Template characterization and correlation algorithm created from segmentation for the iris biometric authentication based on 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. Among the most used biometric signals to set personal security permissions, take increasingly importance biometric iris recognition based on their textures and images of blood vessels due to the rich in these two unique characteristics that are unique to each individual. This paper presents an implementation of an algorithm characterization and correlation of templates created for biometric authentication based on iris texture analysis programmed on a FPGA (Field Programmable Gate Array), authentication is based on processes like characterization methods based on frequency analysis of the sample, and frequency correlation to obtain the expected results of authentication.

1. Introduction
Before proceeding to describe the process of biometric authentication using iris images, it should be mentioned that templates based on iris texture analysis were obtained in reference [1], using the procedure described by R. Wildes [2] and the algorithm designed by Libor Masek [3]. To test the system, a data set of eye images will be used as inputs; a database of 756 grayscale eye images courtesy of The Chinese Academy of Sciences Institute of Automation (CASIA) [21].

The iris biometric authentication is performed by means of the discrete correlation 2D (bi-dimensional) in the frequency domain through Discrete Fourier Transform in two dimensions (Discrete Fourier Transform, 2D DFT [4]), the 2D correlation operation is used to establish similarities between two images and is implemented efficiently by the fast algorithm for calculating the 2D DFT, which is known as the Fast Fourier Transform (Fast Fourier Transform, FFT [5]). The first FFT algorithms have been published by Cooley and Tukey in 1965 [6 - 7].

The hardware implementation of 2D DFT is that used in this research article is an extension of what was programmed in references [8, 9].

In this article is used the description language of hardware VHDL (VHSIC, Very High Speed Integrated Circuit, HDL, Hardware Description Language, [10]), to develop and implement discrete 2D correlation iris templates, VHDL files are directly generated from the Simulink programming environment, using embedded Matlab tools (Embedded Matlab) and Simulink HDL coder [11], these tools allow an easy management of complex signals, overflow, underflow.
etc. All implemented operations in these programming Simulink environments have fixed point numerical representation [12].

The results report about the FPGA programming of iris biometric authentication using the correlation in the frequency domain is organized as follows: Section II presents the 2D correlation operation and FFT. Section III is showed the hardware implementation of iris templates correlation using the 2D FFT. The analysis the obtained results with the raised hardware implementation in the Section III are described in Section IV. Finally, the conclusions are made in Section V.

2. 2D Discrete Correlation And FFT

2D discrete correlation between two images is defined as:

\[ C(n, m) = \sum_i \sum_j h(i, j)g(n+i, m+i) = h(n, m) \otimes g(n, m) \quad (1) \]

If \( H(k,s) \) and \( G(k,s) \) define the 2D DFT of \( h(n,m) \) and \( g(n,m) \), respectively, the correlation operation defined in equation (1) can be written in the domain frequency as:

\[ C(k, s) = H(k, s)G^*(k, s) \quad (2) \]

Applying the inverse 2D DFT to the previous equation:

\[ C(n, m) = \mathcal{F}^{-1}\{H(k, s)G^*(k, s)\} \quad (3) \]

Therefore to calculate the 2D discrete correlation, 2D DFT images are evaluated and both are multiplied by one of them conjugated, and finally to this result is applied a 2D inverse DFT.

2D DFT of an \( NxM \) image is defined as:

\[ X(k, s) = \mathcal{F}\{x(n, m)\} = \sum_{m=0}^{M-1} \left[ \sum_{n=0}^{N-1} x(n, m)W_N^{ns} \right] W_M^{mk} \quad (4) \]

\[ W_N = e^{-2\pi i/N} \quad (5) \]

And the inverse of \( X(k,s) \) is:

\[ X(k, s) = \frac{1}{MN} \sum_{m=0}^{M-1} \left[ \sum_{n=0}^{N-1} X(k, s)W_N^{-ns} \right] W_M^{-mk} = \mathcal{F}^{-1}\{X(k, s)\} \quad (6) \]

The definitions (4) and (6) shows that the 2D DFT can be divided into two 1D DFT (one-dimensional), ie, the 2D DFT can be calculated using first the FFT of the rows and then to this result is applied the FFT to the columns or otherwise, in the beginning the FFT is applied to the columns and then applies the FFT to the rows. With the previous result, 2D DFT is calculated having as a base the FFT.

