International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)

A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.

Research Paper

Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 7, Issue 11
November 2019.
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

Received on: 02.10.2019 Published on: 28.10.2019 Pages : 15 - 24

PROTECTED BIOMETRICS AUTHENTICATION METHOD THROUGH FPGA TECHNOQUE

C. Senthil Singh [a*], K. Manivannan [b]

Department of Electronics and Communication Engineering, Dr. M G R Educational and Research Institute, Chennai-95, Tamilnadu, India

[a] e-mail: senthilsingh@gmail.com, [b] e-mail: manivannank79@gmail.com

Abstract

The objective of this research work is to process the iris biometrics for Iris recognition which is used for secured authentication system using Field Programmable Gate Array (FPGA). Iris recognition system is the most reliable system for an individual identification. However it is a complex process and computationally very expensive. Iris patterns possess a high degree of randomness and uniqueness. Our approach is based on the iris segmentation of the eye, using a mechanism called edge detection with different edge detection algorithms such as Sobel, Prewitt and Canny edge detection algorithms. In this work, the iris image is preprocessed using MATLAB Simulink block set to handle the edge detection operation where in the edge detection block finds the edges in an input image by approximating the gradient magnitude of the image. Further it is post processed to get normalized output image. The processing of image in real time is time consuming hence we have adopted hardware level implementation using FPGA which offers parallelism and significantly reducing the processing time. The different edge detection algorithms mentioned above are implemented in Spartan3E FPGA kit. The proposed design implements the architecture of different edge detection algorithms through XSG (Xilinx System Generator Tool). The generated net lists are synthesized using Xilinx ISE 10.2i development suite. The FPGA implementation is done via hardware software co-simulation from XSG, use limited resources at enhanced maximum operating frequency with lower processing time and higher security than commercial systems.

Keywords: FPGA, Iris, Edge Detection, Thresholding, System Generator, Spartan3E

1. Introduction

In this technology dependant world an absolute requisite is needed to identify users of facilities and services which have importance not only to determine who accesses a system and/or service but also to determine types of services that should be provided to each user. To achieve this identification, biometric is an emerging technology that provides a higher level of secured access, as well as being convenient and comfortable for the user. Biometrics is currently being employed in many different scenarios, where one of the most common applications is Iris biometrics recognition techniques. This paper discusses in detail about the iris recognition using XSG and processing the Iris biometrics for the secured authentication system using FPGA, which involves image processing techniques. Our proposed technique alters the existing image using suitable edge detection algorithm to bring out the features in a desired manner. Any image consists of two dimensional arrays of pixel values.

The edge detection is an essential operation in image processing, useful in the areas of feature detection and feature extraction. The edges are the boundaries of shapes in an image which is defined as a single pixel with local discontinuity in its intensity. Edge detection is a process of identifying the immediate change in the image intensity using specific algorithm. Various factors are responsible for these local discontinuities in images, which are not limited to light, shadows and illumination. These discontinuities produced four different types of edge formation which are the step, ramp, roof and ridge edges. Edge detection highlights either a sharp or gradual discontinuity in the pixel intensity. Thus amount of data to be processed is significantly reduced and irrelevant information is being filtered out. There are several types of operators available for edge detection based on first and second order derivatives. In First order derivative input image is convolved by an adapted mask to generate a gradient image. The most significantly used operators like Prewitt and Sobel are the first order derivative operators also called gradient operators which spot edges by looking for maximum and minimum intensity values. The second order derivative include Canny operator, which is applied after smoothening the image to reduce noise. The Image processing task using FPGA can be accomplished with integrated use of tool such as MATLAB, Simulink and XSG. Simulink is a model based design environment which is integrated with MATLAB.

