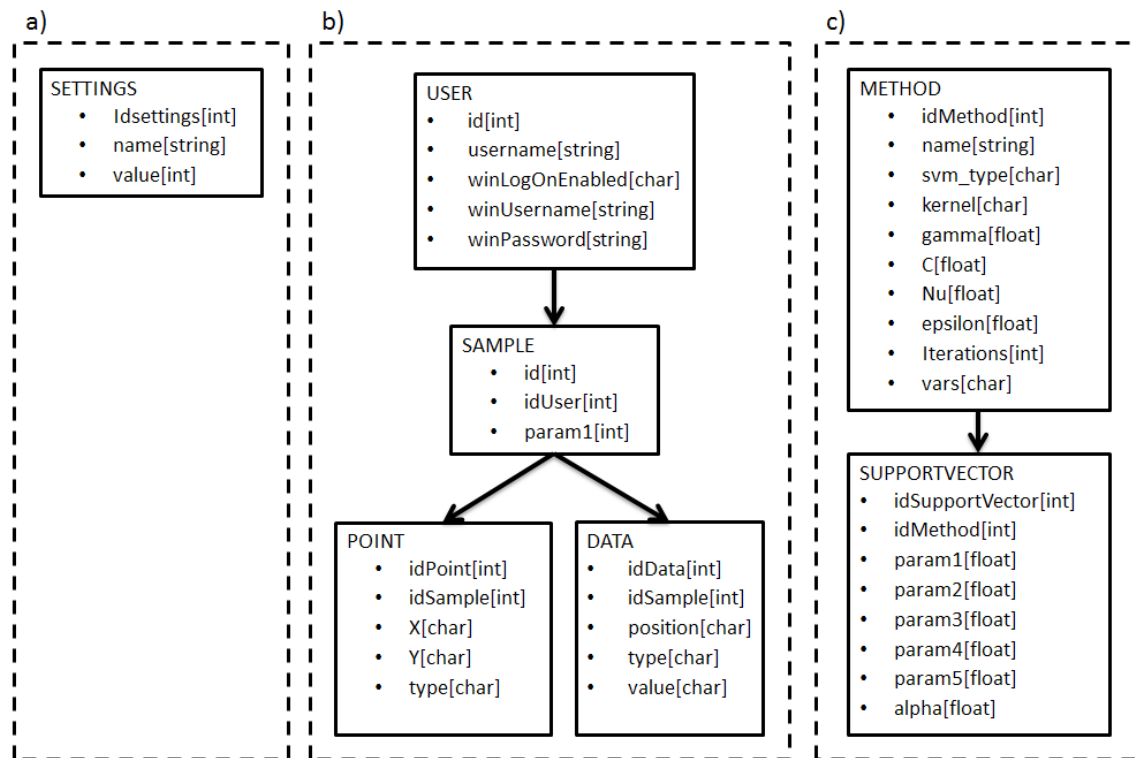


Fig 18: SQL Database for users, settings and Support Vector Machines setup. a) Settings table. b) Users related tables. c) Methods related tables.

Figure 18 shows the tables used in the database to store all the information required by the system. The data is structured in a tree diagram divided in 3 categories: Settings (a), Users (b) and Methods (c).

a) Settings: the settings category consists of a single table (SETTINGS) that contains the setup values of the system. Each setting is stored under an id number, name and value inside that table.

- **idsettings:** Contains the identification number of the setting. The client will use this id to request the value of the setting.
- **name:** Contains the name of the setting. The client doesn't require this but allows the administrator to not require remember the id's to manage the database.
- **value:** Contains the setting value.

b) Users: The user's information is stored into a tree diagram structure that consists of 4 tables: User, Sample, Point and Data.

- **User:** Contains the parameters directly related to the user, this class contains also the Windows LogOn information that was included in order to setup a demo. Such demo and explanation aren't included in this thesis.
 - **id:** The identification number of the user inside the program.
 - **username:** The name that the user inputs during the registration procedure.
 - **winLogOnEnabled:** Can only be 0 or 1. 0 represents false and 1 represents true. This value will be used by the system in order to understand if this user has winLogOn information setup.
 - **winUsername:** Stores the Windows account username used to attempt a LogOn in the demo.
 - **winPassword:** Contains the Windows password associated to the Windows account setup used to attempt a LogOn in the demo.
- **Sample:** Contains the information of the samples. Each sample is related to one user and can contain additional information.
 - **id:** The identification number of the sample.
 - **idUser:** The identification number of the user which the sample is related.
 - **param1:** Stores data about an arbitrary parameter of the samples. In the final setup was used to acquire the acquisition time used during the samples taking process.
- **Point:** Stores the information of a single point of the picture. The points are stored using positive Cartesian coordinates. This is the data required for HD and FD methods.
 - **idPoint:** The identification number of the point.
 - **idSample:** The identification number of the point which the point is related.
 - **X:** Horizontal axis location of the point.
 - **Y:** Vertical axis location of the point.
 - **type:** The nature of the point stored. There is three kind of points that can be stored
 - **End Points:** The end points used for HD method.

- **Cross Points:** The cross points used for HD method.
- **White Points:** All the points that draw the finger vein pattern after all the threshold, thinning and pattern cleaning process.
- **Data:** Stores the information stored in the Minimum and Maximum distance vectors for MMD method.
 - **idData:** The identification number of the data.
 - **idSample:** The identification number of the point which the point is related.
 - **position:** Location of the data inside the vector.
 - **type:** Boolean value that marks the source vector of the data. It can only be two options: Minimum and Maximum (0 and 1).
 - **value:** The numeric value of the data.

c) Methods: The information used to setup each Support Vector Machine (SVM) is stored in this category structured in two tables: Method and SupportVector.

- **Method:** Contains the parameters and name of the method for each method.
 - **idMethod:** The identification number of the method inside the program.
 - **name:** The name of the method.
 - **svm_type, gamma, C, Nu, epsilon, Iterations, vars:** SVM setup parameters.
- **SupportVector:** Information about each of the Support Vectors that define the SVM.
 - **idSupportVector:** The identification number of the Support Vector.
 - **idMethod:** The identification number of the method which the Support Vector is related.
 - **param1, param2, param3, param4 and param5:** Stores the 5 possible values of the vector's components.
 - **alpha:** alpha value associated with the Support Vector.

