FULL HARDWARE SOLUTION FOR PROCESSING IRIS BIOMETRICS

Judith Liu-Jimenez [a], Raul Sanchez-Reillo[a], Carmen Sanchez-Avila [b]

[a] Dpto de Tecnologia Electronica
Grupo de Microelectronica
Universidad Carlos III de Madrid
c/ Butarque 15, 28911 Leganes (Madrid), Spain
Email: {jliu, rsreillo}@ing.uc3m.es

[b] Dpto. de Matematica Aplicada a las Tecnologias de la Informacion
E.T.S.I. Telecomunicacion
Universidad Politecnica de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
Email: csanchez@mat.upm.es

ABSTRACT

As security is becoming one of the most important concerns in Information society, the use of Biometrics is increasing everyday. Iris recognition is a biometric technique that presents low error rates with simpler matching algorithms than other biometrics techniques. Unfortunately feature extraction methods with iris biometrics are computational cost consuming.

When developing new identification tokens, computational cost and processing time should be reduced, to provide cheaper devices, which could allow a viable solution in a commercial system. The authors, in this paper, develop a different hardware solution to be applied to a biometric token, which provides lower processing time and higher security than commercial systems.

1. INTRODUCTION

Nowadays, we are becoming quite used to hear about security. Words as robbery, hacker or intruder are becoming quite well known. Security has become necessary to protect data, money or even buildings. Traditionally Security was focused just on the physical protection, which was provided mainly by automatic devices and sometimes, and when user authentication is required, having a security guard to perform such task. But this situation changes with Information Society, where the necessity of securing remote access to information emerges. Information Society allows the access of different users from all parts of the world to some services or data. Unfortunately this advantage brings also a weak point for the company information system, as those services and data can be accessible for not authorized individuals.

User's identity becomes the key to access to information. Security systems are now required to differ authorized users among those who are not, detecting potential intruders among authorized users. To perform such identification, there has been different techniques used. Those are mainly based on the use of something the authorized user knows, or something he/she possesses (e.g., passwords and/or keys or cards). But these techniques are not as secure as desired. Most of the users use as a password something related to his/her private life, such as the name of the pet or the birthday date of his/her couple. Therefore passwords are easily guessed. Also, the authorized user can loose easily the key or card that allows the access. Due to these lacks of security, still nowadays a human being is needed to perform the identification, such a policeman requiering us the passport to cross the borders to check the photograph of it.

Biometrics appears as a suitable solution for those cases when it is not possible to have a security agent to identify users. Biometrics pretends to perform the identification automatically, but using the same method the human being does it: i.e., using a physical or behavioral characteristic as personal features of the user [1]. This kind of identification avoids some of the problems mentioned above, as it´s almost impossible to lose the device to be matched or, even more, the difficulty in forging such features requiered. On the other hand, there is an important disadvantage: potential damage of

0-7803-9245-0/05/$20.00

the human body could reduce the number of characteristic suitable for perform the recognition.
In this paper the authors introduce a biometric recognition system, which uses as unique features of the user, his human iris. This biometric technique will be introduced in the next section. The following one will be focused on the weak points that any biometrics system could have to handle. These weak points are improved by the hardware solution that the authors propose in the fourth section. After describing its implementation, the results obtained will be shown, leaving last section for conclusions and future work on this area.

2. BIOMETRICS SYSTEM. IRIS RECOGNITON

From the different biometric techniques existing nowadays (fingerprint, face, hand geometry, voice, etc.), this paper uses iris biometrics. The human Iris possesses some characteristics which make it suitable to be used in biometric applications: 1) the possibility of finding an iris equal to another one is considered to be null, even the two iris of the same individual are different. 2) The iris pattern does not change through the user's whole life. 3) It is naturally isolated by the cornea. 4) Modifying it surgically without any risk for the vision is nearly impossible. 5) The physically response to light provides a suitable way to test the aliveness of it (avoiding the use of syntetic eyes).

