International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 1 – JANUARY 2018.

Implementation of Iris Recognition System using FPGA

Uma B[#1] and Pawan Kumar B[*2]

[#]Dept. of Telecommunication Engineering, RVCE, Bengaluru, INDIA

[*] Asst.Prof., Dept. of Telecommunication Engineering, RVCE, Bengaluru, INDIA

***Abstract*— Iris recognition is a standout amongst the most exact and high certainty for user verification technique that used today. The general purpose iris recognition system are of low speed and not portable. Hence there is a need to make use of dedicated hardware for this. Most of the algorithms of this system using MATLAB do not incorporate parallel processing instead comes under sequential processing. In the proposed framework, image pre-processing, segmentation of iris and normalization carried out using MATLAB R2016a software. Feature extraction using Ridge Energy Direction (RED) algorithm and feature matching using Hamming Distance(HD) is simulated using Xilinx ISE(14.7) design suit and synthesized on virtex-5 FPGA(XC5VLX110T) to achieve parallel processing and to improve the time delay. This methodology resulted reduction in number of device utilization on board and simulation time delay of RED algorithm is 4.796 µs and synthesized time delay of HD is 5.22ns. Hence achieved parallel processing using verilog modelling.**

***Index Terms*— Canny Edge Detection (CED); Circular Hough Transform (CHT); RED algorithm; HD; Field-Programmable Gate Array (FPGA) Virtex-5.**

I. INTRODUCTION

The passwords which are utilized for identification proof, to get access to some secured resources, can be forgotten, or guessed. The biometrics give a contrasting option to these strategies. Popular biometric identification proofs like fingerprints which are used generally can be forged, the human face usually changes over a timeframe. Iris identification is particular for any two persons and even for twins. Hence iris identification is popularly used now a days for security purposes. The unique pattern of iris will remain unchanged for most of human's lifetime, the acquisition of iris images cannot be intrusive, hence Iris recognition is a standout amongst the most exact and high certainty for user verification technique.

The automated approach of iris recognition in identification of biometrics uses computative pattern identification technique on acquired images of irises from person's eyes and the patterns inside irises are of complex structures, stable and unique. The iris properties that improve its appropriateness for use in high certainty identification frameworks include: (i) Its protection and inherent isolation from the outside environment. (ii) Its physiological reaction to light, which gives several characteristic tests against artifice and (iii) The iris features are impossible to modify surgically without unacceptable hazard to vision [4].

The eye images taken in VW band show poor clarity of pattern present in iris where as eye imaging in nearby infrared(NIR) band show up extravagantly composed, iris pattern. Because of this reason, for all iris affirmation frameworks which are publicly deployed, the acquisition of eye images has done in the presence of NIR wavelength band i.e. in the electromagnetic spectrum NIR range of 700-900nm. Due to specialized conditions of imaging which utilizes NIR spectrum, features of patterns in range of iris are greatly visible and there is better contrast among sclera, iris, pupil portions. Figure 1 shows the structure of iris and figure 2 shows the eye image in NIR band.

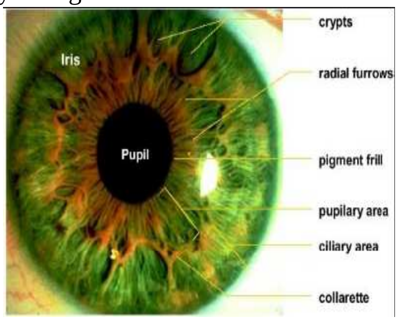

Figure 1: Structure of the Iris

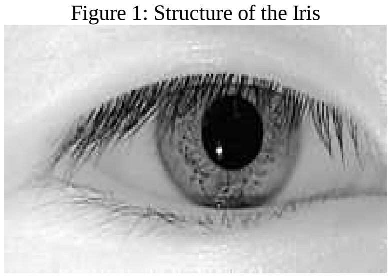

Figure 2: Eye image in NIR band

Most of the algorithms developed for iris recognition are on the basis of time consuming sequential processes executing on central processing units (CPUs). Therefore utilization of multi-processor stages is progressively embraced, in other words FPGA could give good performance of computation, combinational, complex and configurable functions. As a reason of this FPGAs can be utilized as a part of iris recognition to enhance iris recognition process execution by parallel computing.

