Authentication using Iris Recognition with Parallel Approach

Priyam Ghosh, M Rajashekharababu

School of Computer Science & Engineering VIT University, Vellore, India

Abstract
A biometric system is an automatic identification system based on a unique template or feature matching. Biometric system is one of the methods that is used now a days as an useful authentication system. Iris recognition system is among the most reliable and unique biometric identification system. The approach in this paper is to create an authentication biometric system using iris recognition with parallel approach. So far the works done in this field are mostly sequential and takes a lot of time to function. Our approach in this paper is to make the iris recognition system more efficient by parallelizing it.

Keyword:
Parallelization, Canny's edge detection, Circular haugh transform, Daugman rubber sheet model, Hamming distance, MATLAB, image processing, parallelizing and image acquisition toolbox.

1. INTRODUCTION

Biometric system is nothing but a authentication system that match the templates generated by the system, with the actual input factors or the images. If the templates are matched then authentication is given otherwise authentication is not given. There is a plenty number of biometric system available in the market, some of them are face recognition system, retina checking system, sound authentication system, heart sound detection system. The face recognition system works the same way by matching the previously stored templates from the database with the actual input face image. The system may also run for low resolution images. It may affect the quality of the image but for the system it is not a barrier to match the low resolution input image with the high resolution templates or patterns. The sound checking or the heart sound checking system works with the recorded heart sound which is further transformed into binary input matrix values. The retina checking system works with the retina portion of the human eye image and matches it with the system images. In all the cases the input factors are generated by the system. The input image or input factor then transformed into some mathematical forms that are considered to be the templates or the patterns [2], [4], [6 These iris code better known as templates or features or templates are then compared with the templates stored inside the database. There are two possibilities, the subject is identified or else it is not matched and the subject remains unidentified [2], [4], [8].This is how a biometric as well as an iris recognition system works.

Manuscript received May 5, 2013
Manuscript revised May 20, 2013

2. RELATED WORK

The iris recognition system is considered to be the most stable and constant reliable biometric system because the iris recognition system is the system which works with the iris area of the eye, and the iris area structure remains unchanged from 6 months age till death. The retina structure may change after a certain age but the iris area remains unchanged. That is why iris recognition system is considered to be more reliable and constant system than retina checking. The existing methods for iris recognition are mostly based on 4 stages; those are acquisition, segmentation, normalisation and matching. Some existing algorithms are proposed for each of the phases. Among them some efficient segmentation methods are edge detection [14], [15] and circular haugh transformation. In edge detection loads of methods are available, listed as, canny [14], sobel, Prewitt, Robert, zero crossing edge detection methods etc. For transforming the image circular haugh transform method is used. Daugman's rubber sheet method is used as an efficient normalisation method and cumulative sum based method is used in extracting the feature. Finally for matching Euclidian distance calculation or hamming distance calculation method is used. From the angle of feature extraction, previous iris recognition methods are generally divided into three major categories: phase-based methods [3], zero-crossing representation method [1], [2] and texture-analysis based methods [1]. Some parallelising approaches are proposed using FPGA hardware architecture [13]. In this paper, we propose a new iris recognition algorithm with parallel approach.

3. DESIGN

The iris recognition system as discussed above has 5 different phases and in most of the cases those are implemented in a sequential way. The sequential or the traditional model of the existing iris recognition system is given below.

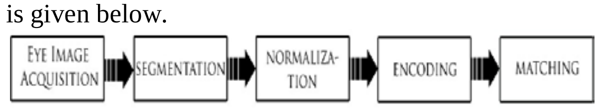

Fig 1: General diagram of phases of iris recognition system

All the 5 stages are implemented in a sequential manner and the time required for the execution in sequential manner is shown in the diagram below.