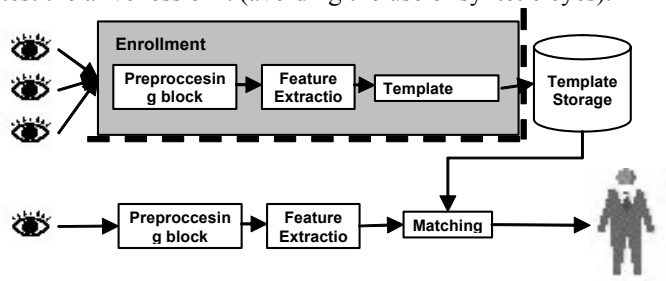

Figure 1: Biometric System scene

Figure 1 shows the typical scenario of biometric systems. Above is the enrolment phase, which consists on obtaining the template vector, which will represent the user in the system. Enrolment usually takes part in a secured environment. Remaining part of Figure 1, represents the verification phase (lower half of the figure). Verification can take place in different environments, and is in charge of performing the user identification each time he/she try to access the system.
In more detail, a biometric system can be divided in the following blocks

2.1. Image Adquisition

As first step in all biometrics systems, there should be a way to obtain the data from the outer world. In case of fingerprint can be a termograph sensor, in case of iris a camera is used. Cameras used for iris, work on the visible or infrared range, as the colour is not relevant and problems of dilatation of the pupil do not appear. They can be photo or video cameras, but it is interesting to obtain several captures of the iris each time, to assure aliveness and quality of the capture.

2.2. Preprocessing block

The preprocessing block prepares the image in order to reduce computational cost and functionalities of the following block. Some typical processes of this block are finding iris boundaries, equalizing the image, changing its size to normalize it and taking out the noise of low band frequency.

To find the boundaries different techniques are used, such those based on the maximum of the circular variation of the luminance or based on the Hough transform. [2] [3] [4]. This is a process that can be integrated in a dedicated camera for iris biometrics.

2.3. Feature Extraction block

The problem of performing a biometric identification can be reduced to a mathematical problem. The main aim is to reduce the image into a vector that represents univocally the user, so that obtain the unique features contained in the image. Once this set of features is extracted, then the following block will do the matching. The feature extraction block is in charge of performing the task of getting such a vector.

Different approaches can be used for the feature extraction. Most of the algorithms used for the feature extraction block are based on the Gabor Transform applied on sectors of the iris [2] [5]. Other methods are based on the used of wavelet transform: Iris signature denotes a data set which allows a suitable extraction of its features [6]. On the iris signature, obtained from an iris annular region, the discrete dyadic wavelet transform [7] in 4 different scales is applied. A digital feature vector is obtained by computation of its zero-crossing representation.

2.4. Matching

This block should be the most robust part of the algorithm. Here the comparison between the previous stored vector (user's template), and the feature vector obtained from the sample is performed. This can be faced as a pattern recognition problem. We can use metrics, based on mathematical distances, such Euclidean, Hamming or Zero-crossing. [6]

The use of one matching algorithm or another depends on the previously feature extraction block used

3. SECURITY PROBLEM OF A BIOMETRIC SYSTEM

Biometrics systems, even performing much better than previous identification systems, are not the panacea for the security problem. They also present some weak points, suitables of being attacked. In this section some of these critical points will be described.

One of the main problems these systems have is storage of personal data. The user template is high sensible information as it defines his/her identity for its whole life. Nowadays there are mainly two types of systems that face this problem: centralized and distributed systems. Centralized systems store data in a unique data base, which requires some preventing for a physical attack or a remote attack of such database. There is also the need of a continuous communication with the terminals that perform the rest of the processes involved. On the other hand, in a distributed system, the data is stored separatelly; each user has his/her personal data stored in a token, reducing the chances of a physical or remote attack.

Communication between terminal and token is also critical. Information transfer can be attacked by intruders or Trojans. Similar situation can take place in the matching algorithm, which should be performed in a high secured way, as an intruder can just force the result of the matching allowing his/her own entering.