The database will have a defined size for the Settings and Methods once the system is running. However, the number of users will vary with time. Users can be added and deleted from the database. In this works setup the number of samples per user was defined at 3. Each sample will

contain an undefined number of cross, end, white points and a defined number of data:

Table 5.1: Estimated database size from a registered user (finger).

Field	Type	N	Size(Bytes)
USER			
id	Int	1	4
username	String	1	32
winLogOnEnabled	BOOL	1	1
winUsername	String	1	32
winPassword	String	1	32
SAMPLE			
id	Int	3	4
idUser	Int	3	4
param1	Int	3	4
POINT			
idPoint	Int	3000	4
idSample	Int	3000	4
X	Char	3000	1
Y	Char	3000	1
type	Char	3000	1
DATA			
idData	Int	840	4
idSample	Int	840	4
position	Char	840	1
Type	Char	840	1
value	Char	840	1
TOTAL			42377

With this information it's possible to calculate the database space requirements of the system as shows Equation (5.1).

$$S = K + 42377N_u \quad (5.1)$$

Where K is the database size after SVM training and system setup and N_u the number of registered users (fingers) and S is the total estimated size of the database. During the testing, the value of K was 383Kbytes. A database with 100 users would be estimated to require

4,6MBytes. If the users were to be stored as images, it would be required to use BMP format to avoid compression artifacts. Each user would consist into 3 pictures of 280 x 180 (50.400) pixels and the size per user would be estimated around 152Kbytes. The same 100 users' database would be estimated at 15.6Mbytes. This calculation proves that the system achieves lower storage requirements despite the database overhead.

5.2. The Client

The client should target the function of server management and user interactions. The client will be divided into four main points.

- **SQL Database Management (C++):** SQL Database must be represented into the client. It will be a key to speed up the system performance.
- **Image Processing (C++):** The client must contain all the required code and intelligence to obtain the finger vein pattern from the source device (NIR Camera).
- **Matching Functions (C++):** The system must have built-in the algorithms and intelligence related to user matching.
- **User Interface (C++/CLI, .NET):** The user interface should offer the controls to manage the SQL Database server (register and delete users, modify settings and update SVM's parameters), realize performance testing (matching statistics) and real world checking test.

5.2.1. SQL Database Management

The database will be cached and loaded during the client start-up. This is a required process given that the computing tasks will be performed at client side. If SQL Database information weren't to be cached, our system would be required to load all the users data each time an identification is requested. Given a 100 users' database as proposed in the previous database size example and a 100Mbps Ethernet connection, it would require around 500ms to request all the users' information without including the queries overhead. The time required for an identification would be similar to the time required for the startup with the setup proposed, calculated around 4m20s for 114 users, around 2,2s/user. However, the time required for identification with the proposed system it's down to 1,2s for a 114 user's database in the testing system or an

average time of 10ms/user, 220 times faster. It also avoids requiring a reliable and strong data connection to our server, since it only affects our starting time. The class diagram that defines the SQL Database used in our system is shown at Figure 19.

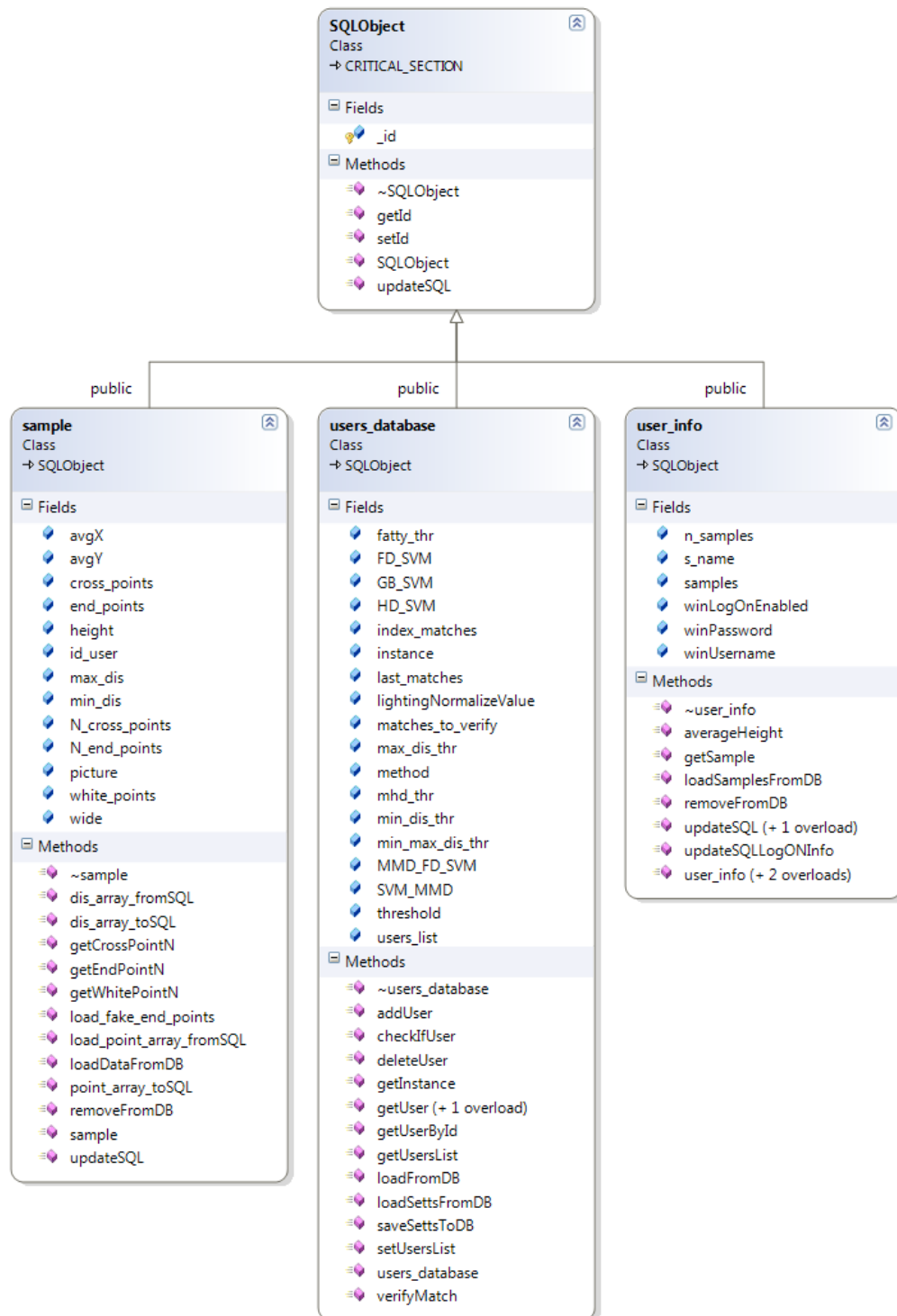


Figure 19 Class diagram with the required classes used in the proposed client to cache the SQL Database.