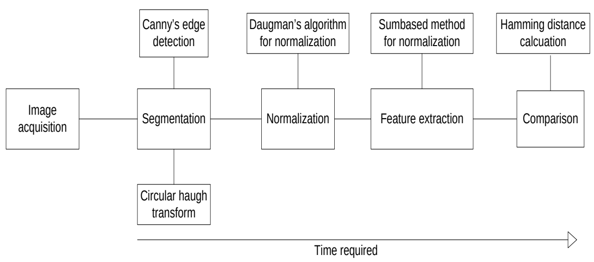

Fig 3: Time taken in sequential execution

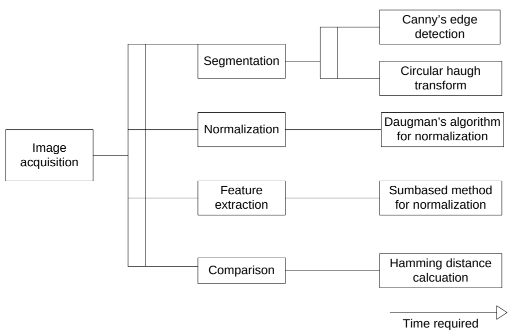

Fig 2: Time taken in sequential execution

As depicted in the diagram, in sequential processing, all the phases are implemented in a linear or sequential way and the time required is stretched from the very first phase to the last phase. This model is not efficient as it takes a huge amount of time to execute it in a sequential way. In our system the time required is much lesser than the sequential approach. In our approach all the phases are implemented in parallel so the execution time is much lower than the sequential approach. The time difference between the sequential and parallel approach makes the parallel system much more efficient than serial system.

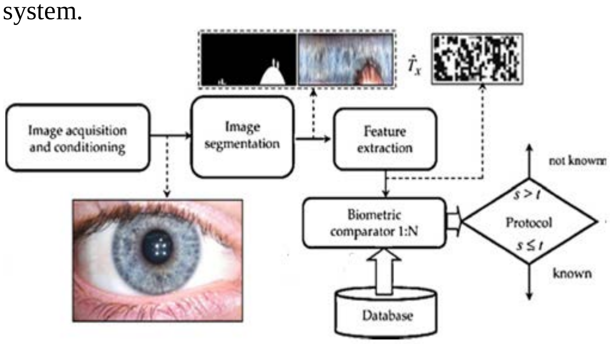

Fig 4: Block diagram of a typical iris identification system.

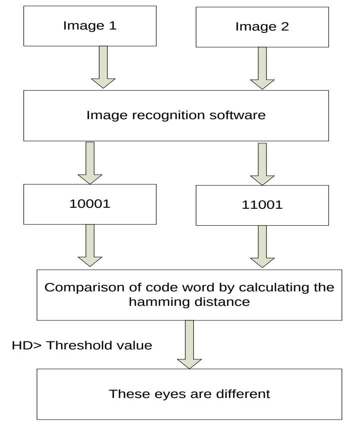

Fig 5: Basic functions

4. IMPLEMENTATION

The implementation of the system has five different phases. They are image acquisition, segmentation, edge detection, haugh transformation, normalization, cumulative sum based feature extraction and finally matching by comparing the hamming distance. All the phases are implemented in parallel way. The implementation details for each stage are given below.

A. Image Acquisition

Image acquisition is a method for capturing the image by web cam and cropping the useful part from it, and it may be implemented by the image acquisition toolbox available in MATLAB. In our system we are working with static images previously stored in the database, though we can use the image previewing by using the image acquisition toolbox.

B. Segmentation:

Segmentation is needed for pointing out that much portion of area which is needed from the whole eye image. It has two different phases. Circular haugh transform is one of the method for segmenting the portion. Segmentation [2], [4], [5] also includes the edge detection methodologies. Certain edge detection methods are there, like canny [13], sobel, Robert, zero crossing, prewitt, gabor wavelet [15] methods etc. In our approach we are using Canny's edge detection method [15]. Segmentation has two different subparts, they are,

a) *Edge Detection:* Edge detection [13], [14], [15] is the method for detecting the edge pixel coordinates of the eye image captured by the acquisition toolbox. The boundary coordinates are separated from the rest of the image based on the intensity value of the pixel and the strength of them. There are many methods that are used in detecting the edge like canny [13], sobel, prewitt, Robert, zero crossing methods. But the most efficient one in our approach is Canny's edge detection method [13].

b) *Circular haugh transform:* To find line patterns within an image, Hough transform is used. In finding circles from an image as in our case implementation using simple Hough transform is a tough task for that reason we will be using circular haugh transform (CHT). The Circular hough transform (CHT) aims to find circular patterns within an image. CHT aims to find circular patterns centre coordinates and the radius of the eye. The algorithm is given below

First find edges. For every point in the edge draw a circle with a center in the point in the edge with radius r and increase each and every coordinates that the perimeter of the circle passes through in the accumulator space. Thirdly, find one or several maxima in the accumulator space and lastly, map the parameters (r,a,b) according to the maxima of the original image.