Even if the previous comments are solved, the feature extraction block is also suitable of being attack. Possible intruder can just get a feature vector obtained and wait for the answer of the system. In case this is affirmative the intruder will now possess a true feature code vector for following attacks.

Considering all these matters, the authors propose a solution which avoids most of the situations mentioned above. This solution is based on performing the whole verification algorithm in hardware. Most of the problems previously mentioned, come from using software platforms, where the information is suitable of being observed. The solution consists on an architecture for a distributed system, where hardware performs most of this risky processes. This piece of hardware will be inside the user's identification token. Therefore, the token proposed will not only store the template, but also perform matching and feature extraction tasks. Two additional advantages are obtained: first security level increases; second, a reduction of the terminal performing is obtained, making it simpler and cheaper.

When designing a token, some requirements should be considered. It should be portable and its processing time should not increase, as the user should be able to carry the token, and the time for recognition should not be large.

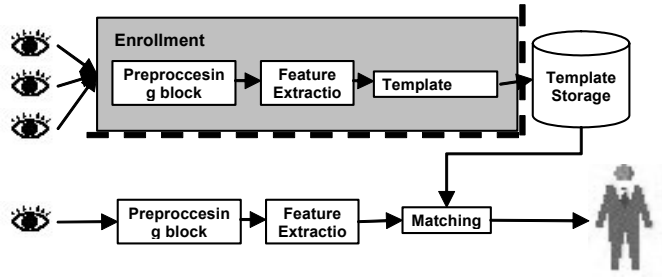

Figure 2: Solution proposed for a Biometric System, where critical processes take part in hardware, so security is higher.

4. BIOMETRIC SYSTEM IMPLEMENTATION

In order to make the development easier, the work is split in two parts: matching algorithm and feature extraction.

In the following subsections, authors will describe each part separately, showing implementation issues. The following section will cover the results obtained.

4.1. Matching

Authors have studied different verifiers: Euclidean distance, Zero-crossing distance and Hamming distance. The best results have been obtained using a Hamming distance algorithm for matching, described as follows:

$$d_H(y,p) = \frac{1}{L}\sum_{i=1}^{L} p_i \oplus y_i$$

Where L denotes the dimension of the vector, yi the ith component of the sample feature vector, pi the ith component of the template feature vector and exclusive-OR.

To implement this formula, three different hardware approaches have been studied: The first one consists in performing the XOR operation parallel for the input vector and after that performing the adding using just an adder. The second one is quite similar to the first one, but the adder is replaced by a shift register and a counter. The third implementation is based on a pipelined structure in order to maximize the use of the hardware area (Figure 3). [10]

Due to the vectors length obtained, an input protocol has been used to introduce data to the system, as it was not possible implementing a matching system on a FPGA where the whole vectors can be processed at the same time. So the verifiers proposed have been implemented following an input protocol.

Feature and Template vectors are split in words. The matching system would process the words and store the result obtained to take it into account for the next words. Using an input protocol, processing rate and area

occupied depend on the word length used in the protocol considered in each case.

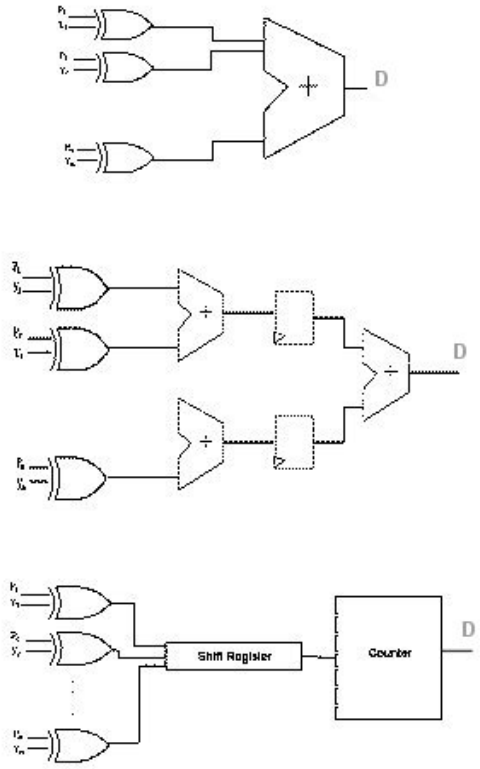

Figure 3: Matching approaches used

4.2. Feature Extraction Block

Feature extraction algorithm chosen is based on two requirements, performance of the whole system and complexity of the hardware required to achieve it.