One of the block library provided by Simulink is Xilinx system generator (XSG) where, the System Generator token along with Xilinx has to be mapped to MATLAB. This heaps all Xilinx block set to the MATLAB, Simulink environment which can be directly used for building algorithms. Our work focuses on iris recognition techniques using edge detection with different operators and thresholding to edge detected image using Xilinx Block set. These algorithms are simulated in Xilinx system generator tool and corresponding output image is viewed through video processing block set within XSG library tool box. After obtaining the results, System Generator is configured for appropriate FPGA board. The FPGA board used here is Spartan XC3S500E-FG320 board. After compilation, programming file in Verilog has been created and can be accessed using Xilinx ISE10.2i IDE (Integrated Development Tool) environment there by obtaining corresponding HDL net list by synthesizing the model with appropriate simulation time. The obtained code is checked for syntax check, synthesized to view the RTL Schematic of the design and then implemented on FPGA. The Xilinx System Generator can generate User constraints file (UCF) for testing the architecture that we have implemented.

Bit stream generation is necessary to program the FPGA. Xilinx System Generator is a DSP design tool setup from Xilinx that enables the use of the Math works model-based Simulink design environment which makes it very easy to handle with respect to other software for FPGA design. This features leads to the use of Xilinx System Generator (XSG) in our work. It focuses in processing pixel to pixel of an image and in the modification of pixel neighborhood's results to the transformation of the whole or partial image. The proposed hardware implementation method is architecturally based on the Xilinx system generator tool integrated with ISE 10.2 development suite and MATLAB 2007a. The design focuses on achieving overall high performance, short development time and low cost.

2. Related Work

The image enhancement techniques use retinal fund us image of human eye, which is carried out on intensity based methods such as brightness control, contrast stretching and histogram equalization. The design is processed using Simulink and implementation was done using Spartan3E development board [1]. The hardware solution for processing iris biometrics in which Gabor Transform is applied on iris to gain feature extraction such as iris signature from its angular region, also by zero crossing representation. The matching of iris is performed by having pattern recognition problem with use of metrics called mathematical distance [2]. The edge detection with non-uniform assembly co-efficient for grey level image is discussed, and equal merge weights by means of multi scalar structure elements for RGB image. Here the edges are done using different operator like Sobel, Canny and log operators [3]. The method based on wild togetherness with clustering process achieves robustness for non cooperative environments. The base template for the process of correlation is generated from the original image and the parameters used are iris segmentation, filtering and edge extraction, thresholding. The hysteresis and Hough transform is used to localize the center circle of iris [4].

The enhancement of brain image using XSG is proposed here, were the images are fed to artificial neural network for normality or abnormality classification of brain. The method used is discrete wavelet transform to extract energy values from enhanced image using filters which offers appropriate hardware implementation [5]. The implementation of image segmentation framework based color hue channel histogram of HSV color space is proposed with design of classical edge detection algorithms such as Sobel, Prewitt, Robert and implemented using Spartan3E kit [6, 7]. The comparison of each edge detection operators is made by checking peak signal to noise ratio (PSNR) and mean squared error (MSE) for resultant edge detected image. This indicates the capability of producing accuracy in edge detection [8]. The Canny edge detection algorithm is implemented on Virtex5 Field Programmable Gate Array (FPGA) board. Using Xilinx platform studio and Xilinx ISE output image displayed on Video Graphics Array (VGA) monitor which is interfaced with board by using DVI connector [9]. The image recognition techniques is designed using system Generator and the approach used here for detection is the Hough transform which is a very efficient approach for location of parametric curves in an image, especially lines. This concept is successfully implemented in the device Virtex 5 ML507 (XC5VSXT platform) [10]. The Sobel edge detector [11] uses a pair of 3X3 convolution masks, estimating gradient in the x- direction and other gradient in y-direction for transferring 2-D pixel array into statistically uncorrelated data for removal of redundant data. Also demonstrates how to build a Sobel detector function of 5 ×5 dimension in MATLAB to find edges.