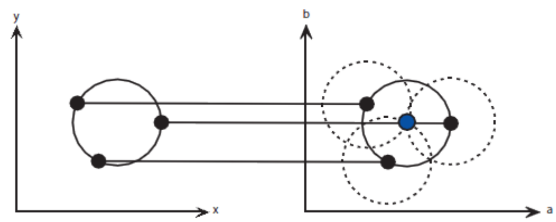

Fig 6: A Circular Hough transform from the x,y-space

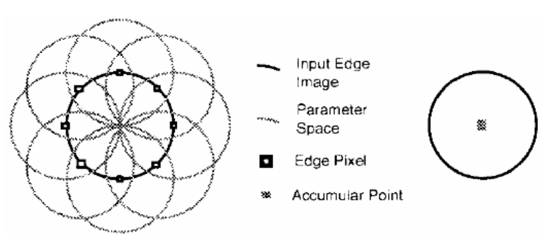

Fig 7: Edge points detected in the Accumulator space

C. Normalisation

Once the eye image is properly segmented, the next stage is to normalize it to allow comparison and for giving it a fixed dimension [11]. The problem in consistency is caused by the stretch and decrease in size of the pupil for the varying level of light and illumination [1], [2], [4], [7]. The method used to implement is Daugman's rubber sheet model. The normalization process will produce iris regions [6], which have the same constant dimensions. Its main task is to produce a rectangular stripe from the segmented iris region to form consistent values of rectangular matrix.

a) *Daugman's rubber sheet model*: The daugman's rubber sheet model developed by John Daugman which remaps each point within the iris area to a pair of polar coordinates (r,θ) where r is the radius on the interval of [0,1] and θ is angle varying from 0 to 2π.

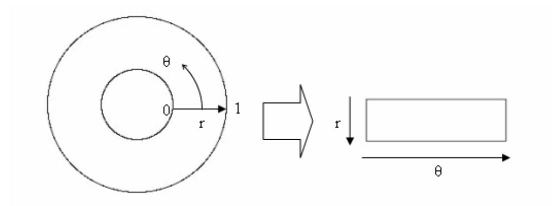

Fig 8: Normalisation of concentric region to rectangular 50 X 1000 pixel normalised matrix stripe.

D. Feature Extraction

To provide proper iris recognition of image, the feature or pattern of the iris region must be extracted properly. The feature extraction [1], [2], [3], [5], [7] is nothing but the process of generation some patters or templates from the normalised eye image to work with or to compare them with the input images. Feature extraction is the method of extracting the patterns or templates Various methods for feature extraction are available, like Daugman's method [8], sum based method [2], ordinal feature [5] extraction methods etc. In our approach, the cumulative sum based method is used as a feature extraction method.

a) Cumulative sum based change Analysis:

After normalization the normalized image is used for the feature extraction. The method is named cumulative sum based method because the cumulative sum is calculated from the cell areas or region. Overall feature extraction processing is described stepwise. Firstly, divide the normalized iris image into basic cell areas to calculate cumulative sums [2].(one cell area is a m x n pixel size). Secondly, group the cell areas in both horizontal direction and vertical direction. (Five basic cells are grouped into one group). Thirdly, calculate cumulative sums [2] over the each group using equation 2. Lastly, generate iris feature codes which are nothing but some pixel values in binary format stored inside and rectangular input matrix.

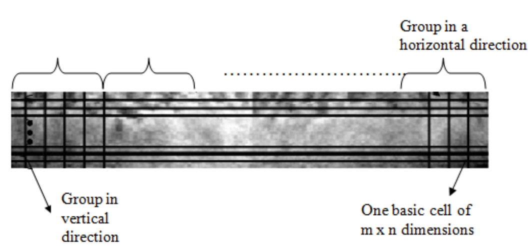

Fig 9: Cell grouping both horizontally and vertically

We have chosen the cumulative sum based change analysis method for feature extraction [2]. The image dimension taken here is 50x1000. Next we are grouping 10x10 pixels of the image in both horizontal and vertical directions into the input matrix. The average gray value is stored inside a 2D array. A maximum of five cells are taken in consideration for forming one group in both horizontal and vertical manner.

