Design and implementation of an algorithm for creating templates for the purpose of iris biometric authentication through the analysis of textures implemented on a FPGA

F. J. Giacometto, J. M. Vilardy, C. O. Torres and L. Mattos
Laboratorio de Optica e Informatica, Universidad Popular del Cesar, Sede balneario Hurtado, Valledupar, Cesar, Colombia
E-mail: franciscogiacometto@unicesar.edu.co

Abstract. Currently addressing problems related to security in access control, as a consequence, have been developed applications that work under unique characteristics in individuals, such as biometric features. In the world becomes important working with biometric images such as the liveliness of the iris which are for both the pattern of retinal images as your blood vessels. This paper presents an implementation of an algorithm for creating templates for biometric authentication with ocular features for FPGA, in which the object of study is that the texture pattern of iris is unique to each individual. The authentication will be based in processes such as edge extraction methods, segmentation principle of John Daugman and Libor Masek’s, and standardization to obtain necessary templates for the search of matches in a database and then get the expected results of authentication.

1. Introduction
Studies of the morphology of the iris began in 1987, when Flom and Safir [1] observed the morphology stability of the iris over human life and felt that the probability of the existence of two similar iris was 1 in $10^{72}$. The iris is widely recognized as one of the most reliable biometric features: it has a random morphogenesis and, apparently, has not genetic penetration.

In relation to the existing biometric authentication algorithms we can do a brief analysis of their characteristics by stages. The initial stage refers to the segmentation of the iris. This involves identifying boundary of the iris, inner (pupil) and outer (sclera). In 1993, Daugman [2] proposed an integrated differential operator to find the inner and outer boundaries of iris. Wildes [3] achieved by the segmentation of the iris through the gradient operator, followed by the identification of a binary matrix consisting of voting points necessary for identification of the circumference of the Hough circular transform.

In [4], the authors propose a method based on the method of Wilde, which, together with a clustering process, achieved robustness for non-cooperative environments. To compensate variations in the size of the pupil and the image distance, it is common to transfer the segmented iris regions in fixed length and dimensionless polar coordinate. This stage is usually done through the method proposed by Daugman [2].
In feature extraction, approaches for iris recognition can be divided into three main categories: methods based in transitory phase (for example, [3]), zero crossing methods (for example, [5]), and methods based on texture analysis (for example, [3]). Daugman [2] uses multiscale wavelets quadrature to extract the information and obtain an iris signature with 2048 binary components. Boles and Boashash [5] calculated the zero-crossing representation of 1D wavelet for different resolutions of concentric circles.

The application of these algorithms in the FPGA (Field Programmable Gate Array) logic-programmable devices can be found in the following references [6], [7] [8] [9]. They use databases in the development of biometric system of iris, as CASIA iris image database collected by the Institute of Automation, Chinese Academy of Sciences [10] and UBIRIS collected by the SOCIA Lab - Soft Computing and Image Analysis Group, Department of Computer Science, University of Beira Interior, Covilh Portugal.

In this article, the implementation of an algorithm to generate templates for biometric authentication using iris texture is presented. The implementation of the algorithm is developed on the Genesys development board, which has a LX50T FPGA from family Virtex -5 of Xilinx, which is distributed by the manufacturer Digilent [11]. This development board provides an efficient platform for a quick verification of designed algorithm.

FPGA technology provides a solution for authentication of low consumption, cost and high effectivity, based on technological considerations of the prototype, may be suggested that the compact size of the FPGA implementation of the authentication system using iris texture is suitable for portable devices.

Section II presents the method of iris segmentation implemented. Section III describes the hardware architecture implemented on the programmable device. Section IV describes the results obtained with the hardware implementation and finally Section V describes the conclusions of this article.

2. Iris Segmentation Process
In the following section, is analyzed the iris segmentation method. There are included the location of internal and external borders of the iris, the normalization of iris region and development of the circular region in a rectangular block of constant size. Fig. 1 shows the anatomy of the human eye and the area of the sclera and iris for references.

![Figure 1: Human eye anatomy.](image-url)
2.1. Scaling
When we begin to study the segmentation algorithm, we find the characteristic of efficiency in the location algorithm improve with the size of the image, but the time in which operations are performed increase proportionally because the voting matrix of Hough transform receives more marked pixel. For this reason it is necessary an image scaling, achieved through the bicubic interpolation process which takes 16 pixels closer to the reference pixel and calculates the new value of intensity to the output matrix. It was experimentally determined that the maximum scaling value that the image can be reduced to achieve minimum losses is 1/4 of the original image.