The system will be structured into 3 main classes that inheritance from a main class that represent any SQL Object in our system:

- **SQLObject:** Contains the common methods and variables commonly used like *id* and *updateSQL*.
- **users_database:** Contains all the information that was stored in the database: Settings, Users and Method's parameters.
- **user_info:** Defines the information of a user in our system, such as the name and samples.
- **sample:** Contains all the information of a sample, such as cross and end points, the pattern picture, etc.

Some values and methods are used for pre-processing and alignment between finger vein patterns such as "**avgX**" and "**avgY**". The average position of the finger vein patterns it's used to center the vein pattern and increase the matching probability.

It's also important to notice the existence of a method called "**load_fake_end_points**". Such method will be used to load the points of a finger vein pattern located in the edge of the image as shown in *Figure 20*. This is of special importance when a finger vein pattern doesn't reports any end point and it may incur into an undefined distance problem between pictures with no end points and pictures that own end points within HD method or two pictures that doesn't own. This is specially required for the fusion of HD with MMD and FD into a multimodal machine, given that it's not possible to leave one of the input values as undefined without requiring skipping the matching.

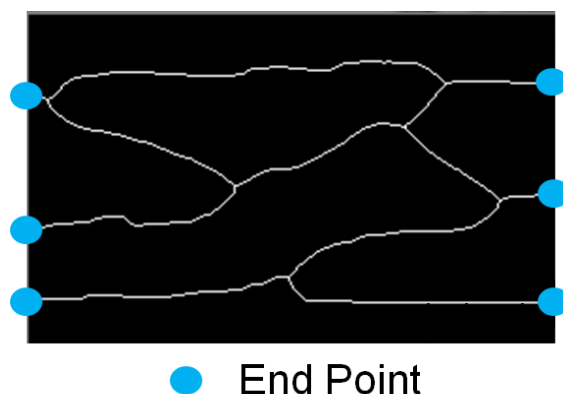


Fig 20: End points found at a finger vein pattern after "**fake_end_points**" method is called.

5.2.2. Image Processing

All the required methods for image processing will be contained inside a single class called "*image_processing_functions*", which class diagram is shown in Figure 21.

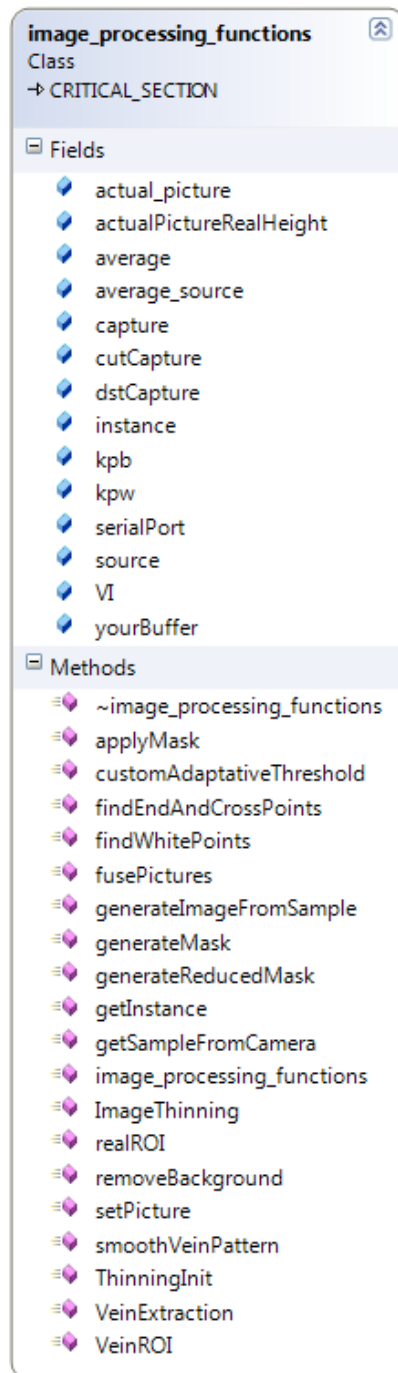


Fig 21: Class diagram of the class used for all the tasks related with image processing.

This class will contain the last finger vein pattern obtained from the source camera ("**actual_picture**"), the video input source ("**VI**"), the serial port manager required to regulate the Near Infra-Red illuminator ("**serialPort**") and the main routine of the client to start the image acquisition process ("**getSampleFromCamera**"). These are the four key points of this class. It's possible to use this class isolated from the system In order to obtain finger vein patterns from a source camera or internally modify all the parameters. The only method that can be accessed from another class is the main acquisition process "**getSampleFromCamera**", and the only two public parameters are the "**actual_picture**" and "**actualPictureRealHeight**". These parameters define the actual vein pattern found and the illumination used to obtain it respectively. The misleading name of the second parameter ("**actualPictureRealHeight**") comes from the fact that the illumination required to obtain a clear pattern is proportional to the finger height. Some of the declared values and methods are legacy or test methods.

5.2.3. Matching Functions

The matching functions and all the methods related to the matching, benchmarking, training of the different support vector machines and data extraction from the source pictures, maybe found at the class "**functions_library**". Figure 22 shows the class diagram of the class.