II. PROPOSED IRIS RECOGNITION FRAMEWORK

In the proposed framework, CASIA dataset has been used for iris recognition, here the original eye image was pre-sampled to (320×240) pixels to crop the unneeded parts of the eye image and to decrease the time of processing during the pupil and iris boundaries detection[6]. These eye images are in the .bmp file format and each eye image of the file size 225KB. The proposed framework of iris recognition is as presented in figure 3. It consists of four major steps, they are segmentation of iris, process of normalization, extraction of features and finally matching of templates[2].

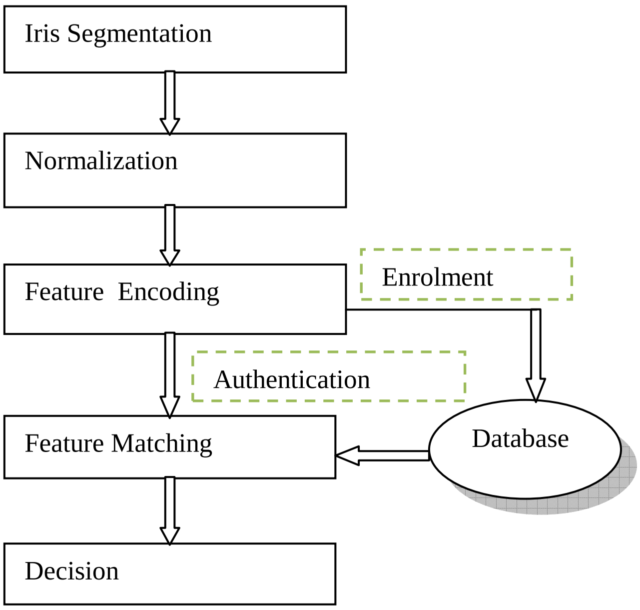

Figure 3: Iris Recognition Framework

A. Iris Segmentation

For Iris localization, Richard Wildes[5] approach has been used and it consists of 2 main algorithms namely CED algorithm and CHT.

The Flow chart of CED algorithm is as shown in figure 4. This algorithm consisting of 5 different steps: (i) Smoothing: convolution of image pixel values and gaussian smoothing filter values yields blurred image, this smoothing is mainly to remove the noise of image. (ii) Finding gradients: gradient magnitude is computed using two 3x3 convolution masks; after smoothing, to find out gradient values at each pixel, sobel operator is used. The sobel operators in two directions x and y are as shown in figure 5. Magnitude and direction of gradient of a pixel is given by the equations (1) and (2).

$G= (G_i^2+G_j^2)$ ----------(1)

$\theta=\arctan(G_j/G_i)$ ----------(2)

(iii) Non-maximum suppression: the pixels which are of non-maximum values will get suppressed and replaces values with '0' except points at local maxima. (iv) Double thresholding: in order to extract only the strong edges present in image, thresholding concept is used, which makes easier to remove false edge points. To make this thresholding more accurate double thresholding is utilized and two values are used, where one of the threshold value is double of the other. and (v) Tracking of edges by hysteresis: edge points will be tracked by suppressing the most of edges that are not have connection with the strong edge line.

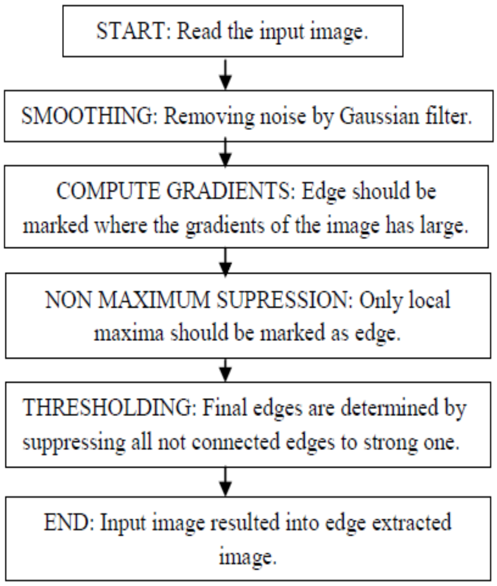

Figure 4: Flow chart for CED algorithm

-1	0	1
-2	0	2
1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

Figure 5: Sobel operators in x and y directions

The CHT is a fundamental procedure utilized as an important part in DIP, for recognizing objects with shape of circle in digital picture. The reason for this method is to find out circles in improper picture inputs. The circle edge points are created by "voting" in Hough parameter space and after that selecting the nearby maximum edge values in the matrix of accumulators.