3. Proposed Design

The proposed edge detection algorithms comprises several transformations such as,

- Image pre processing block set
- Edge detection algorithms using XSG
- Image post processing block set

The different edge detector operator used to detect the iris image namely Sobel, Prewitt, Canny operators are implemented using FPGA. The goal of edge detection is to produce a line drawing of a scene from an image and to extract important features from the edges of an image like corners, lines, curves. Those features can be used by higher level of computer vision algorithms e.g., recognition etc. The image pre processing and image post processing are common operations for all image processing applications which are designed using Simulink block set. Figure 1 shows the actual implementation of different edge detection algorithms using XSG.

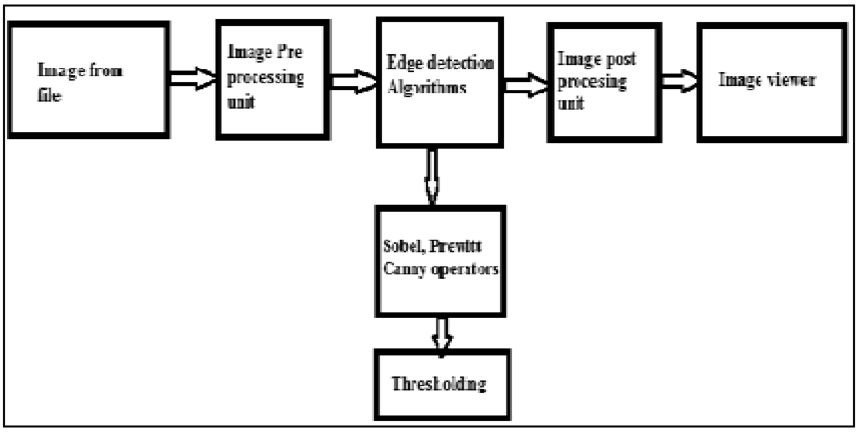
Image from file
Image Pre processing unit
Edge detection Algorithms
Image post procesing unit
Image viewer
Sobel, Prewitt Canny operators
Thresholding

Figure 1 Design Flow of Edge Detection Implementation in XSG

The Hardware implementation of edge detection algorithms are performed using the Spartan3E development board. All the necessary file compilation are done in XSG(Xilinx system generator).

Image Pre Processing Unit: Image pre-processing in MATLAB helps in providing input to FPGA. It provides the necessary test vector array which is applicable for FPGA bit stream compilation using system generator. Input images which could be color or grayscale are provided as input to the File block. A color space conversion block converts RGB to grayscale image and this data which is in 2D is to be converted to 1D for further processing. Frame conversion block sets output signal to frame based data and provides it to un-buffer block which converts this frame to scalar samples output at a higher sampling rate.

Image Post Processing Unit: The image post-processing blocks are used to convert the image output back to floating point type. It uses a Buffer block which converts scalar samples to frame output at lower sampling rate, and produces an output with different frame size by redistributing the data in each column of the input. Further it undergoes 1D to 2D (matrix) format signal block and Sub matrix block extracts a contiguous sub matrix from the M-by-N matrix. Finally a sink is used to display the output image back in the monitor, utilizing the Simulink block sets.

4. Methodology

Edge Detection: Edge detection is a fundamental operation used in the image processing and computer vision applications. The main objective of edge detection is to locate and identify sharp discontinuities from the given image. These discontinuities are due to abrupt change in pixel intensity which characterizes boundaries of objects and also to find boundaries between different regions in the image. Such boundaries are used to identify objects for segmentation and matching purposes. Edge detection technology not only detects the image gray value but also to determine their exact location.

Sobel Operator: It works by calculating the gradient of image intensity at each pixel within the image. Also it finds the direction of largest increase in intensity from light to dark and the rate of change in that direction. The result shows the changes in image at its each pixel, and therefore the pixel representation of an edge. The Sobel operator calculates the approximate image gradient of each pixel by convolving the image with a pair of 3x3 filters. These filters estimate the gradients in the horizontal(x) and vertical (y) directions and the magnitude of the gradient is simply the sum of these two gradients.