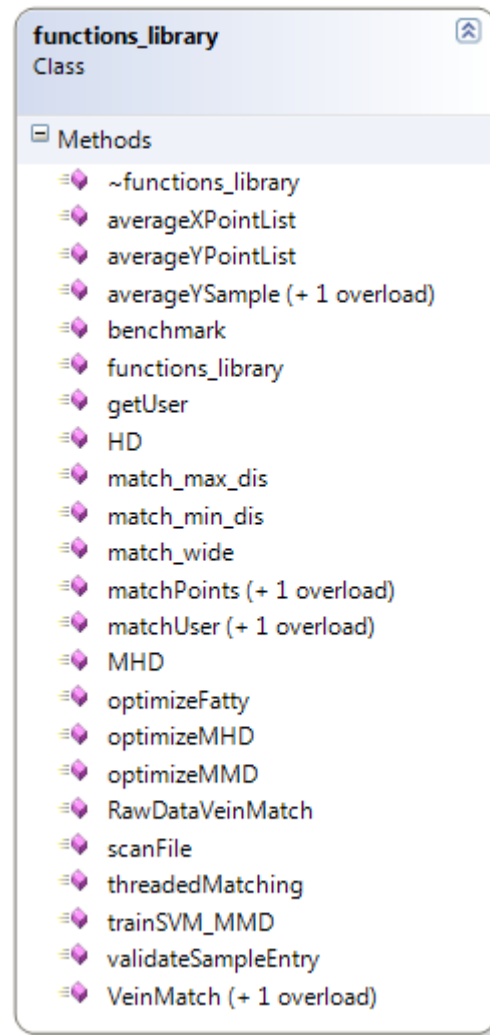


Fig 22: Class diagram for "*functions_library*".

As shown in the class diagram there is a collection of different methods used. As shown, there are no values contained in this class. This class contains only static methods and functions. The key points from of this class are:

- **Benchmark:** creates a CSV (*Coma Separated Values*) file that can be used with Matlab for statistics extraction or open using Excel in order to obtain a visible representation about what's happening with each method.
- **getUser:** returns the user found in the database that matches the given finger vein pattern. If no match it's found returns NULL (0). This method is used for identification proposes.
- **scanFile:** returns a sample that it's created from the given finger vein pattern.

- **matchUser:** matches a given sample or finger vein pattern with a given user and reports true or false as matching result. This method is used for validation proposes.

5.3. User Interface

The user interface implemented during this thesis has been realized using C++/CLI, .NET language. This gives us the possibility to keep the key classes being C++ (native code) with fast processing speed but at the same time give us the "easy to code" UI layer that .Net Framework provides for Windows.

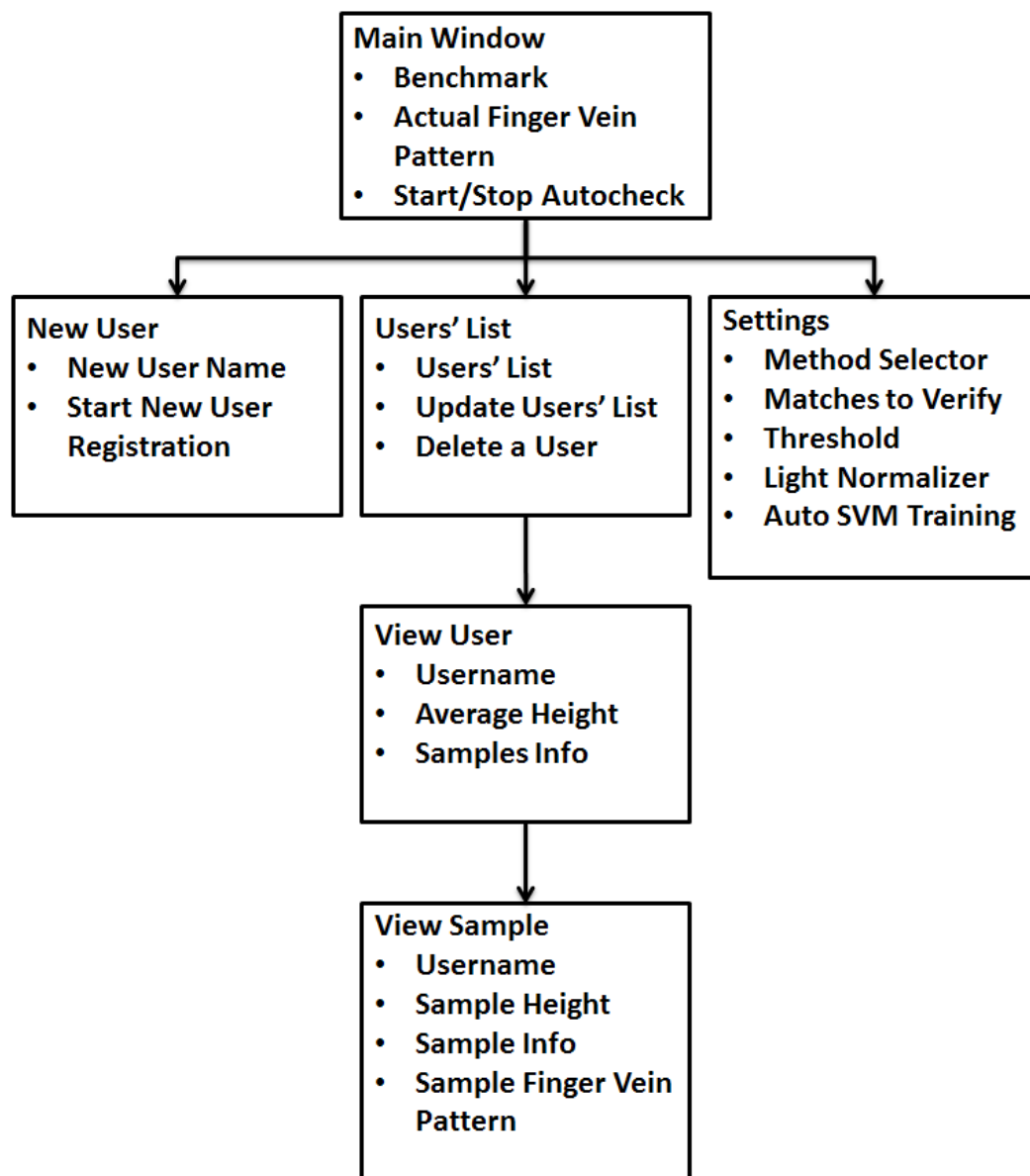


Fig 23: User interface tree diagram.

Figure 23 shows the user interface tree diagram. The squares represent each window of the program and the access to secondary windows is related with an arrow. As example, the set of arrows relating the "Main Window" with "New Users", "Users' List" and "Settings", means that we can open those windows from the "Main Window". The list under the window's name implies the actions and information displayed in the windows. As example, "Users' List" window shows the users' list information and has access to update the users' list from the server and delete a user from the server.