The algorithm used for extracting the feature vector does not present the best results, but it is simple enough to be implemented in low semiconductor area. The algorithm proposed is based on Daugman's [2] algorithm, but it has been simplified: Gabor Filters have been used instead of Gabor transform. Using these filters the results obtained are not as good as Daugman's algorithm, but the method is much simpler. The main idea is obtaining the same information, but instead of perform the whole integration of the image, performing it by sectors [5]. The algorithm can be summarized as Figure 4 shows.

The preprocessed image is stored in a RAM memory after the transfer from the terminal. Using the same memory or another different, we translate the image from a Cartesian to Polar coordinates and store it. Only the two lateral parts of the iris are considered. To achieve the mentioned coordinate change, a CORDID algorithm is used. [8]

Afterwards, it is necessary to stretch the image, to enlarge the histogram and emphasize the contrast. And at last, weighting different sections of the image by Gabor filters, is performed. As filter coefficients are fixed, i.e. they do not depend on the image; this part has been implemented using fixed multipliers. After multiplying, these results are integrated for the later comparison with the threshold, to determine whether it is a '1' or a '0' bit.

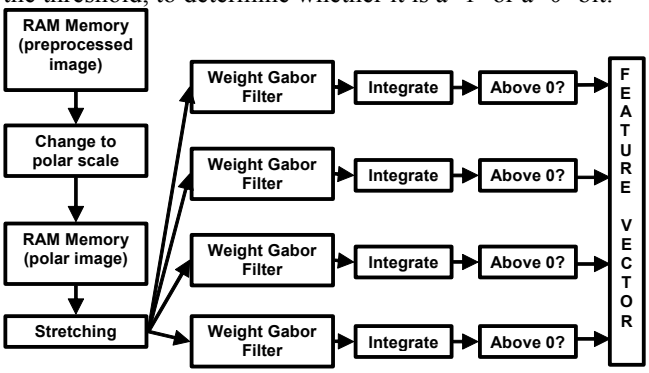

Figure 4: Feature Extraction Block

5. RESULTS OBTAINED

5.1. Matching

The FPGA used to develop a prototype is a Xilinx VIRTEX 2. This FPGA has 484 pins, which can be configured as input or output, so the maximum word length is 242. Results obtained for the three approaches used are shown below.

5.1.1.XOR-Adder

As the Figure 5 shows, the area needed for this implementation is mainly determined by the number of flip-flops (dffs) needed. This area increases with the number of bits of the input word as it was expected.

The CLK frequency is almost the same in all the different cases as the critical path is independent of the word length introduced. The number of cycles needed to compute a vector does not either depends on the number of bits. Taking in consideration that this process should repeat on the 1856 bits/word length, the time needed for the complete process of matching decreases with the word length introduced, the best time will be gotten with the largest vector. In all cases results are much better than using a commercial solution based on a potential SmartCard, where the typical frequency is 3.57 MHz.

