Integro-Differential Operator (IDO)-A Computational Approach For Iris Localization

Shikha, Sonal Beniwal

Abstract: Now a days, problems becoming more and more complex and demand of better and fast solution in lesser time we need a fast algorithm to processes the data intensive and do computations on large data set. One of the approach for such a technique is making the small bundles of given data, and then processing the data of small bundles using parallel processing, as a result we reduce the time of execution for processing the information.

Keywords: IDO, Localization, Parallel Computing, CH Transform, Iris Reorganization

I. Introduction

In early era, while performing computation, we keep in mind complexity arising, in both space and time and try to make a program efficient in both time and space frame. But with the change in time and technology complexity[1-3] in space, is now not a constraint, because of better hardware in terms of memory space. On the other hand, the time space is still a limit, and always demand is there to reduce the processing time. So a lot of work is going on to reduce processing time to processes the information fast and efficient manner[4-6]. So for receiving the required quality of outcome, we used two processes one is design and develop fast hardware and other is developing an algorithm for high-performance computing. The main reason for developing an algorithm for high-performance computing is getting speedup[1-2, 7-8]. Many fields like weather forecasting, financial modeling, Medical imaging and diagnosis, scientific computation, Image processing, Statistics, etc are application of technique called nonlinear processing with parallel computing with massive data input. In general algorithm are written for serial computation and software program is developed to run on a single processor, during this approach a problem is sub divided into an isolated instructions with a series of different execution steps to be performed one by one in a specified pattern (series) and there is a limit to execute one and only one instruction at a time using serial processes.

II. Processes

During parallel processing a computational problem is divided into distinct sub-parts so that these sub pats can be processed in tandem; each sub part is further separated into steps of instructions; further programmer employs coordination mechanism simultaneously to execute processing instructions. In this processes we get results faster than any other method. The basic object at the back of parallel computing is speed up. Using the basic elements of measurement, we compare the speed of execution between the two approaches i.e. serial & parallel approach and how much it is faster than its counterpart Amdahl's defined[9], "Speed up (S)" as:

$$s = \frac{T_s}{T_p} = \frac{1}{\left((1-f)+\frac{f}{n}\right)} \qquad (1)$$

where,
T_s is execution time of a program executed sequentially on a given processor, and
T_p is time of execution of a program when it is executed on parallel "*p*" processors.

Here we assumes that for a code that is "inherently serial a fraction *(1−f)"* of the execution time involves and for "infinitely parallel with no scheduling overhead" fraction " *f"is involved*. Further the factors effecting processes of parallelism are:

1. Dependency of Data
2. Synchronization
3. Communication Latency
4. Load balancing

The reasons of not getting enviable speed, while using parallel processing is due to improper Communication between the executed task and scheduling of the processes. The other factor that resulted in inefficiency is the "synchronization" as during synchronization processes[10-11], all tasks which are "executed" must "completed" first before the start of next set of instructions, as a result the computation time increases.
The different methods for iris localization are:

- Integro differential operator
- Circular Hough Transform
- Essams Approach
- El-Bakry and Other Methods

Integro Differential Operator
Given by Daugman,[2, 6–8] and he used it for detecting the circular shaped region of pupil and iris in captured image. Further he used to detect the arcs of lower and upper regions of eyelids. Integro differential operator is mathematically expressed as:

$$Max(r, x_0, y_0) \left| G_\sigma(r) * \left(\frac{\delta}{\delta r}\right) \int_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds \right| \qquad (2)$$

We continuously applies the operator on the captured image with the objective of getting a "maximum contour integral derivative" with continuously increasing radius at "successively finer scales" at the three parameters, i.e. center coordinates (x_0, y_0) and radius (r). Similarly we localize upper and lower eyelids. In Daugmans algorithm we used first derivative of the picture and also search for geometric variable, so we does not face the threshold problem. However, if noise is present in the captured image which may be due to reflection, than we did not get the required results. After localization of Iris, a 2048 bit code is

- *Shikha, Department of Computer Science & Engineering, BPSMV Khanpur Kalan, Sonepat Email- 1995lambashikha@gmail.com*
- *Sonal Beniwal, Department of Computer Science & Engineering, BPSMV Khanpur Kalan, Sonepat*

calculated and then compared with templates that present in the database. These patterns are encoded with the help of 2D Gabor demodulation.

Circular Hough Transform

Hough transform is based on the, “Property of isolating features of a particular shape” from a captured image. Hough transform for circle is represented as:

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

$$h(x_j, y_j, x_c, y_c, r) = \begin{cases} 1 & if\ g(x_j, y_j, x_c, y_c, r) = 0 \\ 0 & otherwise \end{cases}$$

$$g(x_j, y_j, x_c, y_c, r) = (x_j - x_c)^2 + (y_j - y_c)^2 - r^2$$

Where, (x_c, y_c) are the coordinates of centre of circle and the radius is defined by r.

The steps used during Circular Hough Transform are:

1. Firstly, by applying edge detector like Morphological, Canny, or Sobel operation we determine all the edges present in captured images.
2. We, draw a circle with definite radius, at each edge point after applying edge detector.
3. Using voting procedure we find Maximum intersected point, and the obtained point becomes the potential center of circle.