As reference [8], in this article will be used the algorithm base-2 decimation in time to calculate FFT [5].

Figure 1 show clearly the procedures for carrying out the identification between two existing templates, there are the respective transform of each template, the correlation in frequency domain of the templates and the inverse transform on the resulting image from the correlation process, which indicates whit a intensity value if both templates belong to the same subject.
3. Hardware architecture

In Figure 2 there is a block diagram of the hardware architecture for 2D discrete correlation of iris templates with embedded Matlab functions programmed, this architecture calculates: the 2D FFT for the iris template of reference and the iris template to authenticate of RXS pixels ($R = 20$ rows and $S = 240$ columns), multiplication of the previous two 2D FFT (to one of them must be applied the complex conjugate), inverse 2D FFT (from 2D FFT algorithm) the outcome of the previous multiplication and finally the evaluation of the correlation peaks to determine the similarity between the iris template of reference and the iris template to authenticate.

All embedded Matlab blocks were programmed concurrently and using fixed point numerical representation, on below it is described the functionality of the blocks of Figure 2: $\text{FFT}_{64}$, $\text{FFT}_{512}$, multiplicacion$_{\text{conjugada}}$, control$_{\text{correlacion2D}}$ and evaluacion$_{\text{pico}}$$_{\text{correlacion}}$.

In the previous section was showed that FFT is the base algorithm for calculate the 2D DFT, in Figure 2 the blocks responsible for calculating the FFT are $\text{FFT}_{64}$ and $\text{FFT}_{512}$, these blocks calculate the FFT 64 and 512 complex points (if entry to transform is real, the imaginary part is taken as zero and it forms a complex signal), using base-2 decimation in time [9]. $\text{FFT}_{512}$ $\text{FFT}_{64}$ blocks work in streaming mode (receive data continuously) and process them when start signal of these same blocks are at high or one, after a delay ($D = 188$ clock cycles for $N = 64$ points or $D = 1288$ clock cycles for $N = 512$ points [8]) blocks $\text{FFT}_{512}$ $\text{FFT}_{64}$ and throw the results of the 1D DFT of $N$ points ($N = 64$ and $N = 512$) and the dvalid signal of each block is turned to one. The image transform with a 2D FFT has a size of RXS pixels ($R = 20$ rows and $S = 240$ columns, but is filled with zeros until $N = 64$ rows and $M = 512$ columns, in order to avoid aliasing effects at outcome of the correlation and obtain a power of 2 in the length of the 1D FFT, achieving a more efficient calculation of these FFT), and it is processed as a column vector of NM points ($64 \times 512 = 32768$ points), from previous information the $\text{FFT}_{64}$ block perform $M = 512$ 1D DFT of 64 points and $\text{FFT}_{512}$ block perform $N = 64$ 1D DFT of 512 points, in order to obtain a 2D DFT of an image of NxM pixels ($N = 64$ rows and $M = 512$ columns), this procedure is calculated for the iris template of reference and the iris template to authenticate. To calculate the inverse 2D DFT that precise the result of convolution of equation (3) there are used the same blocks $\text{FFT}_{512}$ and $\text{FFT}_{64}$, just before use and after them use, data input and output of these blocks should be applied the complex conjugate.