The cumulative sums are calculated as follows: Suppose that Y1, Y2……Y5 be the five representative values of each cell regions within group (horizontal or vertical).

$$\bar{Y} = \frac{Y_1 + Y_2 + Y_3 + Y_4 + Y_5}{5} \qquad (2)$$

First calculate the average. Then calculate cumulative sum from 0: CS_0 = 0. Calculate the other cumulative sums by adding the difference between current value and the average to the previous sum,

$$CS_i = CS_i - 1 + (Y_i - \bar{Y}) \text{ for i=1,2,3,4,5} \quad (3)$$

After calculation of the cumulative sums (CS_i), binary iris codes are for generated for each cell by comparing the cumulative sum values (CS_i). The binary iris codes are stored inside a rectangular metric. This algorithm generates iris codes by analyzing the changes of grey values of iris patterns [2]. This matrix of a particular dimension is considered as the pattern or the feature of the iris region in our system. It consists of basic binary pixel values stored inside matrix. This feature is considered as pattern in matching phase for calculating the hamming distance by comparison.

E. Matching

Matching phases comes in the last phase after all the initial phases and the templates or features generated in the feature extraction method is taken as the input value for matching the iris image with the iris codes[6]. The methods that are used for this purpose are Hamming Distance [2], [11], Weighted Euclidean Distance and normalised correlation, Fragile Bit Distance[11] etc. In this paper hamming distance [11] is applied for matching the templates or the iris code [6]. Hamming distance calculation is less complex than the Euclidean or Fragile bit distance calculation method. More over it simply calculate the difference between the iris codes or the distance by using the simple XOR operation over the plotted matrix of the binary iris codes.

a) Hamming Distance: Hamming distance[11] is a method that calculate the bit difference by using the XOR operation on the iris patterns. The distance calculated is nothin but the bit difference of the iris patterns. It calculate the distance or difference between two templates or patterns and detetecs that the input image's template and the image in the iris database are same or not by comparing the calculated hamming distance with certain threshold value previously set. In our case we are considering the threshold value 149.5. The hamming distance is calculated both horizontally and vertically by the equation given below.

$$HD = \frac{1}{2N}[(\sum_{i=1}^{N} A_h(i) \oplus B_h(i)) + (\sum_{i=1}^{N} A_v(i) \oplus B_v(i))] \quad (4)$$

Where $A_h(i)$, is the enrolled iris code in horizontal direction and $A_v(i)$ is the enrolled iris codes in vertical direction. And $B_h(i)$ is the input value of the iris code in horizontal direction, $B_v(i)$ is the input iris codes over the vertical direction. N is total number of cell and $\oplus$ means the XOR operator, its functionality is defined in the equation below.

$$x \oplus y = \{0,1\}[x = y, x \neq y] \qquad (5)$$

A certain threshold value is given for comparing the patterns with the hamming distance calculated. If the hamming distance exceeds the threshold value then the patters or templates are not matched, thus authentication is not given and if the hamming distance is lower than the threshold value, the authentication is given. The threshold value is determined by checking all the possible values in trial and error method. In our case the threshold value previously set is 149.5 which give best result among all and error acceptance in the lowest.

5. RESULT ANALYSIS

Finally after verifying the time taken in execution and the elapse time and after analyzing the result of the sequential or serial processing and parallel processing of the traditional iris recognition