2.2. Filtering and Edge Extraction
For determinate the boundaries of the iris, this stage becomes vitally important. It is performed applying a low pass Gaussian filter of 13 * 13 positions to the original image in order to soften the image and try to eliminate any existing noise.

\[
G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}
\]  

Equation (1) describes the Gaussian filter used, where the parameter \( \sigma = 2 \).

Subsequently, the gradient operator is applied in the directions of the rows, columns, right and left diagonal, finding the regions of higher contrast and the direction of maximum variation of the image; achieving an edge matrix of the image and an orientation matrix with values between 0 and 180.

2.3. No Maximum Pixel Removal
Analyzing the image obtained from the filtering process, are observed many pixels that are not local maximums, which are not considered as vital information when they are used in the circular Hough transform, but they provide a large computational burden if they are processed by this algorithm, therefore must be filtered the image to process of these discarded components.

In this step the orientation mask obtained from the gradient is used in conjunction with the image gradient to determine if the pixel is a local maximum or not. This is accomplished by comparing the value of the analyzed pixel bilinear interpolation of their respective sides, left and right, if the pixel value exceeds the interpolation achieved is considered a local maximum.

2.4. Tresholding using histeresis
In this step, the resulting image of the process described above is filtered, removing irrelevant elements using double thresholding, this corresponds to the determination of the condition if the analyzed pixel is equal to or greater than a threshold, if so is marked as vertex for the second step of thresholding, in the second step the pixels marked as vertices are a reference to find its 8 neighboring pixels, these pixels are analyzed with a different threshold, if they exceed the requirement of being above this threshold are marked as vertices.

The pixels marked as vertices then form a binary matrix, which may be treated with circular Hough transform.

2.5. Circular Hough Transform
The circular Hough transform is a standard algorithm of computer vision that can be used to determine parameters of simple geometric objects such as lines or circles present in an image. The circular Hough transform in our case is used to derive the radius and the coordinated center of the pupil and the iris.
Because circular Hough transform is a transform in three dimensions, the previous stages of image processing with any reduction in voting points were required. This reduces greatly the computational load on the selected device.

At this stage the marked pixels \((x_j, y_j), j = 1, \ldots, n\) in the input matrix are processed by the algorithm, generating a set of concentric circles according to preset radios (2), these radios are determined by experimental analysis of the iris and pupil opening in the examined database.

\[
g(x_j, y_j, x_c, y_c, r) = (x_j - x_c)^2 + (y_j - y_c)^2 - r^2 \tag{2}
\]

The points which pass each circle created are marked with a weight of unit value, resulting in a new space for the voting when there is a change in the radio value.

\[
H(x_c, y_c, r) = \sum_{j=1}^{n} h(x_j, y_j, x_c, y_c, r) \tag{3}
\]

Where

\[
h(x_j, y_j, x_c, y_c, r) = \begin{cases} 1 & \text{if } g(x_j, y_j, x_c, y_c, r) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}
\]

At the end of this process a matrix of three dimensions must be processed to find the position that maximizes the amount \(H\) from (3), this is the most value and will be considered as the center of the circle sought by the previous stages explained, in our case the iris or pupil circumference. The ratio will be determined by the third variable of the three-dimensional matrix that point to the radio of the imaginary circles that presented the biggest intensity in the voting of a position.

2.6. Location of the external border

The steps explained above, are part of the segmentation algorithm developed to find the circles that describe the edges between the pupil-iris and iris-sclera, these are called internal and external border. Because the algorithm must be reused making slight variations to determine serially the external border and then the interior, has been broadly showed the technique and then further details.

Departure image for this process is the original image of size 280 * 320 pixels, this is scaled to only a quarter of its size 70 * 80 pixels. This image is filtered with a Gaussian lowpass filter, and then the gradient operator is applied only in vertical direction because the eyelashes and eyelids are oriented horizontally and these cover the top and bottom of the iris in most cases, thus leaving only the vertical walls of the iris easily identifiable, this also reduces interferences from edges marked for lids and lashes. The pixels removal and hysteresis thresholding are performed and circular Hough transform is applied to obtain the center and radius of the circle that describes the external border of the iris.
2.7. Location of the internal border

Departure image this process is the original image of size 280 * 320 pixels, this image is cutted leaving only the pixels that meet the condition of being within the perimeter defined by the radius and center defined above for location of the exterior border, this ensures that the eyelids and other elements in the image does not affect objectively the computational calculation of the segmentation algorithm.