5.3.1. The Main Window

The main window contains the image from the last finger vein pattern captured with the camera and the identification result performed using the actual configuration of the system against the users' list loaded from the server.

The main window allows the access to perform a "benchmark button" that will create a "coma separated values" (CSV) file that will contain the matching results used to characterize the system and an "Start/Stop Autocheck" button in order to stop the real-time identification process.

From the main window we can access the settings, the users' list and the new user window.

Figure 24 shows an image of the "Main Window". This window contains 5 buttons to access different areas of the application such as registration of new users and users management and perform a benchmark of the full database, a figure frame that shows the finger vein pattern that is being acquired and a text box containing the username of the registered vein pattern if it's found. Otherwise it reports "User Not Found".

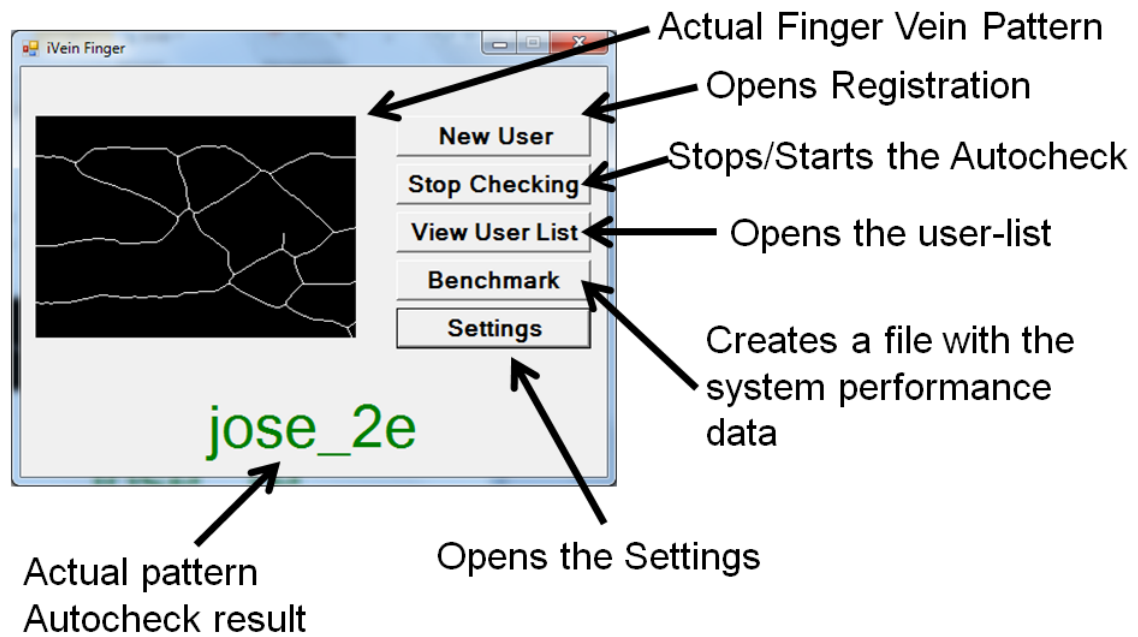


Fig 24: Main Window.

5.3.2. New User

New User window contains the actual finger vein pattern just as the Main Window does a notification area that guides the user during the user registration process and a text input area for the desired user name. The registration process starts when the user presses the "Take User" button.

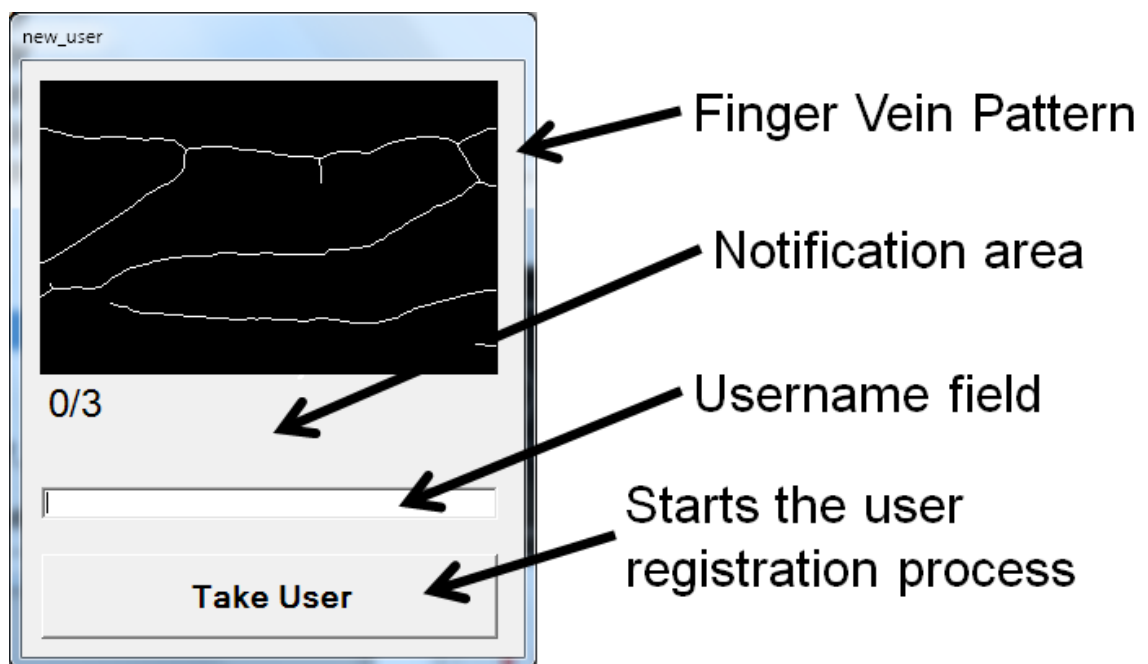


Fig 25: New User Window.

Figure 25 shows an image of the “New User” window. The registration process was designed to be automatic. The user is requested to retire the finger from the device between samples. A threshold in the level of illumination in the input picture of the camera triggered the detection of a finger vein pattern and a timer between the first vein pattern found after the user removed the finger from the device was setup to trigger the initial capture procedure.