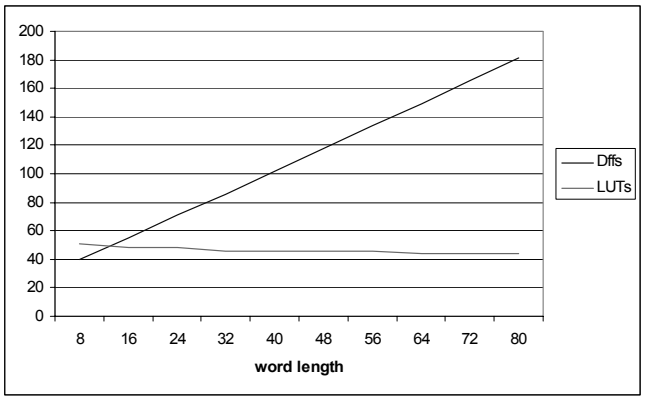

Figure 5: Area needed for a XOR-Adder implementation

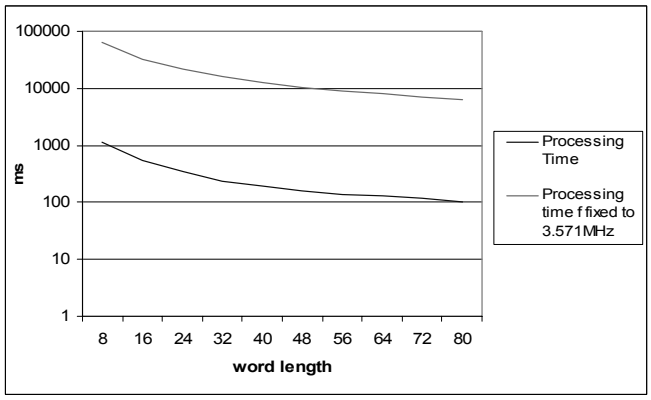

Figure 6: Comparison of Processing Time (ns) for a XOR-Adder Implementation (blue line) and a SmartCard (red line)

5.1.2. XOR- Shift Register-Counter

The function generators and registers needed to implement the addition are proportional to the number of bits of the input vectors.

In this case, the processing time of each of the divisions of the original vectors is dependant on the number of bits of each division. Also the time of the whole processing is depending on this number, because the number of repetition is inverse proportional to the bits introduced. Considering the maximum CLK frequency the curve obtained can be seen in the Figure 8.

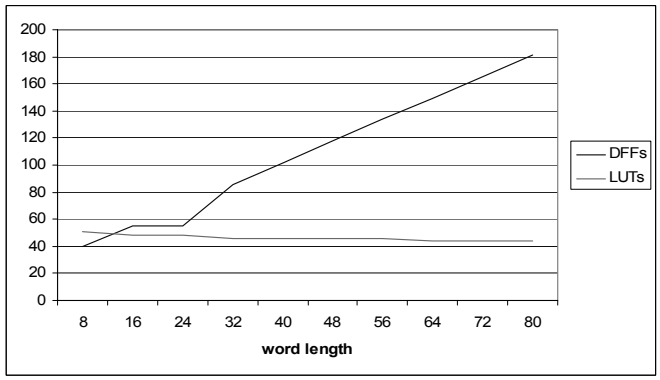

Figure 7: Area occupied by a XOR-Shift register-counter depending on the length of the input vectors

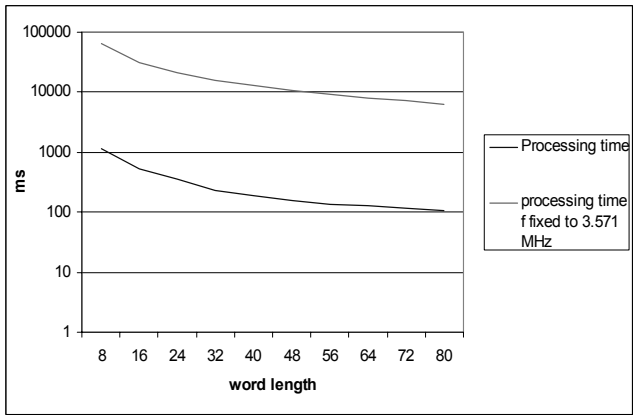

Figure 8: Complete processing time (ms) depending on the length of the input vector for a XOR-Shift Register-Counter implementation (blue line). In red the time expected for a commercial solution (SmartCard)

5.1.3. Pipelined Architecture

This implementation is more flexible than previous ones. Designer can change the segmentation level to be realized, which will affect the number of addends in the different adders or the length of the input vector.