This image is scaled to half its size, filtered using a Gaussian lowpass filter, and then the gradient operator is applied in vertical and horizontal direction. This is possible because the pupil is usually not covered by eyelids, eyelashes or any interference element in the detection of this circle. The removal of pixels, hysteresis thresholding and circular Hough transform are applied to obtain the center and radius of the circle that describes the internal border of the iris.

Figure 2: Images resulting from processes described in this stage (a) original image, (b) gradient the scaled image, (c) mask orientation, (d) removal of pixels no maximum, (e) thresholding by hysteresis.
2.8. Normalization

In this stage the base template for the process of correlation is generated from the original image and the parameters of the outer and inner border found in stages previously exposed. The process of creating the template is based on the development of the circular region that makes up the iris in a rectangular region [2] as shown in Figure 4.

The reallocation of the iris image \( I(x, y) \) Cartesian coordinates; the normalized non concentric polar representation can be modeled as:

\[
I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta)
\]  

(5)

Where

\[
x(r, \theta) = (1 - r)x_P(\theta) + rx_l(\theta)
\]  

(6)
\[ y(r, \theta) = (1 - r)y_p(\theta) + ry_l(\theta) \]  

(7)

Where \( I(x, y) \) is the region of the iris, \((x, y)\) are the original Cartesian coordinates \((r, \theta)\) are polar coordinates for standard and \(x_p, y_p\) and \(x_l, y_l\) are the coordinates of the borders of the pupil and iris along the direction \(\theta\).

This algorithm runs for 20 marked positions created imaginary concentric circles with radii greater than the radius found for the internal border and less than that found for the upper boundary, these circles are composed of 240 equally spaced points, taking the coordinates denoted by circles starting from the lowest to the highest radio, you can get the pixels to which he applied bicubic interpolation, this in order to obtain samples as representative as possible, the result of this interpolation vector is stored in rows of each radio, thus producing an array of 20 * 240, which is taken as the reference template.

![Images](image1.jpg)

Figure 5: Images resulting from the processes described in this stage, this can be seen the borders of the iris region, obtained by the segmentation process, the concentric circles marking the pixels to be molded into the rectangular region obtained in the last image.

3. Hardware architecture

In Figure 6 there is a block diagram of the hardware architecture for the creating algorithm of templates with embedded Matlab functions programmed. This architecture performs the process of segmentation and normalization to a matrix of 280 * 320.
All Matlab embedded blocks were concurrently and behavioral programmed and using numerical representation in fixed point. Below are described the functionality of the blocks of Figure 6: Escalamiento, Operador_Canny, Supresion_no_maximos, Umbralizacion_histeresis, Transformada_circular_Hough, Normalizacion, Segmentacion_imagen_ori, Control_plantilla.

In the introduction to this paper, we showed different databases which are normally tested through fit proves to the recognition algorithms of ocular patterns. In our case the methods described were tested with images taken from CASIA iris image database version 1.0. The CASIA database is made up of 756 iris images taken for both eyes of the user, each of these images is escalating levels of 8-bit gray and has a resolution of 320x280 [10].

At start of our observation of the block diagram of the template creation algorithm is appreciable that the original image acquired from the database is transformed into a column vector of 89 600 positions and then is stored in an external RAM (Random Access Memory) with fixed-point format of 22 bits; that in order to use that process repeatedly when it is demanded. You can also recognize a selector that points the vector that will be processed by the block Escalamiento in its original position. It let past the original vector of the image.

The block Escalamiento is responsible for apply the bicubic interpolation in the input vector according to 16 surrounding pixels, but because the processing is applied to a vector, we must reshape the indexes around of the pixel.

\[
\begin{bmatrix}
  n - r & n - 1 & n + r & n + 2r + 1 \\
  n - r & n - 1 & n + r & n + 2r + 1 \\
  n - r & n - 1 & n + r & n + 2r + 1 \\
  n - r & n - 1 & n + r & n + 2r + 1 \\
\end{bmatrix}
\]

In (8) there are described the indexes at which interpolation is applied, where r is equal to 280 and n is the index of the pixel that is subject to interpolation. The output of this block is a vector with 22 400 points, four times less than input if the s_switch selector is at its starting position.

The block Operador_canny is responsible to apply Gaussian filtering of 13*13 points and is also responsible for the application of the operator gradient, the latter depends in turn on the selector s_switch. In the original position of this selector, the gradient only will be applied in the vertical direction highlighting the external border to the iris; this in turn will produce a guidance vector that consists of zeros because this is the result of the tangent inverse horizontal gradient divided the vertical. In the second position of the selector it will be applied a gradient
in both directions, thus producing an orientation matrix with values between 0 and 180 degrees. For both processes, output vectors have the same size that the input vector.