5.3.3. Settings

The settings window provides access to the main four parameters of the system:

- **Identification Method:** this value will identify the method that will be used for the identification performed by the Autocheck routine of the system, the recommended selection is FM+MMD.
- **Matches to Verify:** this is the number of matches required during the identification. The recommended value is 1.
- **Threshold:** the decision value for the matching. A study using the benchmark data will be required in order to define this value objectively.
- **Light Normalize Value:** as shown before, this is the value used for the normalization of the lighting. The recommended value is 0.75, but we may increase this value if we have problems with dark areas or decrease it if we detect problems of over-exposition.

The settings window also includes the “AUTO” button. This button performs the training of the different SVM for each method and stores the SVM setup in the database.

Figure 26 shows an image of the “Settings Window” window

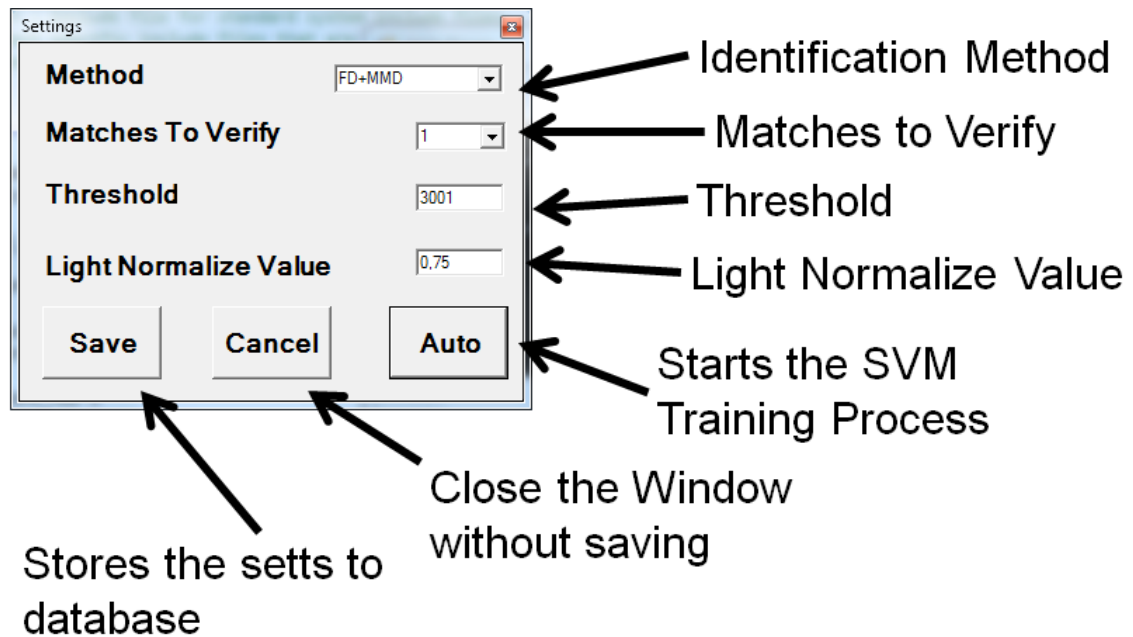


Fig 26: Settings Window.

5.3.4. Users' Window

The users' window main function is to show the list of registered users in the system, access the possibility to view more information about the user and perform basic operations such as update the user list from the server and delete a user from the server.

Figure 27 shows an image of the "Users' List" window. It contains the list of users ordered by id and showing the username as identifier. It contains the functions required to update the user list from the server, delete a user, view the samples of given user and manage a user's LogOn information.

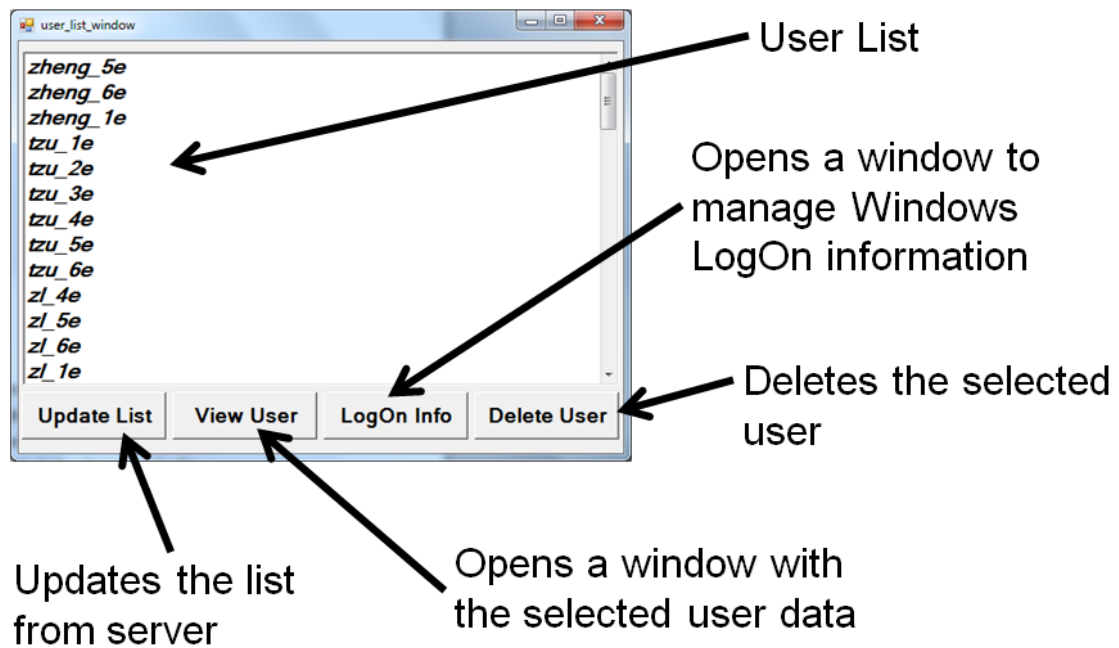


Fig 27: Users' List window.

5.3.5. View User Window and View Sample Window

The user window shows the parametrical data stored for each sample of the user and the average height of all the samples of the user. It also has the possibility to open a window that contains the information of a single sample, with only the parametrical, the height and the finger vein pattern from a given sample.

Figure 28 shows an image of the "View User" window. It contains the username, the extra defined parameter and the possibility to show any sample by itself in a separate window.