Authors have implemented a segmentation in three levels, obtaining results considering different addends in each step, or varying the number of adder in each step.
If the length of the input vectors is 8, the area occupied is independent of the number of addends in each level. Area is determined by the number of look-up tables (LUTs), also called Function Generators, needed. Then area variations between different number of addends in the segmentation level, are not significant.

The time employed for the different possibilities studied in this pipelined solution is almost constant independent of the number of adders and the addends of

each one. In case of an 8 bit long vector, the time expected for a completed processing of the vector is 1.27 ms.

5.2. Feature Extraction Block

5.2.1. Cordic Results

CORDIC algorithm introduces some errors in the calculation of the address to be pointed due to the lack of floating point operations and truncate real numbers. These variations on the address are nor significant enough: even the address of the pixel is not the correct one, the error just causes a shift in the address, moving it to an address of a surroing pixel of the desired one. The value of the pixel found can be different for the desired, but as it is besides it, both values are quite similar.
To get better results the number of bits used has been varied, as larger vectors produces better results, but increases the hardware area. The solution chosen is using vectors of 16 bits.

5.2.1. Stretching

The stretching has been done to separate spectral components of the image, which usually are concentrated on one bandwidth. The result is an image quite similar to the previous one but with higher contrast.
In this operation, errors are committed again, the reason of these errors are the same than the ones mentioned above. In this case, errors are assumed to be impossible to correct. So they have to be considered and try to correct them in the following part of the algorithm.
Figure 9 shows the results obtained in these two steps, the row above shows the results obtained using hardware simulation, the row below are the images obtained using software. The results seem to be quite similar, but there are some differences not perceptible to the human eye, that shows the difference between hardware and software that will be more outstanding when obtaining the feature vector.

5.2.3. Gabor Filter Ponderation

Before weighting by the coefficients of Gabor filters, the image should be split into sectors, and on each sector is applied the Gabor filter ponderation. The size of the sector is chosen to be 20x20, so the image is split into 1680 sectors. Gabor filters should be the same size.

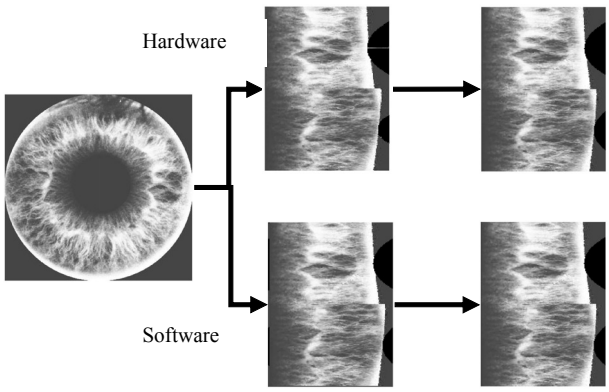

Figure 9: Difference between hardware ad software results

Gabor filters used are in four orientations; therefore, the number of multipliers is reduced to four. Gabor filters coefficients should be multiplied by a factor and rounded to integer numbers, introducing an error in all the operations with them. This factor is due to previous steps, as normalization and division have been postponed.
Considering this, the solutions proposed performs access to memory at the time the multiplier is applied to the previous pixel value and the integration of the value before is computed, optimizating the processing time.
Once the whole sector is computed, the value obtained should be compared to a threshold. This threshold has been fixed empirically, considering errors and factors introduced by previous steps of the algorithm.
Tests performed till nowadays with this system do not show same results to software. Using integer numbers and the lack of float point operations makes the feature code completed different than expected. However, these new feature vectors allow performing matching as they seem to be different among them. Performing the feature extraction and the matching using hardware instead of software, the false rejection rate and false acceptance rate are the same than in software, but the distance between samples is reduced.
For prototype development a Xilinx VIRTEX 2 FPGA has been used. In this case, area occupied is determined by flip-flops (dffs) and Function Generator (LUTs). The number of both of them is quite similar, 93 dffs and 92 LUTs.
The best improvement obtained with this new architecture is related to processing time, the software time for extracting the features is 0.701 seg, time for the process using a hardware solution is 4 ms, a reduction of the processing time of almost a 175% is obtained. This significant reduction allows performing several verifications a the same time a single verification is obtained with a software implementation. This leads to a situation quite comfortable for the user, as most of the systems require several verifications for providing an acceptance or rejection.