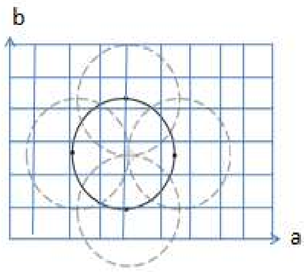

Figure 6: Circle detection using CHT

$(x-a)^2 + (y-b)^2=r^2$ -----(3)

$a = x- r \cos\theta$ -----(4)

$b = y- r \sin\theta$ -----(5)

Circle can be described by Equation (3), where x, y are coordinates of circle, a and b are centers of the circle and these can be written as given in equations (4) and (5), where as r is radius of the circle. The circle detection using CHT is as depicted in figure 6. Figures 7, 8 and 9 show preprocessed gray image of eye, edge detected eye image and iris segmented eye image using MATLAB.

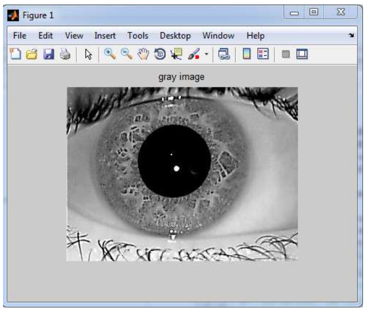

Figure 7: Preprocessed gray image of eye.

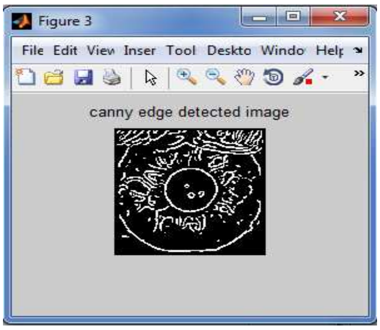

Figure 8: Edge detected eye image

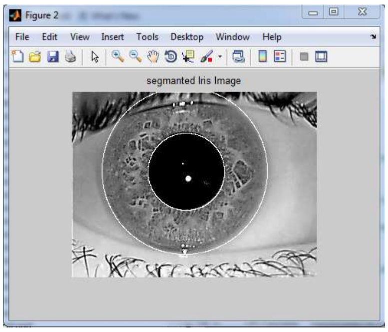

Figure 9: Iris segmented eye image

B. Iris Normalization

The process of normalization includes iris part unwrapping and conversion of it to the polar equivalent. It can be done utilizing rubber sheet(RS) model proposed by Daugman[6]. Daugman's RS model remaps each and every points inside region of iris from cartesian points (x, y) to the polar points (r,θ) of non-concentric normalized image, where θ is in the angle of range [0, 2π] and r is of the interval [0,1] as given in equation (6);

$I(x(r,\theta),y(r,\theta)) \rightarrow I(r,\theta)$ -----------(6)

The x(r, θ) and y(r, θ) in equation 6 are described as linear sequence of both sets of boundary points of pupillary region ($x_p(\theta),y_p(\theta)$) and limbos boundary points (xi(θ),yi(θ)), and are given by equations (7) and (8).

$x(r,\theta)=(1-r)x_p(\theta)+rx_i(\theta)$ -----------(7)

$x(r,\theta)=(1-r)y_p(\theta)+ry_i(\theta)$ -----------(8)

Due to the fact that pupil is non-concentric to the iris, so in implementation process, a formula of remapping is necessary to rescale the points and this depends on angles surrounding the circle as mentioned in figure 10, and are given by equations (9) to (11).