Equation (3) gives us the result of the correlation between two images using the 2D DFT, for this equation is presented as an intermediate step the multiplication 2D DFT of the iris template of reference and the template of the iris to authenticate (before applying the multiplication operation of these two 2D DFT it must be applied the complex conjugate), this multiplication and application of complex conjugate is carried out by the block $\text{multiplicacion$_{\text{conjugada}}$}$ if $\text{s$_{\text{habilitacion}}$}$ signal of this block is in high or one. Finally in block evaluacion$_{\text{pico}}$$_{\text{correlacion}}$ is evaluated the correlation peak intensity to determine the degree of coincidence between the...
Figure 2: Simulink Implementation of the hardware architecture of 2D discrete correlation iris templates for VHDL code generation with Simulink HDL Coder.

reference template of the iris and the iris template to authenticate. The correlation peak value is between zero and one, and when such value is close to one is established a logical one in the signal salida_comparacion in Figure 2, achieving a real authentication, otherwise you will obtain a logical zero determined like a fake authentication.
Iris templates of reference and iris template for the iris to authenticate are shown in the following figure:

![Iris templates](image)

Figure 3: (a) Reference iris template and (b) Template of the iris to authenticate.

The correlation of the image in Figure 3 (a) with itself and the correlation of the images in Figure 3 (a) and Figure 3 (b) obtained from proposed architecture are presented in the following figures:

![Autocorrelation](image)

Figure 4: Autocorrelation of the reference iris template: (a) 2D and (b) 3D view.
Figure 5: Correlation of the reference iris template and the template of the iris to authenticate: (a) 2D and (b) 3D view.

It should be annotated that a real authentication shows a correlation peak close to the value of one and this peak is located in the central part of the operation result of correlation between the two templates.

For the hardware architecture of 2D discrete correlation of iris templates presented in Figure 2 after being programmed, it is applied the tools available to the Simulink HDL coder, which are: compatibility checker of code written in Simulink with respect behavioral VHDL implementations available to the encoder, generation of VHDL files from the programmed Simulink codes and ultimately the generation of files in the test benches that allow the simulation of VHDL codes generated in the simulation tool ModelSim (these last files have the same name that the VHDL files generated followed by identifier _tb). The Simulink HDL coder separates automatically the real part from the imaginary part of complex numbers when there are generated the VHDL files into two different signals with respect to operations which join in such complex numbers. These operations are re-written with the two generated signals from each complex number.
The hardware architecture of proposal 2D discrete correlation, was synthesized from processes described in Section 2 and designed taking as optimization central parameter processing time in the FPGA device.

Using Matlab/Simulink we are able to rapidly simulate various partitions of the applications on Hardware and software. MATLAB/Simulink provides a hardware-software co-simulation environment where we can execute portions of the applications as software programs and other portions as customized hardware implementation. The hardware-software co-simulation can be further realized as a hardware-software co-implementation by generating the Hardware Description Language and bitstreams for FPGA implementation.

4. Results
The implementation of the hardware architecture of 2D discrete correlation of iris template shown in Figure 2 was successfully synthesized in the Virtex-5 FPGA LX50T of Xilinx with the programming Xilinx tool ISE Web Pack 12.1, and a working frequency of 100 MHz The Virtex-5 FPGA LX50T Xilinx is on the development board Genesys Virtex -5 FPGA Development Kit that is distributed by the manufacturer Digilent, this development board has an external memory RAM: DDR2 SDRAM SODIMM with 256Mbyte and word width of 64-bit, a 100 MHz oscillator, ports: HDMI, 10/100/1000 Ethernet PHY, 2 USB, serial, expansion, eight switches, eight LEDs and 3 Pushbuttons, etc. [13]. Figure 5 shows the development board Genesys.

Figure 7 shows the resource consumption of the FPGA for the implementation of the hardware architecture in Figure 2. Figure 7 show that there isnt any over-mapping for any of the resources used in the FPGA, allowing the implementation of the hardware architecture of 2D discrete of correlation iris templates in the chosen FPGA device. Memory blocks of dual RAMs (used 10 of the 48 available) and DSP48Es (employee 36 of the 48 available) are used exclusively for the blocks FFT_64, FFT_512 and multiplicacion_conjugada. Logic using of Slices for programmed hardware architecture uses 62% of the available FPGA slices, showing that the hardware implementation of 2D discrete correlation has a significant resource consumption due to the large number of operations that have to be performed.
![Figure 7: Resource Virtex-5 FPGA LX50T used by the hardware architecture of 2D discrete correlation iris templates depicted in Figure 1.](image)