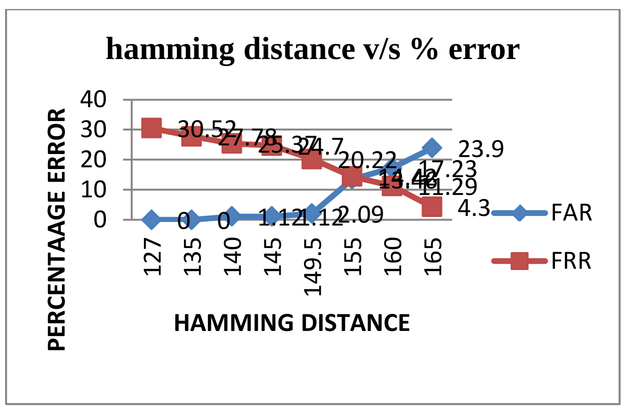

Fig 10: Hamming distance v/s error

system, we can come to a conclusion that the parallel processing of the traditional iris recognition system is much more efficient computationally and more reliable to work with. The result analysis report says, CPU usage is more than 95% (100% sometimes) and though it degrades sometimes, the average CPU usage is not less than 90% in any point of time in case of serial execution. The memory usage is also higher as it uses only one of the cores to execute, other cores are idle. The execution time is more than 60 seconds (depends on the number of images stored in the database) when it works with 50 iris image database. These implementation issues are the barriers in the sequential iris recognition system. But if we implement it in a parallel way with the parallel image processing toolbox in MATLAB using the parallel for loop and activating pmode and matlabpool open command the system usage is expected to degrade from 100% to 50% and the memory usage should also be lesser in the parallel approach. Comparing the system execution time with the sequential system we also recognize that the execution time is also lesser in the parallel approach as compared with the sequential approach. Hence the system is considered to be an computationally efficient parallel system.

The performance evaluation of proposed system was measured by the two error rates such as FRR and FAR. False Acceptance Rate (FAR) is the ratio of images accepted wrongly to the total number of images tested upon. False Rejection Rate (FRR) [2] is the ratio of images rejected incorrectly to the total number of images tested upon.

$$\text{FAR}(\%) = \frac{Number\ of\ false\ acceptance}{Number\ of\ imposter\ attempt} \quad (6)$$

$$\text{FRR}(\%) = \frac{Number\ of\ false\ rejection}{Number\ of\ authentic\ attempt} \quad (7)$$

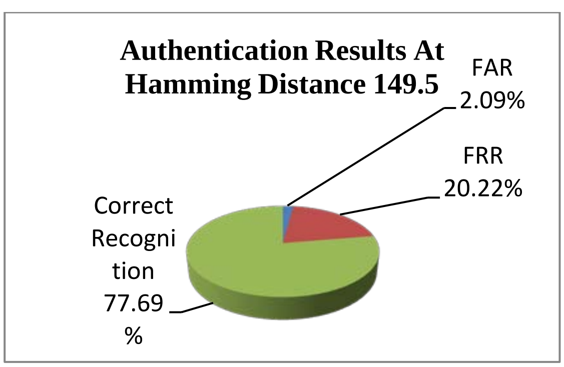

Fig 11: Authentication results for hamming distance (FAR, FRR)

We can measure the time difference between serial and parallel approach each time we increase the number of images in the database. But if we increase the number of images in the database the execution time does not vary from the case where lesser number of images was there in the database, and if at all it varies, the time difference is so small that it is almost negligible. The time taken in sequential approach for 750 images is varied from 190-220 seconds. But in parallel approach for the same number of images the time taken is reduced to 10-15 seconds.

Let's assume, for X_1 number of images the time taken is Y_1 and for $(2*X_1)$ number of images the time taken is $(2*Y_1)-Z_1$. Similarly for $(n*X_1)$ number of images, the time taken is $(n*Y_1)-(Z_0+Z_1....+Z_n)$ So execution time taken for n number of images is formulated as the equation given below-

$$\sum_{n}^{50} x = (n*y) - \sum_{n-1}^{i=0} z_i \quad (8)$$

Where, x is the number of images varying 50 to n. n is the maximum number of images. y is the initial time taken for 50 images. z is the amount of time which is less than double of the time taken in previous case varying from 0 to (n-1).

The table below shows the time taken in both serial and parallel approach for varieties of number of images ranging from 50 to 19840. By increasing the number of images in the database the execution time also varies in serial approach and the time difference is bit lesser than the double of time taken in the previous execution. But in parallel approach the time difference for doubling the number of images is very small and it is almost negligible.