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figure 2 Convolution Mask of Sobel Edge Detector

Prewitt Operator: The Prewitt is a differentiation operator it computes an approximation of the gradient of the image intensity function. The Prewitt operator is based on convolving the image with a small, separable and integer value d-filter in horizontal and vertical directions It is relatively inexpensive in terms of computations like other operators. It detects the image using edge model by taking the maximum value of the model operator that is most similar to the detected region as the output of the operator. Both Prewitt operator and Sobel operator uses the same differential and filtering operations.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & -1 & 1 \\ 0 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 3 Convolution Mask of Prewitt Edge Detector

The function of Prewitt edge detection is almost same as of Sobel edge detection operator but Prewitt has different kernels. The Prewitt edge detection operators include a pair of 3x3 convolution kernel for eight directions. The XSG model block diagram of Sobel and Prewitt edge detection algorithms is shown by,

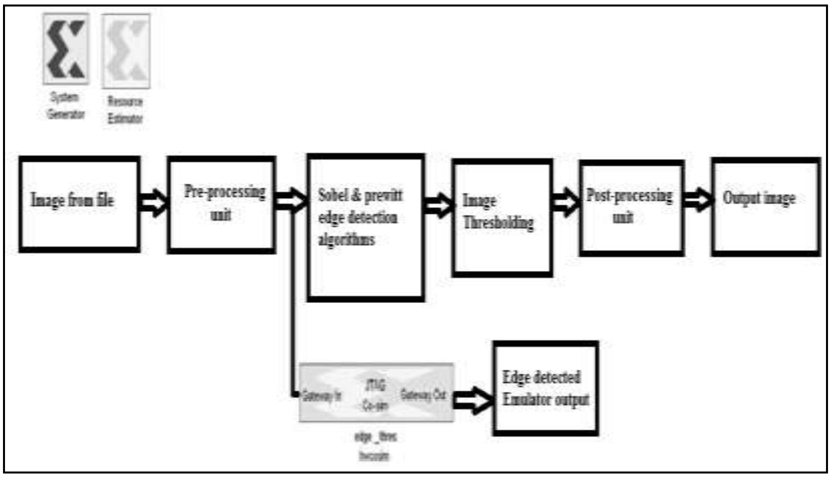

Figure 4 XSG Model Design of Edge Detection

Canny Operator: The Canny edge detector is an edge recognition operator that uses a multi-stage algorithm to identify a wide range of edges in images. It is mainly focused to find out the optimal edge. An optimal edge detector means good detection, this algorithm marks the real edges in the image as possible. Also it provides good localization which means marked edges should be as close as possible to the edge in the real image. To implement canny edge detector algorithm there are some processing steps. The initial step is to remove noise elements present in an image before applying edge detection. Image enhancement is basically improving the interpretability or perception of information in images for human viewers and providing better input for other automated image processing techniques. And image filtering is done using model based design with different filtering architecture which can be defined and Xilinx block can be created. The XSG model block diagram of canny edge detection algorithm is shown by,