$r' = (\alpha\beta) \pm (\alpha\beta^2-\alpha-r_i^2)$ -----------(9)

with,

$\alpha = o_x^2+o_y^2$ ---------(10)

$\beta=\cos[\pi-\arctan[\frac{oy}{ox}]-\theta]$ ---------(11)

Displacement of the pupil's centre relative to the iris's centre are represented by o_x , o_y and r' symbolize the distance between edge of iris and edge of pupil at an angle θ around the region, and r_i radius of the iris.

After unwrapping phase, an image or a two dimensional array having radial resolution as vertical dimensions and angular resolution as horizontal dimensions is produced. Figure 10 shows the implementation of unwrapping step and figure 11(a) and 11(b) illustrates Daugman's RS model and its normalized iris image respectively.

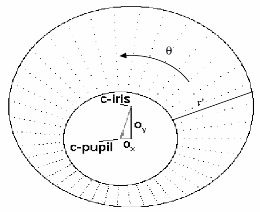

Figure 10: Implementation of unwrapping step

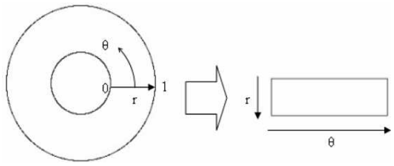

Figure 11(a): Daugman's RS model

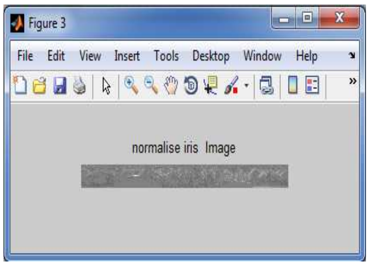

Figure 11(b): Normalized iris image

C. Feature Extraction using RED algorithm

In iris acknowledgment, RED algorithm is utilized to extract the feature of iris normalised image. Feature extraction depends on the bearing of the edges that show up on the picture [2]. The energy of every pixel is just the estimation of the infrared power inside the pixel and is used to identify features.

The flowchart for RED algorithm as shown in figure 12, tells that process of filtering the normalized iris of rectangle form with directional filters to decide the presence of edges and the orientation of edges . More particularly, the output of RED algorithms is to multiply filter values by the pixel values of normalized iris image. 9*9 RED filters are designed to filter and highlight the strong edges present in normalized image[2].

For the normalized rectangular iris image obtained from normalization process, the RED filters need to be applied. There are two filters made use, they are horizontal and vertical RED filters. These filters are designed in such way that, it should not corrupt original pixel values of normalized iris. The vertical and horizontal RED filters are applied on the normalized iris image. 9*9 filter values and 9*9 image pixel values are multiplied and added to get new filtered pixel , that is RED filtered feature extracted pixel value. On each and every pixel of rectangle iris, this process will be repeated, row by row, column by column throughout the image until this filtering should apply to the every pixels of the iris in rectangle form.

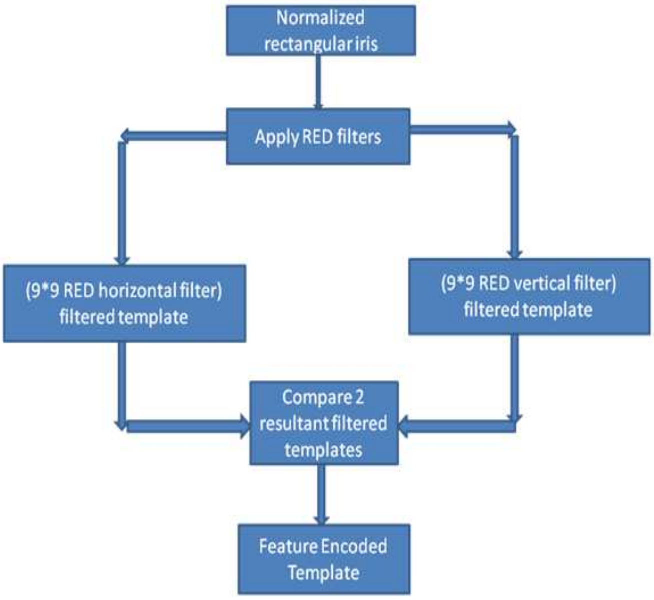

Figure 12: Flow chart of RED algorithm

D. Feature Matching using Hamming Distance

Generally the Hamming distance can be explained as, the number of different symbols at same positions while comparing the two strings of same length. In other words, the number of error bits present in one string when compared with other string in order to find out number of different symbols in two equivalent sequences.