Table 1: Number of images, time taken in serial and parallel manner

Number of images	Serial execution time	Parallel execution time
10	34	14.56032
20	52	14.87778
40	74	15.09722
80	89	15.23412
160	102	15.23421
320	120	15.51243
640	133	15.55521
1280	154	15.65332
2560	171	15.68422
5120	193	15.69922
1240	200	15.72112
2480	218	15.87641
4960	232	15.89546
9920	254	15.98721
19840	261	16.00213

We can measure the time difference between serial and parallel approach each time we increase the number of images in the database. But if we increase the number of

images in the database the execution time does not vary from the case where lesser number of images was there in the database, and if at all it varies, the time difference is so small that it is almost negligible.

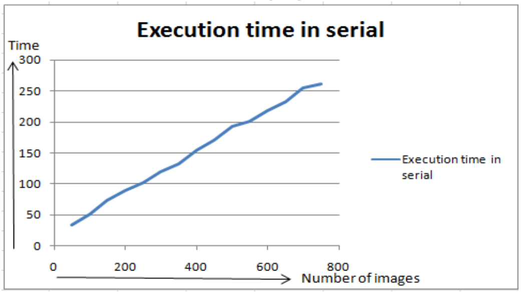

Fig 12: Execution time taken in serial approach

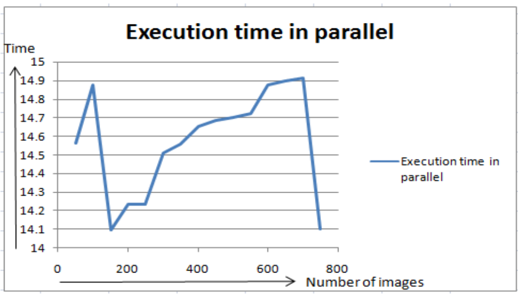

Fig 13: Execution time taken in parallel approach

As we can see in the graphs that the execution time increases gradually when the number of images is increased, but in parallel approach it is not determined that the time is increased or not when the number of images are increased. Sometimes the change is so small that it is almost negligible. The parallel approach is efficient enough when the number of images is high, because when the number of images is doubled, the time taken is almost the same as in the previous case.

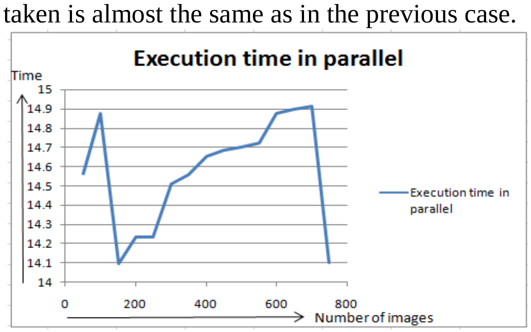

Fig 15: Performance in parallel approach

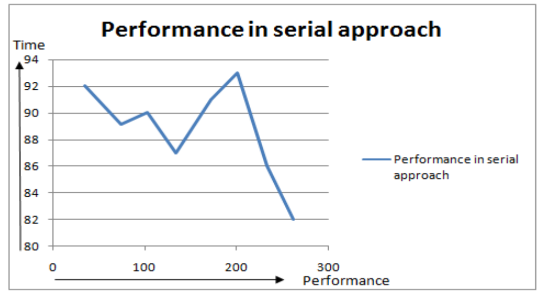

Fig 14: Performance in serial approach

The system performance in serial is not well determined when the execution time or the number of images in the database is increased. In that case sometimes the performance is higher and sometimes it degrades. The performance is higher as the number of images are increased or in other words number of images per time unit in increased in the parallel approach.

6. CONCLUSION

The segmentation resulted in an accuracy of about 95%. The most efficient edge detection method found is the canny's edge detection method and hence used with a positive constant upper and lower bound value. For normalization daugman rubber sheet model is used to normalize the image and to make it constant. Sun based feature extraction is used for extracting feature and generating template and iris patterns in an efficient way. For verifying or matching purpose hamming distance method is used amongst other various methods like Euclidian distance etc. To make the system parallel, parallelizing toolbox is used with parallel for loop and p-mode activation in MATLAB. All the existing methods are implemented in proper way both serial and parallel way. Finally after analysing the system performance and comparing the result or the system in both parallel and serial approach we can come to a conclusion that our system is computationally efficient in both the ways.