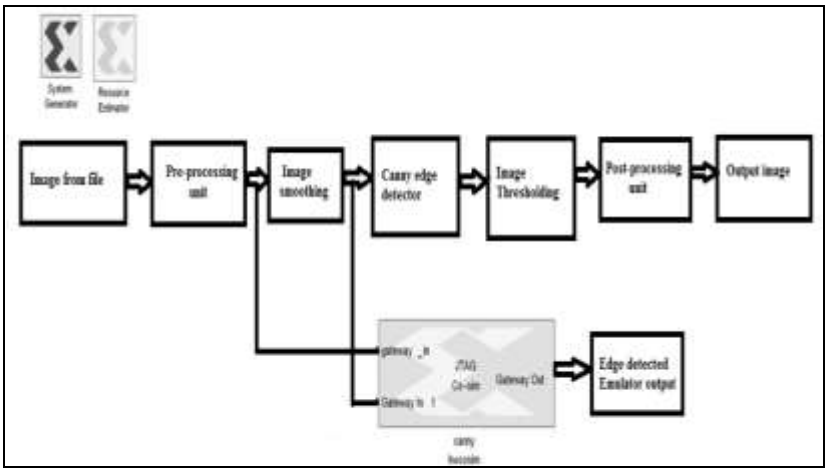

Figure 5 XSG Model Design of Canny Edge Detection

5. Hardware Software Co-Simulation and Results

The Hardware board is installed using XSG (Xilinx System Generator) Board Description Builder to implement the design in Spartan3E development kit. The clock is given with a frequency of 100 MHz, and set to single step clock for the hardware in lock step with software simulation. The automatic code is created by selecting generate button in system generator block dialog box. The code generator produces a FPGA configuration bit stream file for the design that is suitable for hardware co-simulation and also comprises of additional interfacing logic to communicate with the design using a physical interface between the FPGA platform and the PC. The completion of compiling design creates a new JTAG Co-simulation block. This block consists of all Xilinx blocks within the gateway in and out blocks of the original system. The Gateway in and Gateway out blocks is used to convert the Simulink input type such as integer, single, double format to Xilinx data type. The Co-simulation process displays the output signal produced by the FPGA hardware. The synthesis tool used to synthesize the design is XST tool. In this model, verilog is set for hardware description language. The image shown here is of size 250X250 which is given as input to the design for Edge detection and thresholding operations, whose obtained results using an efficient algorithms like Sobel, Prewitt are shown below and the corresponding FPGA/Hardware output are also listed here.

Figure 6 shows the Sobel edge detection outputs, and the Figure 7 shows the Prewitt edge detected output using XSG. Also the input image given to canny algorithms is of size 250x250 pixel iris image, whose optimized edge detected output is obtained for both XSG model based design and hardware software co-simulation using Spartan3E development board, Figure 8 shows the corresponding results. The generated HDL files are synthesized using Xilinx. The synthesis of the code will give the information about the errors, warnings, RTL schematic of that code, device utilization summary etc. Synthesis is a process by which an abstract form of designed circuit behavior or register transfer level (RTL) has been converted into a design implementation which is in terms of logic gates. The synthesis of verilog code for Sobel & Prewitt algorithms has been carried out by Xilinx Synthesis Technology (XST) tool, which is a part of Xilinx ISE 10.2i software.