In image processing, two images are to be compared pixel by pixel to figure out the number of different pixels between two images. The HD between two feature extracted iris templates is measured using equation. These two iris templates to be compared are of binary sequences and of equivalent length and corresponding bits are compared and hamming separation is measured. The identification of irises will be more accurate as the value of HD more close to zero.

$$HD = \frac{\|(Template\ A \otimes Template\ B) \cap Mask\ A \cap Mask\ B\|}{\|Mask\ A \cap Mask\ B\|} \quad \text{------(12)}$$

In the equation (12), template A and template B represents feature extracted template of captured image and feature extracted template of stored images. ⊗ is a symbol indicating binary Exclusive-Or operation to detect different binary symbols in two iris templates. ∩ is a symbol indicating binary AND operation. ‖●‖ symbol shows summation. Mask A and mask B represent the masks of binary values that are associated with template A and B respectively. The denominator part of equation ensures that calculation involves only the valid bits.

III. Results and Analysis

A. Simulation Results of RED And HD Calculation

The reading of normalized image using text files in order to import the image to Xilinx environment and performing RED calculation as said, finally the output of this consists of binary values 0 or 1, the Simulation result of RED algorithm is as shown in figure 13. In image processing, two images are to be compared pixel by pixel to figure out the number of different pixels between two images. The HD between two feature extracted iris templates is measured using equation (12). These two iris templates to be compared are of binary sequences and of equivalent length, corresponding bits are compared and hamming separation is measured. The Simulation results of Hamming distance are given in figure 14 and figure 15.

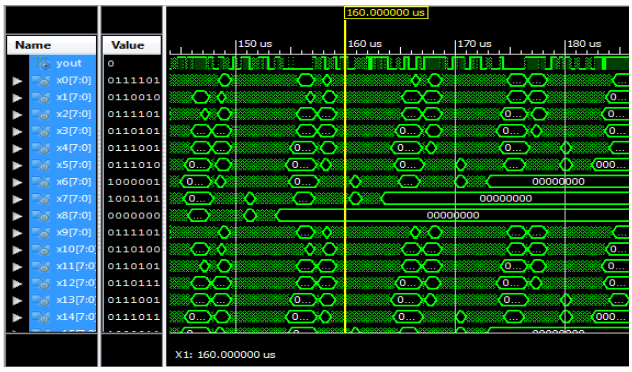

Figure 13: Simulation result of RED algorithm.

As observed in simulation results of hamming distance and experimental values obtained, for exact same two eye images HD is 0 and for different eye images there obtained different values as the two sequence of iris templates are consisting of different binary bits in equal positions. When the image is small enough for extraction of feature and for matching but as the resolution of normalized image increases, it gives more number of bits for matching which in turn increases the accuracy and gives more time delay but this can be compensated with threshold values.