7. FUTURE WORK

Though the proposed system works efficiently in a parallel way, but still there are some places where further modifications can be done. Firstly, a camera can be attached with the system to allow the dynamic images in the system to work with authentication match and detecting unauthorised access. Liveliness detection is also one of the major problems associated with iris recognition system as camera is unable to check the liveliness of eye. Moreover the false acceptance rate (FAR) should be reduced than what is now. These problems should be considered in the future work. The system should consider more on the CPU and memory usage as they are the main factor that should be reduced during parallel execution of the system by utilizing all the four cores of the processor. In this case the system is parallelised by modifying the for loops to parallel for loops in MATLAB and executed in four CPU cores. In future the system may give even much more efficient results if it runs on more than 4 cores (can be considered up to 16 cores of the CPU). So the image acquisition and the parallel execution are the two fields where further modification can be done in future for betterment of the system.

REFERENCES

[1] Li Ma, Tieniu Tan, Yunhong Wang and Dexin Zhang, Personal Identification Based on Iris Texture Analysis, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 12, DECEMBER 2009, pp -1519-1533

[2] Jong-Gook Ko , Yeon-Hee Gil and Jang-Hee Yoo, Iris Recognition using Cumulative SUM based Change Analysis, International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS2006) Yonago Convention Center, Tottori, Japan, pp-275-278

[3] Kazuyuki Miyazawa, Koichi Ito, Takafumi Aoki, Koji Kobayashi and Hiroshi Nakajima, An Effective Approach for Iris Recognition Using Phase-Based Image Matching, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 30, NO. 10, OCTOBER 2008, pp – 1741-1756

[4] Zhaofeng He, IEEE, Tieniu Tan, Zhenan Sun and Xianchao Qiu, Toward Accurate and Fast Iris Segmentation for Iris Biometrics, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 9, SEPTEMBER 2009, pp- 1670-1684

[5] Zhenan Sun and Tieniu Tan, Ordinal Measures for Iris Recognition, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 12, DECEMBER 2009, pp- 2211-2226

[6] Adams W. K. Kong, David Zhang, and Mohamed S. Kamel, An Analysis of IrisCode, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 19, NO. 2, FEBRUARY 2010, pp- 522-532

[7] Chia-Te Chou, Sheng-Wen Shih, Wen-Shiung Chen, Victor W. Cheng, and Duan-Yu Chen, Non-Orthogonal View Iris Recognition System, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 20, NO. 3, MARCH 2010 417, pp- 417-430

[8] John G. Daugman, High Confidence Visual Recognition of Persons by a Test of Statistical Independence, IEEE TRANSACTIONS ON PA'ITERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 15, NO. 11, NOVEMBER 2007, pp- 1148-1161

[9] Wenbo Dong, Zhenan Sun and Tieniu Tan, Iris Matching Based on Personalized Weight Map, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 9, SEPTEMBER 2011, pp-1744-1757

[10] Kamil Grabowski and Andrzej Napieralski, Hardware Architecture Optimized for Iris Recognition, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 21, NO. 9, SEPTEMBER 2011, pp- 1293-1302

[11] Improved Iris Recognition through Fusion of Hamming Distance and Fragile Bit Distance, Karen P. Hollingsworth, Kevin W. Bowyer and Patrick J. Flynn, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 12, DECEMBER 2011, pp-2465-2475

[12] Ryan N. Rakvic, Bradley J. Ulis, Randy P. Broussard, Robert W. Ives and Neil Steiner , Parallelizing Iris Recognition, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 4, NO. 4, DECEMBER 2009, pp-812-823.

[13] Paul Bao, Lei Zhang, and Xiaolin Wu, Canny Edge Detection Enhancement by Scale Multiplication, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 9, SEPTEMBER 2005, pp – 1485-1490.

[14] Yang Zhang and Peter. I Rockett, The Bayesian Operating Point of the Canny Edge Detector, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 15, NO. 11, NOVEMBER 2006, pp-3409-3416.

[15] Wei Jiang, Kin-Man Lam, and Ting-Zhi Shen, Efficient Edge Detection Using Simplified Gabor Wavelets, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 4, AUGUST 2009, pp- 1036-1047.