**Table 1: Comparison between the implementation hardware described in this article and other existing hardware implementations for the process of identification.**

| Reviewed Article | Hardware | Algorithm                        | Accuracy(%) | Time(ms) |
|------------------|----------|----------------------------------|-------------|----------|
| Proposed Method  | FPGA(Virtex 5, LX50T) CLK: 100MHz | Frequency correlation using FFT | 96.52       | 16.11    |
| [14]             | ADSP-BF561 EZ-KIT LITE CLK: 600MHz | Hamming Distance                | 98.62       | 198.96   |
| [15]             | FPGA(Virtex xc5vlx30) CLK: 550 MHz | Euclidian Distance              | 92          | 0.0149   |
| [16]             | FPGA(Virtex 4, sx family) CLK: 153.53 MHz | Hamming Distance                | 88          | 2.725    |
| [17]             | FPGA(Stratix IV) CLK: 500 MHz | Hamming Distance                | -           | 0.002    |
| [17]             | FPGA(Cyclone-II EP2C35) CLK: 50 MHz | Hamming Distance                | -           | 0.02     |
| Reviewed Article | Hardware | Algorithm | Accuracy(%) | Time(ms) |
|------------------|----------|-----------|-------------|----------|
| [17]             | CPU, Intel Xeon X5355 | Hamming Distance | -            | 0.383    |
|                  | CLK: 2.66 GHz            |                        |             |          |
| [18]             | DSP (TMS320DM642)        | -                      | -           | 471.56   |
|                  | CLK: 720MHZ              |                        |             |          |
| [19]             | CPU, Pentium IV CLK: 3.2 GHz | Hamming Distance | 97.21       | 1.82     |
| [20]             | FPGA(Cyclone EP1C12Q240) | Hamming Distance | -          | 32 µs    |
|                  | CLK: 300MHz             |                        |             |          |

Hamming distance is defined in [14, 20] and Euclidian Distance in [22, 23].

The number of clock cycles (Nc) that takes the hardware architecture described in this paper to calculate the 2D discrete correlation between the two iris templates (reference image and image to authenticate) of NxM pixels (N = 64 rows and M = 512 columns), is given by the clock cycles consumed by the three 2D DFT over how long is the operation executed by the block multiplicacion_conjugada of Figure 2: the clock cycles of the 2D DFT of an image of 64x512 pixels of this article are dominated by the clock cycles of the 1D FFT of 512 points (block FFT_512 in Figure 2, since it is the major length of points and the blocks FFT_512 and FFT_64 in Figure 2 were programmed concurrently), and in this way with N = 512 and D = 1288 are NC1 = 526 351 clock cycles to calculate the 2D DFT an image of 64x512 pixels [9], with this result the total number of clock cycles it takes for the hardware architecture of Figure 2 to calculate the 2D discrete correlation of iris template NxM pixels (N = 64 rows and M = 512 columns) is:

\[
N_c = 3Ncl + NM = 3(526351) + (64)(512) = 1611821 \text{ clockrate}
\]  \hspace{1cm} (7)

With a working frequency of 100 MHz yields a clock cycle time of 10 ns and therefore the time for calculating the 2D discrete correlation iris template of 64x512 pixels is 16.11821 ms.

To validate the performance of the proposed correlation algorithm, we generate with [1], 876 templates that represent the number of templates corresponding to the images of the irises contained in the database CASSIA 1.0 [10], for which the process correlation shows a 96.52% accuracy and a false rejection rate of 3.48%.

Table 1 shows different types of implementations for the identification process of the template obtained from the process of segmentation and coding of the iris.

5. Conclusions
Hardware architecture for calculating the 2D discrete correlation of iris template using the 2D FFT with numerical representation of fixed point was developed using VHDL code generation using Simulink HDL CoderTM and embedded Matlab. The hardware implementation shown in this article uses the 1D FFT algorithm to calculate the 2D FFT and likewise use the same hardware architecture of the 2D FFT to calculate the inverse 2D FFT, we can see that the results were optimal in comparison to other existing architectures in specialized developing cards for DSP operations.