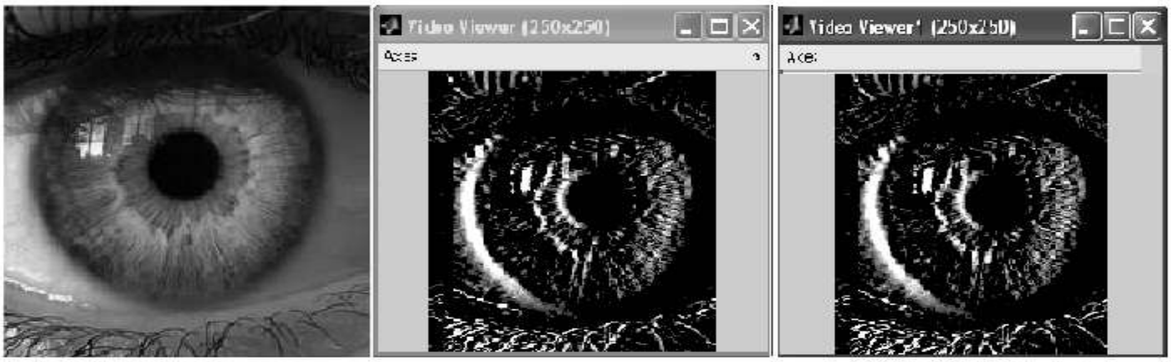

Figure 6a) Original image b) Simulink software output c) Emulator FPGA output

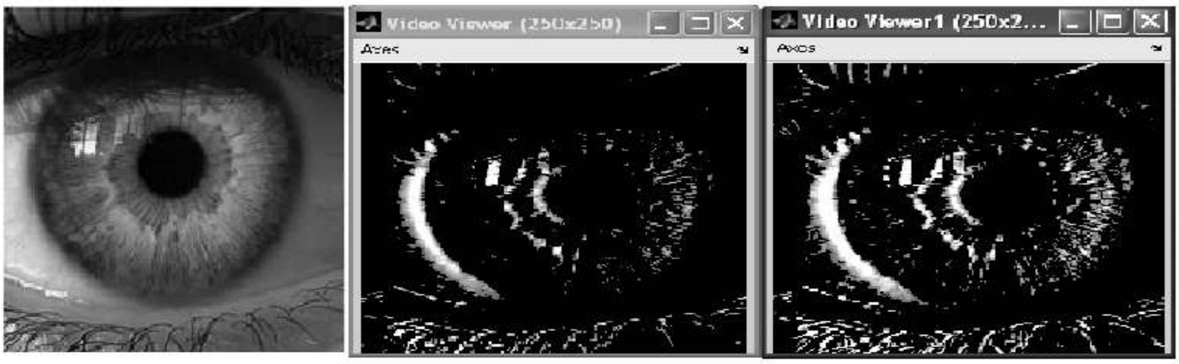

Figure 7 a) Original image b) Simulink software output c) Emulator FPGA output

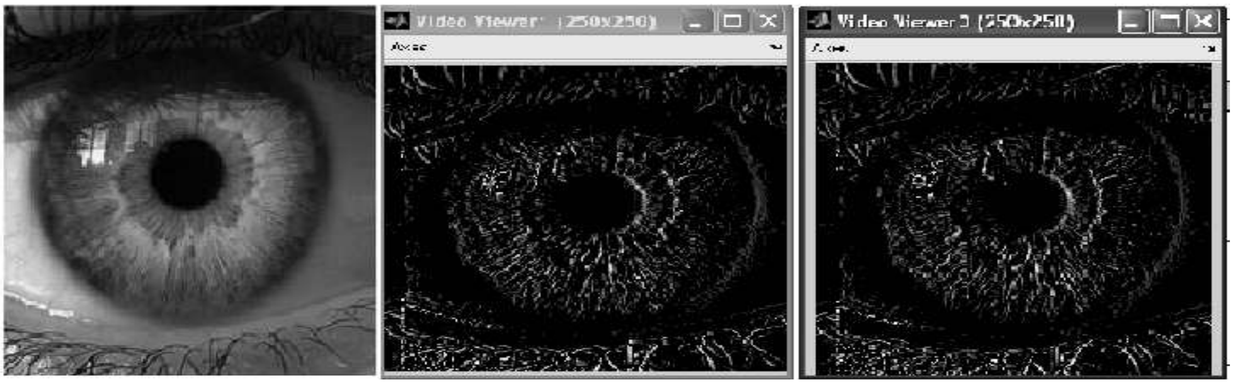

Figure 8a) Original image b)Simulink software output c) Emulator FPGA output

The Top-level RTL schematic of the proposed canny architecture is as shown in Figure 10, this is a schematic representation of the pre optimized design shown at the Register Transfer Level (RTL). The RTL Schematic representation of this design is in terms of generic symbols, such as adders, multipliers, AND gates, OR gates and it is generated after the HDL synthesis phase. The system blocks are designed for the Spartan3E starter board. The synthesized design also says about the utilization of target device for the proposed algorithms as a design summary report.