5.3. Full Hardware Solution

Performance and robustness of the whole system have been reduced, as the distance between samples is reduced.With the database used, the FAR and FFR rates are the same than in sofware [6], as there seems to show a enough separation (not as much as in software) among samples to maintain these rates.
On the other hand, time reduction is outstanding. The hardware system used pipelined architectures, reduces the computational time by more than a 200%, as the matching processes a word of the feature vector at the same time the feature extraction block is processing the following word, without waiting for the previous block to finish.

6. CONCLUSIONS

In this paper, authors present a new biometric architecture based on a combination of hardware and software, more secure and faster than commercial solutions.
Matching algorithm performance is the same than software solutions developed, not introducing errors. Processing time using this new approach has been reduced almost an 80%, introducing data by an input protocol of 8 bits word length and a pipelined structure.
Within the feature extraction block although, processing time is reduced but performance is worse than the software one. But even though, results of the whole system present the same performance level as the software solution, with a dramatic reduction of time. Nowadays authors are studying possible ways to improve this performance with constraints of hardware area and time need for the token.

7. ACKNOWLEDGMENTS

Work partially funded by the European Union, through the research project IST-2002-001766 BioSec (Biometrics and Security, http://www.biosec.org), in the IST (Information Society Technologies) priority of the 6th Framework Program

8. REFERENCES

[1] Jain A. K., Bolle R. Pankanti S., et al., *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publishers. 1999.

[2] J. Daugman, "High Confidence Visual Recognition by Test of Statistical Independence", *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol 15, pp. 1148-1161, 1993

[3] Ma L., Tan T., Wang Y. and Zhang, D. "Efficient Iris Recognition by characterizing Key Local Variations", *IEEE Trans. Image Processing*, vol 13, no.6, pp. 739-750.

[4] L. Ma, et al. "Iris recognition using circular symmetric filters". *International Conference on Pattern recognition*, vol.2, pp.414–417, 2002.

[5] R. Sanchez-Reillo, C. Sanchez-Avila," Processing of the Human Iris Pattern for Biometric Identification" *Proc. Of IPMU 2000 (8th International Conference on Information Processing and Management of Uncerinty in Knowledge Based Systems)*, pp. 653-656, Spain, 2000.

[6] C. Sanchez-Avila, R. Sanchez-Reillo, "Multiscale Analysis for Iris Biometrics". *Proc. Of 36th Annual 2002 International Carnahan Conference on Security Technology*, pp. 35-38, 2002.

[7] Q.M. Tieng, and W.W. Boles, "Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol 19 (8), pp 910-916, 1997.

[8] Javier Hormigo, Manuel Sanchez, Mario A. Gonzalez, Gerardo Bandera, Julio Villalba, "Optimized FPGA Implementation of Trigonometric Functions with Large Input Argument" Proc. Of *XIX Conference on Design of Circuits and Integrated Systems*, 2004

[9] Judith Liu-Jimenez, Raul Sanchez-Reillo, Carmen Sanchez-Avila, Luis Entrena "Iris Biometrics Verifiers for Low Cost Identification Tokens", Proc. Of *XIX Conference on Design of Circuits and Integrated Systems*. pp.38, 2004

[10] Judith Liu-Jimenez, Raul Sanchez-Reillo, Carmen Sanchez-Avila "Biometric Co-Processor for an Authentication System using Iris Biometrics " *Proc. Of 38th Annual 2004 International Carnahan Conference on Security Technology*.