Essams Approach

For high and acceptable accuracy and quick processing of iris segmentation processes he proposed three level morphological and threshold computation. Here, the localization is performed in two steps. In first step acquired image is processed to get grey image using three level thresholding. On grey image he applied morphological process for pupil detection; here the step is defined as “Coarse stage”. While in second stage the image size is reduced to one-fourth and IDO algorithm is applied to find iris and pupil boundaries and is known as “Fine stage” Neural Network based algorithm is used in El-Bakry[11] approach for iris localization. Bonney [12] uses LSB plane of image for detecting the pupil in acquired image, and processes the data using erosion and dilation operations for locating pupil, and after that limbic boundaries are detected using the standard deviation in the vertical and horizontal direction. The process of iris recognition is sub divided into four sub processes i.e. segmentation followed by normalization, and then encoding and in last matching. Each sub-stage applies an algorithm based on different technique. The final segmentation of the iris boundaries is carried out in the following steps.

1. Set the index of the current step to $i = 0$.
2. Initialize a contour C on the given image.
3. Dilate the current contour C_i into its contour neighborhood $CN(C_i)$ within an outer contour OC_i.
4. Identify all the vertices belonging inside and outside the initial contour C as the source S_1 and the sink t_i, respectively.
5. Compute the S-t min cut to obtain a new boundary that better separates the inner boundary from the outer boundary.
6. Terminate the algorithm if the resulting contour c reoccurs, otherwise set I = I + 1, and return to step 1.

III. Implementation

The research experiment is conducted using Matlab R2010a. Iris recognition has to pass through the segmentation technique and normalization technique to produce a template to measure the biometric system. The flow starts by first getting the data from iris CASIA database. The iris is reduced to smaller size. Then the iris is converted from RGB to grayscale iris image. After that, the random noise is applied to the grayscale iris image for removing the noise. It is to provide better texture of iris and iris information. Then, the pupil localization and the iris localization were extracted. The iris feature is produced through the segmentation technique, either using Hough Transform or Integro-Differential Operator. There are two forms of information we gathered from iris feature; pixel values and the binary code. Once this information achieved, it is stored into the database. For matching process, hamming distance is used for detecting the same binary code either in the database is same or vice versa.

IV. Result

For the computation processes we used Intel Core i7 CPU, with 8 GB system memory for implementing “serial processes” while for Parallel processes we used NVIDIA 64 core work station GeForce 860 GPU, with 8 GB system memory device. Further we used iris images of 640 * 560 size [13-15]. After computation, we found that execution time is fifteen percent less for parallel algorithm in comparison to that we used for serial algorithm.

V. Conclusion

The evaluation of samples and complex algorithm are the bottle neck to develop new systems for the biometrics research community. In this paper, we tried to discuss various approaches used for computation of biometric system. We further discussed the factors that control the computational features of biometric samples, which make the processes complicated. We tried to present different bottlenecks present in system design for various applications. Further the parallel implementation of edge detection part of iris localization will help in making fully parallelized Iris recognition system.

VI. References

[1]. Ananth Grama, and et.al. “Introduction to Parallel Computing”. Pearson Education, (2007).

[2]. L. Flom and et.al. “Iris recognition system”. (U.S. Patent) 4, (1987).

[3]. J. Daugman. “Biometric personal identification system based on iris analysis”. (U.S. Patent), 29160, (1994).

[4]. A.L. Yuille, and et.al. “Feature extraction from faces using deformable templates”. Computer Vision & Pattern Recognition, (IEEE Conference) 104-109, (1989).

[5]. Y. Huang, and et.al. “An efficient iris recognition system”. International Conference on Machine Learning and Cybernetics, volume 1, 40-44, (2002)

[6]. R.P. Wildes. “Iris recognition: an emerging biometric technology”. Proceedings of the IEEE, 8(9):1348-1363, (1997)

[7]. Rajesh Bodade and et.al. “Novel approach of accurate Iris localization form high resolution eye

images suitable for fake iris detection", International Journal of Information Technology and Knowledge Management, 3(2), 68-690, (2010).

[8]. J. G. Daugman: "High confidence visual recognition of persons by a test of statistical independence". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1, 1148 - 1161, (2013).

[9]. J. G. Daugman. How Iris Recognition Works". IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 21 - 30, (2004).

[10]. J. G. Daugman. "The importance of being random : statistical principles of iris recognition". Pattern Recognition, 36(2), 279 - 291, (2003).

[11]. EssamM. "An Efficient Iris Localization Algorithm". 29th National Radio Science Conference, Egypt, 285-292, (2012).

[12]. Hazem M. El-Bakry. "Fast iris detection for personal identification using modular neural networks". IEEE ISCS, 52-55, (2001).

[13]. B Bonney, and et.al. "Iris pattern extraction using bit planes and standard deviations". 38th Asilomar Conference, 1, 582- 586, (2004).

[14]. J. Cooper, "Location of the pupil-iris border in slit-lamp images". ICIAP, (1999).

[15]. Fatma Z. and et.al. "High Performance Iris Recognition System On GPU", IEEE ICCES, 237-242, (2011).