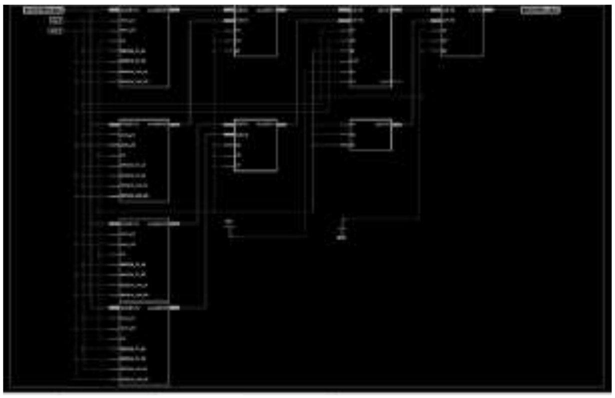

Figure 9 RTL Schematic of Sobel & Prewitt Edge Detection Algorithm

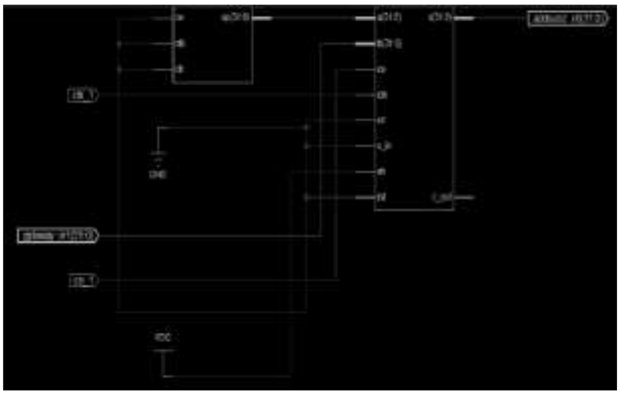

Figure 10 RTL Schematic of Canny Edge Detector

Device Utilization Summary: Device utilization summary shows the information related to device utilization analysis. The generated report shows the detailed report of the design being processed. The Slice Logic Distribution, and Synthesis results estimates how the design will be packed and placed into the target architecture, also the map provides details about utilization after packing and placement have been occurred. The device utilization summary obtained for Sobel, Prewitt, and canny architecture is given as, with the comparison results of above discussed edge detection algorithms, the canny edge detector operator possess a greater results which contains 76% reduction in area and 78% increase in its speed of operation than the Sobel and Prewitt operators.

Table 1 Device Utilization Summary

Parameters	Sobel Operator	Prewitt Operator	Canny Operator
Slices	75	75	12
LUT	132	132	32
IOB	66	68	64
Combinational path delay	9.595 ns	9.595 ns	5.363 ns
Maximum Operating Frequency	104.22 MHz	104.22 MHz	186.46 MHz

Evaluation Parameter: The performance evaluation of edge detections algorithms are accomplished by detection of true edges, processing time, error ratio, and noise level etc. In noise level, two metric standards are universally followed, Mean Squared Error (MSE), and Peak Signal to Noise Ratio (PSNR). The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are used in Image Compression. But, here it is used to compare image edge detection quality. Comparison of edge detection operators with common image is made. The operator which gives resultant image with less PSNR and high MSE is the one with high edge detection capability.

Mean Squared Error: The MSE incorporates degradation function and statistical characteristics of noise in the edge detected image. It measures the average squared difference between the estimator and the parameter. The MSE in terms of image edge detection could be higher by ensuring more edge points on the image and also it is capable of detecting weak edge points. This indicates a greater difference between the original and processed image.

$$\mathrm{MSE} = \frac{\sum \mathrm{M,N}[\mathrm{I}_1(\mathrm{m,n}) - \mathrm{I}_2(\mathrm{m,n})]^2}{\mathrm{M,N}}$$

Peak Signal to Noise Ratio: The PSNR is usually expressed in terms of the decibel (dB) scale. PSNR is a rough estimation to human perception of reconstruction quality. In edge detection PSNR should lesser to achieve proper results. The PSNR calculated based on the MSE by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{R}^2}{\mathrm{MSE}}\right)$$

The calculated MSE and PSNR for Sobel, Prewitt and canny edge detection algorithms are given below,

Table 2 Results of MSE &PSNR

Edge detector operators used	MSE	PSNR
Sobel	39.52	32.20
Prewitt	37.87	32.38
Canny	42.48	31.79

The value of MSE for Canny is estimated to have 10% greater than Prewitt operator and 6% more than the Sobel operator. And from the obtained PSNR value, Canny is 2% lesser than Prewitt operator and 0.5% lesser than the Sobel operator.