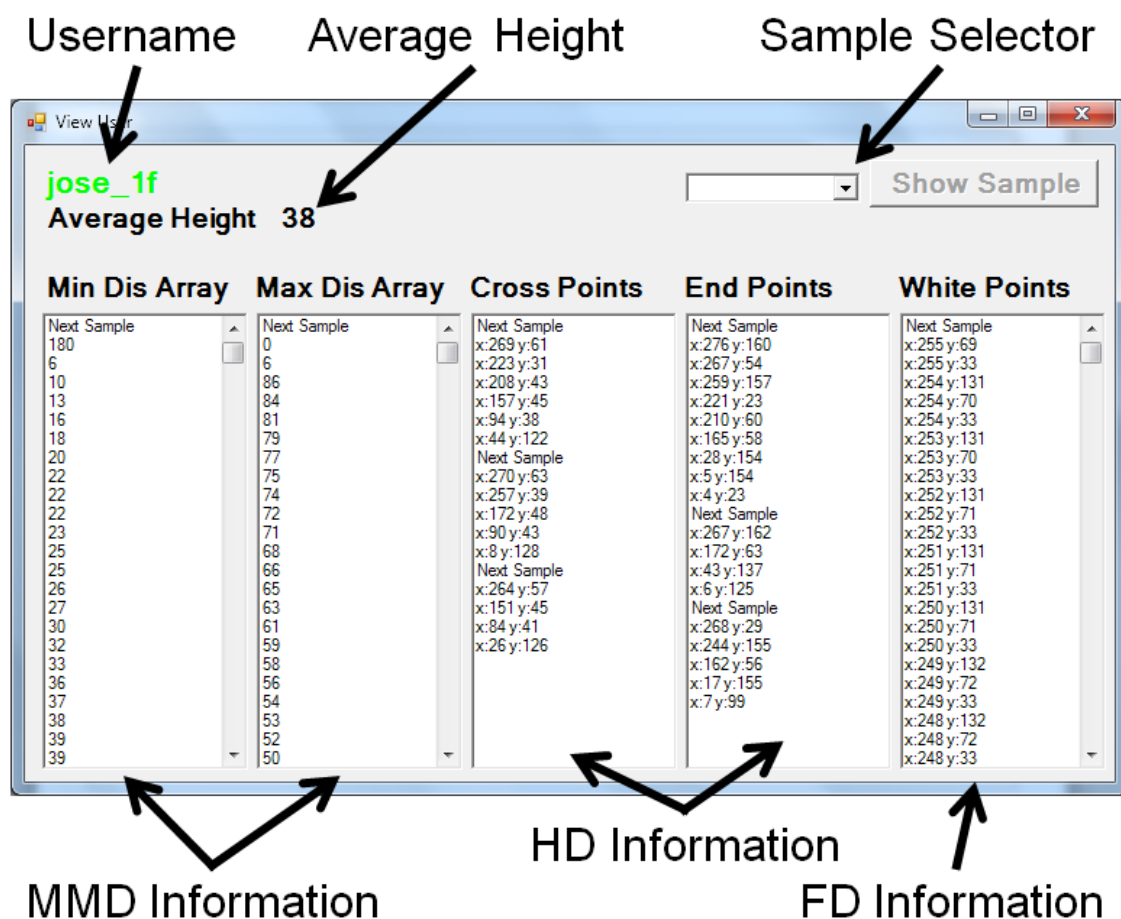


Fig 28: View User window.

Figure 29 shows an image of the "View Sample" window. This window shows only information related to a single sample of a given user, containing the same arrays of points found in the input finger vein pattern, but also contains an image of the vein pattern itself stored in the database and created from the White Points array.

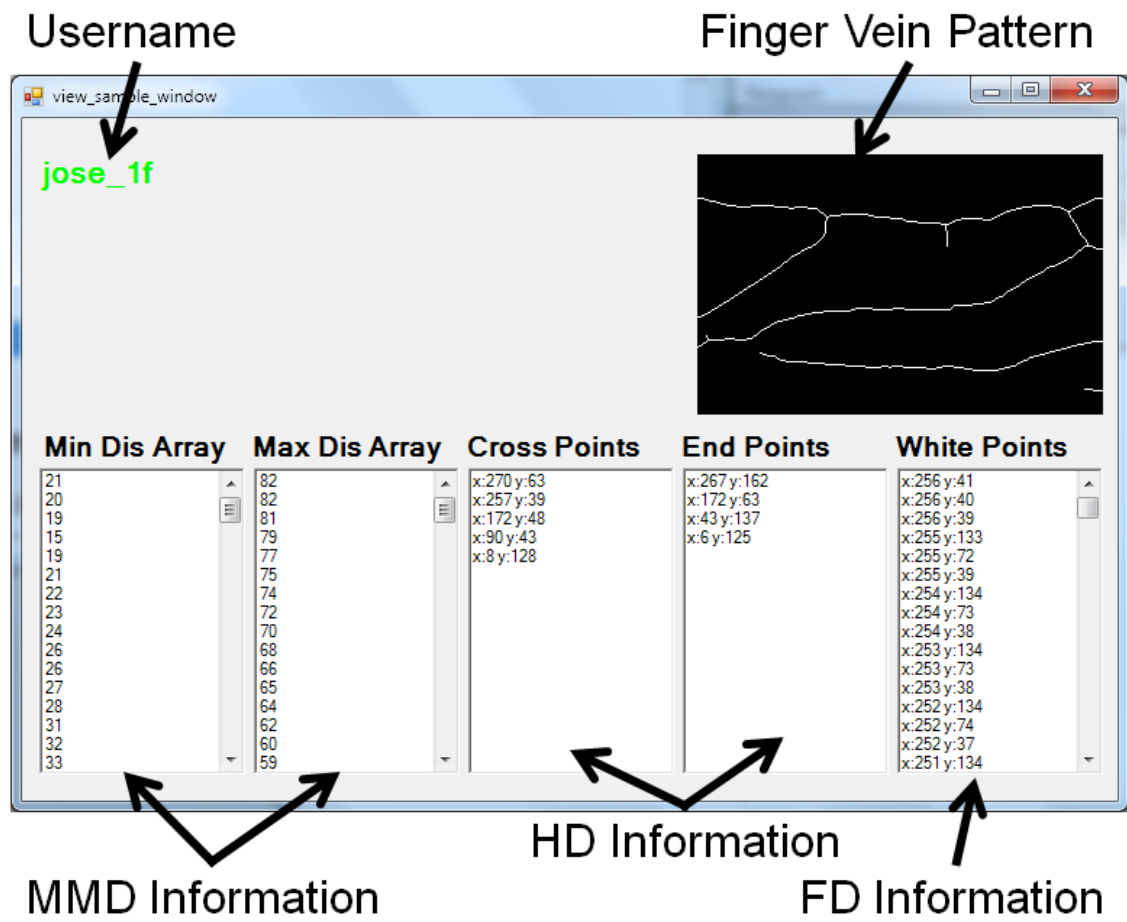


Figure 29: View Sample window.