The block Supresion no maximos is responsible for reducing of pixels in the input vector, placing null values that do not exceeding established conditions. The condition for the acceptance of valid pixels is based on bilinear interpolation of the same; if the pixel value exceeds the value of that interpolation by right and left, is considered a valid pixel. The output vector will have the same size of the input vector.

The block Umbralizacion histeresis is responsible for applying a second filtering on the input vector through the process described in section II of this article. This block changes its threshold values according to the position of a switch selector.

The block Transformada circular Hough is responsible for determining the parameters of the circumference that describe the external border. The output variable dato valido indicate that it has been found the wanted parameters. If it changes to ”1” means that the next time that blocks of this algorithm will be executed, it will be finding the parameters of the inner edge of the iris.

The block Normalizacion receives the parameters thrown by the block Transformada circular Hough; with these parameters performs the process described in Section II, after that the second data is received.

The block Segmentacion imagen ori receives the parameters thrown by the block Transformada circular Hough; with these data cuts the original image at the dimensions that describe the iris, according to the parameters of the external border. For perform this process is read the original image from RAM external.

The block Control plantilla is responsible for reading and writing of external RAM whenever dato valido output changes. This happens each time that are found the parameters of the iris border. Finally, we applied the tools available in the Simulink HDL (Hardware Description Language) coder to hardware architecture of templates generation algorithm in Figure 6 after being programmed, these tools are: compatibility code checker written in Simulink with respect to behavioral VHDL (VHSIC, Very High Speed Integrated Circuit, HDL, Hardware Description Language), implementations that are available to the encoder, generating of VHDL files from the codes programmed in Simulink and finally, the generation of files of the test stands that allow the simulation of VHDL codes generated in the ModelSim simulation tool (these last files have the same name that the VHDL files generated followed by identifier _tb).

4. Results
4.1. Experimental results
In this experiment, we use the database CASIA provided by the Institute of Automation Chinese Academy of Sciences [10].
In Figure 7 can be clearly seen that the location for internal and external borders is achieved when a large area of iris is discovered (Fig. 7 (a), (b), (e) (f)), because the circular Hough transform needs a geometric shape similar to a circle; if this is not achieved in the different steps of edge filtering and hysteresis thresholding, the result of the voting matrix with errors will be a bad location of the boundaries (Fig. 7 (c), (d), (g) (h)). In this example can be seen that the internal boundary of the iris was found successfully in all four cases, because this was never occlusal.

The data provided in (Fig. 7 (e) (f) (g) (h)) belongs to the radio found for the inner circle “P” and exterior circle “I”, measured in pixels. Note that both circles for each figure have their centers next in cases that can be clearly predict the iris edges (Fig. 7 (e) (f)).

4.2. Hardware details
The implementation of the hardware architecture of the creating templates algorithm for iris biometric authentication in Figure 6, was successfully synthesized in LX50T FPGA from Xilinx Virtex -5 family with the programming tool Xilinx Web Pack ISE 12.1 and a working frequency of 100 MHz.

The LX50T FPGA from Xilinx Virtex -5 family is in Genesys development board that is distributed by the manufacturer Digilent; this development board has an external RAM: 256 Mbyte DDR2 SODIMM, 100 MHz oscillator, Ports: HDMI, serial, PS / 2, expansion, USB, Ethernet 10/100/1000 PHY etc, [11].

Figure 7: Examples of the location of internal boundaries (pupil) and outer (sclera) of the iris for CASIA database [10].
Table 1: Comparison between the implementation hardware described in this article and other existing hardware implementations for the process of segmentation.

| Reviewed Article | Hardware | Algorithm | Time(ms) |
|------------------|----------|-----------|----------|
| Proposed Method  | FPGA(Virtex5, LX50T), CLK: 100MHz | Intensity gradient image and Hough transform circular. | 56.587 |
| [13]             | ADSP-BF561 EZ-KIT LITE, CLK: 600MHz | Intensity gradient image and Hough transform circular. | 683.16 |
| [14]             | DSP (TMS320DM642), CLK: 720MHz | - | 390.94 |
| [15]             | CPU, Pentium IV, CLK: 3.2-GHz | Elliptical model and modification of the Mumford-Shah model. | 1.82 |

Table 1 show a comparison of the implementation described in this article with other existing; the card used in this implementation works with a frequency of 100 MHz which gives a clock cycle time of 10 ns and time for calculating the segmentation process equal to 56,587 ms.