6. Conclusion

Sobel, Prewitt and Canny edge detection algorithms are designed and implemented on Spartan3E FPGA kit using Xilinx System Generator for Iris recognition. It is simpler to generate a stream of bit files rather than writing thousands of code lines for implementing the image processing techniques on FPGA and hence we have adopted XSG for our design. From the obtained implementation results, the performance of different edge detector algorithms have been discussed and concluded that canny has 76% of less area utilization than Sobel and Prewitt operators, also 78% increase in its Maximum operating frequency. The Canny architecture is composed of optimized design and it is less prone to noise. This made canny operator to be better than the other existing algorithms, since it has best MSE and PSNR values which is used to estimate the performance of algorithms in terms of producing accurate edges to determine object boundaries.

7. References

[1]. Galbally, J., Fierrez, J., Alonso-Fernandez, F. and Martinez-Diaz, M. Evaluation of direct attacks to fingerprint verification systems, Journal of Telecommunication Systems, 2010, 47(3-4), 243-254.

[2]. Baldisserra, D., Franco, A., Maio, D. and Maltoni, D., Fake fingerprint detection by odor analysis, International Conference on Biometric Authentication, 2005, 3832, 265-272.

[3]. Poh, N. et al. Benchmarking quality-dependent and cost-sensitive score-level multimodal biometric fusion algorithms, IEEE Transactions on Information Forensics and Security, 2009, 4(4), 849-866.

[4]. Nagar, A., Nandakumar, K. and Jain, A. K. A hybrid biometric cryptosystem for securing fingerprint minutiae templates, Pattern Recognition Letters, 2010, 31, 733-741.

[5]. Ross, A., Jain, A. and Reisman, J. A hybrid fingerprint matcher, Pattern Recognition, 2010, 36(7), 1661-1673.

[6]. Jain, A. K., Feng, J. and Nandakumar, K., Fingerprint matching, IEEE Computer, 2010, 201, 36-44.

[7]. Pan, S.B., Moon, D., Gil, Y., Ahn, D. and Chung, Y. An ultra-low memory fingerprint matching algorithm and its implementation on a 32-bit smart card, IEEE Transactions on Consumer Electronics, 2003, 49(2), 453-459.

[8]. Mostafa Abd Allah, M., A fast and memory efficient approach for fingerprint authentication system, IEEE Conference on Advanced Video and Signal Based Surveillance, 2005, 259-263.

[9]. Gupta, P., Ravi, S., Raghunathan, A. and Jha, N. K., Efficient fingerprint-based user authentication for embedded systems, Design Automation Conference, 2005, 244-247.

[10]. Lindoso, A., Entrena, L., López-Ongil, C. and Liu, J., Correlation-based fingerprint matching using FPGAs, IEEE International Conference on Field-Programmable Technology, 2005, 87-94.

[11]. Schaumont, P., Hwang, D. and Verbauwhede, I., Platform-based design for an embedded fingerprint authentication device, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2005, 24(12), 1929-1936.

[12]. Lindoso, A. and Entrena, L. High performance FPGA based image correlation, Journal of Real-time Image Processing, 2007, 2(4), 223-233.

[13]. Jiang, R. M. and Crookes, D. FPGA-based minutia matching for biometric fingerprint image database retrieval, Journal of Real-Time Image Processing, 2008, 3(3), 177-182

[14]. Danese, G., Giachero, M., Leporati, F., Matrone, G. and Nazzicari, N. An FPGA-based embedded system for fingerprint matching using phase-only correlation algorithm, IEEE Euromicro Conference on Digital System Design: Architectures, Methods and Tools, 2009, 672-679.

[15]. Razak, A. H. A. and Taharim, R. H., Implementing Gabor filter for fingerprint recognition using Verilog HDL, International Colloquium on Signal Processing & Its Applications, 2009, 423-427.