6. Conclusions

This thesis study the possibilities and performance of different methods and setup of a finger vein identification system and propose a networked implementation of a finger vein identification system.

It has been explained the process to extract the finger vein pattern from an input Near Infrared camera and the required steps to make it something reliable for a finger vein identification system. This thesis has shown how it is possible to improve the problems related to a non-homogenous illumination of the finger from an array of illuminators. However, it would be recommendable for future device designs to avoid such design faults.

During the experimentation with the finger vein segmentation and extraction, as well as matching process, was possible to identify the effects related to rotations and different curvature levels of the finger. It would be recommendable for future designs to reduce the distance between the illuminators and the camera and consequently, reduce the height of the device, reducing the degrees of freedom of a given user at the time to introduce the finger.

It has studied and defined the properties of different methods and scoring setups in order to define the final parameters of our proposed solution.

It is also shown that methods with low levels of accuracy can be used to define a multimodal system with an increased accuracy but, at the same time, it has been proved that the minimum FAR required by the system can suffer a great penalty if one of the combined systems and the combination of more than one sample per user can greatly improve the performance of the system.

It has also been proposed a distributed setup using networked solution to provide the authentication information to different devices at remote locations.

7. References

- [1] Milan Sonka, Vaclav Hlavac, Roger Boyle: "Image Processing, Analysis and Machine Vision, Second Edition, Second Edition" ISBN 0-534-95393.
- [2] Jiang Hong, Guo Shuxu, Li Xueyan, Qian Xiaohua: "Vein Pattern Extraction Based on the Position-Gray-Profile Curve", 978-1-4244-4131-0/09/\$25.00 ©2009 IEEE
- [3] Byung Jung Kang, Kang Ryoung Park: "Multimodal biometric authentication based on the fusion of finger vein and finger geometry", Optical Engineering Letters 090501-1, September 2009, vol 48(9).
- [4] Septimiu Crisan, Ioan Gavril Tarnovan, Titus Eduard Crisan: "Radiation optimization and image processing algorithms in the identification of hand vein patterns", Computer Standarts & Interfaces 32, 2010, pp 130-140
- [5] Hao Luo, Fa-Xin Yu, Jeng-Shyang Pand, Shy-Chuan Chu, Pei-Wei Tsai: "A Survey of Vein Recognition Techniques", Information Technology Journal 9(6), 2010, ISSN 1842-5638, pp 1142-1149.
- [6] E.C. Lee, K.R. Park: "Restoration method of skin scattering blurred vein image for finger vein recognition", Electronics Letters, 8th October 2009 Vol. 45 No. 21.
- [7] Naoto MIURA, Akio NAGASAKA, Takafumi MIYATAKE: "Extraction of Finger-Vein Patterns Using Maximum Curvature Points in Image Profiles", IEICE TRANS. INF. & SYST., VOL.E90-D, NO.8 AUGUST 2007.
- [8] MEI Cong-li, XIAO Xiao, LIU Guo-hai, CHEN Yu, LI Qian-an: "Feature Extraction of Finger-vein Image Based on Morphologic Algorithm", 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, pp 407-411.
- [9] Junichi Hashimoto: "Finger Vein Authentication Technology and its Future", 2006 Symposium on VLSI Circuits Digest of Technical Papers.
- [10] Yi-Bo Zhang, Qin Li, Jane You, Prabir Bhattacharya: "Pam Vein Extraction and Matching for Personal Authentication", G. Qiu et al. (Eds): VISUAL 2007, LNCS 4781, pp. 154-167, 2007.
- [11] Naoto Miura, Akio Nagasaka, Takafumi Miyatake: "Extraction of Finger-Vein Patterns Using Maximum Curvature Points in Image Profiles", MVA2005 IAPR Conference on Machine Vision Applications, May 16-18, 2005 Tsukuba Science City, Japan.
- [12] Naoto Miura, Akio Nagasaki, Takafumi Miyatake: "Automatic Feature Extraction from non-uniform Finger Vein Image and its Application to Personal Identification", MVA2002 IAPR Workshop on Machine Vision

Applications, Dec. 11 - 13.2002, Nara- ken New Public Hall, Nara, Japan.

- [13] Toshiyuki Shimooka, Koichi Shimizu: "Artificial Immune System for Personal Identification with Finger Vein Pattern", M.Gh. Negoita et al. (Eds.): KES 2004, LNAI 3214, pp. 511–518, 2004.
- [14] N. Paveši, S. Ribari, D. Ribari: "Personal authentication using hand-geometry and palmprint Features – the stat of the art", Studies in Classification, Data Analysis, and Knowledge Organization, 2006, Part 16, 630-637, DOI: 10.1007/3-540-31314-1_77.
- [15] Anil K. Jain, Fellow, IEEE, Arun Ross, Member, IEEE, and Salil Prabhakar, Member, IEEE: "An Introduction to Biometric Recognition", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 14, NO. 1, JANUARY 2004.
- [16] Modris Greitans, Mihals Pudzs, Rihards Fuksis: "Palm Vein Biometrics Based on Infrared Imaging and Complex Matched Filtering", MM&Sec '10 Proceedings of the 12th ACM workshop on Multimedia and security ACM New York, NY, USA ©2010 table of contents ISBN: 978-1-4503-0286-9
- [17] Jian-Da Wu, Siou-Huan Ye: "Driver identification using finger-vein patterns with Radon transform and neural network", Expert Systems with Applications 36 (2009) 5793–5799.
- [18] KANG Wen-Xiong, DENG Fei-Qi: "Vein Image Segmentation Based on Distribution Ratio of Directional Fields", ACTA AUTOMATICA SINICA December, 2009 Vol. 35, No. 12.
- [19] Ajay Kumar, K. Venkata Prathyusha: "Personal Authentication using Hand Vein Triangulation and Knuckle Shape", IEEE Transactions on Image Processing, September, 2009.