Figure 8: Genesys development board.
Figure 9: Resource Virtex-5 FPGA LX50T used by the hardware architecture implementing the algorithm of creating templates for iris biometric authentication is depicted in Figure 1.

Figure 9 show that there is not any over-mapping for any of the resources used in the FPGA, allowing the implementation of the hardware architecture of the algorithm in the FPGA device selected. Dual RAM Memory blocks (used 2 out of 48 available) and blocks dedicated to the addition, subtraction and multiplication (DSP48Es, employees 24 of 48 available). Logic utilization of Slices for programmed hardware architecture uses a 5% of slices available of the FPGA.

5. Conclusion
Hardware architecture of an algorithm for creating templates for the purpose of the iris biometric authentication through the analysis of textures was developed using Simulink HDL Coder™ and embedded Matlab. Finally we have the easier programming of: Registrations, dual RAM memories, finite state machines and handling of overflow and underflow, through the Fixed-Point Toolbox of MATLAB, Embedded MATLAB Function and Simulink HDL Coder, achieving generate architecture hardware described in this article in less than six weeks, according to experiences of the authors of this research article.

The hardware architecture synthesized and implemented in this article can be used in applications as authentication of biometric features that operate with measures of vitality, in this case, combining authentication methods iris textures and patterns obtained from the capillaries that can be observed with a thermal image of the eyeball.

References
[1] Flom L and Safir A 1987 Iris Recognition System US Patent 4 641 394
[2] Daugman J G 1993 High Confidence Visual Recognition of Persons by a Test of Statistical Independence Trans. IEEE Pattern Analysis and Machine Intelligence Vol. 25 no. 11 p. 1148-1161
[3] Wildes R P 1997 Iris Recognition: An Emerging Biometric Technology Proc. IEEE Vol. 85 no. 9 p. 1348-1363
[4] Proenca H and Alexandre L A 2006 Iris Segmentation Methodology for Non-cooperative Recognition Proc. IEEE Vision, Image and Signal Processing Vol. 153 no. 2 p. 199-205
[5] Boles W W and Boashash B 1998 A Human Identification Technique Using Images of the Iris and Wavelet Transform Trans. IEEE Signal Processing Vol. 46 no. 4 p. 1185-1188
[6] Liu J, Sanchez R, Lindoso A and Hurtado O 2006 FPGA Implementation for an Iris Biometric Processor IEEE Int. Conf. on Field Programmable Technology ISBN:0-7803-9729-0
[7] Mohd F, Asin Y, Tan L A and Rea M I 2004 The FPGA Prototyping of Iris Recognition For Biometric Identification Employing Neural Network 16th International Conference on Microelectronics ISBN: 0-7803-8656-6
[8] Rakvic R, Ulis B, Broussard R and Ives R, Steiner N 2009 Parallelizing Iris Recognition IEEE Trans. on Information Forensics and Security vol. 4 no. 4 ISBN: 1556-6013.
[9] Kannavarra R and Bourbakis N 2009 Iris Biometric Authentication based on Local Global Graphs: An FPGA Implementation IEEE Proc. Symp. on Computational Intelligence for Security and Defense Applications ISBN: 978-1-4244-3763-4.
[10] CASIA Iris Image Database Chinese Academy of Sciences Institute of Automation. Database of 756 Greyscale Eye Images, http://www.sinobiometrics.com Version 1.0, 2003.
[11] Genesys Virtex-5 FPGA Development Kit - Tutorial http://www.digilentinc.com/Data/Products/GENESYS/Genesys_fshbs_design.zip
[12] Masek L 2003 Recognition of Human Iris Patterns for Biometric Identification School of Computer Science and Software Engineering (University of Western Australia Press)
[13] Fatt R Y Ng, Tay Y H and Mok K M 2009 Iris Verification Algorithm Based on Texture Analysis and its Implementation on DSP Int. Conf. on Signal Acquisition and ProcessingDSP ISBN: 978-0-7695-3594-4
[14] Zhao X and Xie M 2009 A Practical Design of Iris Recognition System Based on DSP Int. Conf. on Intelligent Human-Machine Systems and Cybernetics ISBN: 978-0-7695-3752-8
[15] Vatsa M, Singh R and Noore A 2009 Improving Iris Recognition Performance Using Segmentation, Quality Enhancement, Match Score Fusion, and Indexing IEEE Trans. On Systems, Man, And CyberneticsPart B: Cybernetics Vol. 38 NO. 4 ISBN: 1083-4419