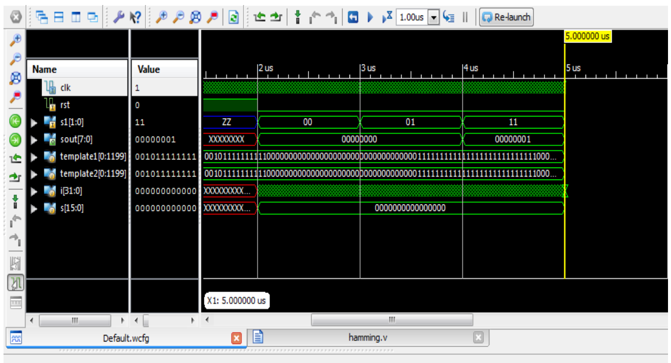

Figure 14: Simulation result of HD for same two eye images

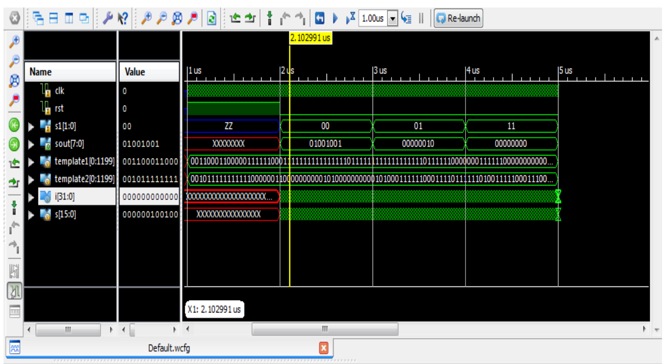

Figure 15: Simulation result of HD for two different eye images

B. *Synthesis Results*

i) Device utilization Summary

Table 1 shows the estimation of device/resource utilization summary of synthesis on FPGA.

TABLE 1: SUMMARY OF DEVICE UTILIZATION

Sl. No	Slice Logic Particulars	Utilization in numbers
1	BUFGs	1 out of 32 3%
2	External IOBs	12 out of 640 1%
3	LOCed IOBs	11 out of 12 91%
4	OLOGICs	5 out of 800 1%
5	Slices	2 out of 17280 1%
6	Slice Registers	0 out of 69120 0%
7	Flip Flops	0
8	Latches	0
9	LatchThrus	0
10	Slice LUTS	2 out of 69120 1%
11	LUT-Flip Flop pairs	2 out of 69120 1%

1) *Timing Summary*

For RED algorithm, the timing summary shows simulation time of 4.796 µs, fuse memory usage is of 27508 KB and fuse CPU usage is of 920 ms. Synthesize timing report of HD shows, speed grade is -1, minimum arrival time for input before the clock is 1.963ns and maximum time required for output after clock is 3.259ns, therefore total synthesizing time of hamming distance on FPGA is of 5.22ns.

IV. CONCLUSION

For eye image pre-processing, localizing the iris using CED algorithm and CHT, and for normalisation processes MATLAB software has been used. Xilinx ISE design suite has been used to simulate the RED algorithm for feature extraction, hamming distance for feature matching. Experimental test on the CASIA database has been carried out, the RED algorithm, hamming distance implemented and synthesized using Xilinx FPGA virtex-5 (XC5VLX110T). There is reduction in number of device utilization on board and simulation time delay of RED algorithm is 4.796 µs and synthesized time delay of hamming distance is 5.22ns. Hence achieved parallel processing using verilog modelling and improved time delay.

V. FUTURE SCOPE

Other algorithms - like active contour - could be used in segmentation to achieve localization that is more accurate. In addition, Support Vector Machine (SVM), neural networks, or other classifier may be used instead of hamming distance to increase the identification rate. Most current database from CASIA or other database could be utilized as a part of the test. To improve the time delay further, the advanced DSP kits can also be used and iris recognition can be implemented in real time.

REFERENCES

[1] Mateusz Trokielewicz, "Iris Recognition with a Database of Iris Images Obtained in Visible Light Using Smartphone Camera" IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2016), Feb 29 – Mar 2, 2016.

[2] Safaa S. Omran, Aqeel A. Al-Hillali, "Quarter of Iris Region Recognition Using the RED Algorithm" IEEE 17th UKSIM-AMSS International Conference on Modelling and Simulation, 2015.

[3] Khaoula Mannay, et. al, "Embedded system of the iris segmentation module" 7th International Conference on Modelling, Identification and Control Sousse, Tunisia - December 18-20, 2015.

[4] H. Proenca, "Quality Assessment of Degraded Iris Images Acquired in the Visible Wavelength" IEEE Transactions on Information Forensics and Security, pp. 82–95, 2011.

[5] Richard P. Wildes, "Iris Recognition: An Emerging Biometric Technology" proceedings of the IEEE, vol. 85, no. 9, september 1997.

[6] Daugman, J. G. "How iris recognition works" IEEE Trans. Circuits Syst. Video Technology, vol. 14, no. 1, pp. 21-30, Jan. 2004.