The hardware architecture of 2D discrete correlation of iris template synthesized and implemented in this work can also be used in DSP applications that involve the convolution operation using the 2D FFT.
References

[1] Giacometto F J, Vilardy J, Torres C and Mattos L 2010 Design and implementation of an algorithm for creating templates for the purpose of iris biometric authentication through the analysis of textures implemented on an FPGA Proc. abstract book of the VII Ibero-American Conference on Optics y X Latin-American Meeting on Optics, Lasers and Applications (Lima) ed J B Vazquez p 257

[2] Wildes R P 1997 Iris Recognition: An Emerging Biometric Technology Proc. IEEE Vol 85 no. 9 p 1348-1363

[3] Masek L 2003 Recognition of Human Iris Patterns for Biometric Identification School of Computer Science and Software Engineering (University of Western Australia Press)

[4] Jahne B 2003 (Digital Image Processing) Ed. Springer-Verlag, (Alemania)

[5] Hayes M 2001 (Digital Signal Processing) Ed. McGraw Hill (Estados Unidos) p 262-286

[6] Cooley J W and Tukey J W 1965 An algorithm for machine calculation of complex Fourier series Math. Computation vol. 19 p 297-301

[7] Cooley J W, Tukey J. W. and Welch P. D. 1969 The Finite Fourier Transform IEEE Transactions on Audio and Electroacoustics vol. AU-17 no. 2

[8] Giacometto F J, Vilardy J, Torres C and Mattos L 2010 Design and implementation in VHDL code of the two-dimensional fast Fourier transform for frequency filtering, convolution and correlation operations Proc. abstract book of the VII Ibero-American Conference on Optics y X Latin-American Meeting on Optics, Lasers and Applications (Lima) ed J B Vazquez p 189-190

[9] Yaruro Y and Vilardy J 2009 Descripción y Síntesis de Arquitecturas Hardware de La Transformada Rápida de Fourier en VHDL para Sistemas de Comunicaciones Basados en La Técnica de Multiplexación por División de Frecuencias Ortogonales (OFDM Estándar IEEE 802.11a) School of electronic Engineering (Universidad Popular del Cesar Press)

[10] Moraleda U 2001 Diseño de Hardware Digital con VHDL Ed Madrid p. 23-87

[11] Simulink HDL Coder 1 - User Guide http://www.mathworks.com/access/helpdesk/help/pdf_doc/sldhlcoder/sldhlcoder Ug.pdf

[12] Fixed-Point Toolbox 2 - User Guide http://www.mathworks.com/access/helpdesk/help/pdf_doc/fixedpoint/FPTUG.pdf

[13] Genesys Virtex-5 FPGA Development Kit - Tutorial http://www.digilentinc.com/Data/Products /GENESYS/Genesys_bsb_design.zip

[14] 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

[15] Kannavara 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.

[16] Liu J, Sanchez R .Lindo R and Hurtado O 2006 FPGA Implementation for an Iris Biometric Processor IEEE Int. Conf. on Field Programmable Technology ISBN:0-7803-9729-0

[17] 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 ISCN: 1556-6013.

[18] 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

[19] 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 ISCN: 1083-4419

[20] Hu Z and Xie M 2010Iris Biometric Processor Enhanced Module. FPGA-based Design Proc. Second International Conference on Computer Modeling and Simulation 259-62

[21] 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.

[22] Leeser M, Theiler J, Estlick M and Szymanski J Design Tradeoffs in A Hardware Implementation Of The K-Means Clustering Algorithm Ed. IEEE ISBN: 0-7803-6339-6-00

[23] Kota S, Raheja J, Gupta A, Rathi A and Sharma S 2009Principal Component Analysis for Gesture Recognition using SystemC proc. International Conference on Advances in Recent Technologies in Communication and Computing Ed. IEEE ISBN: 978-0-7695-3845-7/09