

Multicore and GPU Based Pupillometry Using Parabolic and Elliptic Hough Transform

Annah Nasim, Adnan Maqsood, and Tariq Saeed

Research Centre for Modeling & Simulation, National University of Sciences and Technology, Islamabad, Pakistan

Email: annah.164@gmail.com, adnan@rcms.nust.edu.pk, tariq@rcms.nust.edu.pk

Abstract— Pupillometry is the measure of pupil size and has pertinent applications in the area of neuroscience, cognitive behavior assessments and different psychologically evoked responses. In this work, Hough transform, a high-fidelity accurate technique, is used for pupil detection and measurement. The eye images are taken from an off-the-shelf webcam. During the eye segmentation process, eyelash-eyelid detection is based on parabolic Hough transform whereas iris and pupil detection is done through elliptic Hough transform. The developed eye segmentation algorithm show high accuracy, however, is computationally expensive. To deal with the problem, a parallel data distribution framework is employed that uses the raw computational power of emerging multicore CPUs and many core GPUs thereby boosting the performance of proposed algorithm. A performance comparison is carried out for sequential and parallel framework of proposed algorithm. The experimental results indicate a speed-up with a factor of 1.86x and 3.56x on a Core i7 CPU and Tesla T10 GPU respectively. The proposed parallel approach can be used to accelerate pupillometry applications on multicore and GPU based high performance computing platforms.

Index Terms—pupillometry, pupil dilation, cognitive load, Hough transform, parallel computing

I. INTRODUCTION

Pupil-size variations consists of dilation and constriction and are proven to be more sensitively linked with the psychologically invoked responses than the physiological ones [1]. The concept for capturing meaningful pupillary responses became authentic with the seminal work provided by Hess and Polt [2], [3] and Beatty and Wagoner [4], [5]. Numerous experimental studies have been conducted subsequently on the effects of emotion and task evoked mental effort on pupil size changes [6]-[10].

A review by Laeng et al.[11] emphasize on the direct correlation between pupillary changes and the ongoing neurological activity in the human autonomic nervous system. The greatest promise of pupillometry may lie in its ability to reveal online processes that occur in clinical subjects who are either preverbal [12]-[15] or simply lack language (animals) [16]. One potential application of pupillometry could involve subjects that cannot normally understand instructions or provide controlled, verbal

responses (aphasic patients, patients with locked-in syndrome).

There are wide number of application areas where pupillometry has established its significance. Palinko et al. [17] show that pupil diameter can be used as a physiological measure of cognitive load in driving simulators. Kun et al. [18] in a recently published study confirm that it is feasible to use pupil diameter to differentiate between parts of the dialogue that increase the cognitive load of the drivers. The goal of the study is to build a spoken dialogue system that can adapt its behavior when the driver is under high cognitive load either from the driving task or dialogue task. From medical diagnostic perspective, Gaigg and Bowler [19] and Kuchinke et al. [20] in 2011 use a pupillometric evidence to support the hypothesis that functional connectivity between cognitive and emotional processing is impaired in patients suffering from Asperger's Syndrome. Pupillometry has also found applications in an authentic measure for early diagnosis of different neurodevelopmental disorders in infants and young children by testing their mental activity. Daluwatte [21, 22] suggests that the Pupillary Light Reflex (PLR) differences are implicated in a wide range of neurodevelopmental disorders. Daluwatte [22] and Parnandi [23] observe a correlation between heart rate variability and pupillary fluctuations on a small sample space. Jianga et al. [24] explore how pupillary response reflects mental workload changes in a surgical environment. A simulated surgical task is conducted that has three different subtasks with different mental workload requirements. It is found that a significant effect among these different subtask groups is observed by measuring pupil diameter change rate. The findings can improve patient safety in a real operating room by non-intrusively monitoring the surgeon's cognitive workload while performing a surgery using an eye-tracking system.

Hough transform, a key component of the proposed methodology, provides an added advantage over other methods in terms of accuracy. Chen et al. [25] propose a parallel Hough transform scheme after comparing the coarse-grain and fine-grain parallelization techniques for Hough transform on multi-core processors. Wu et al. [26] presented two additional parallel frameworks based on Thread Building Block (TBB) and CUDA. Their results show that algorithms of circle detection are extremely

good in consideration with the amount of the processing time. Braak et al. [27] introduce two novel methods for the Hough transform on a Graphical Processing Unit (GPU), one is a fast method and other is an input-data independent method. Hough space is much larger when detecting circles as compared to lines. It is even larger for detecting ellipses and parabolas. Therefore, due to the amount of computation required for scanning and detection of parabola and ellipse, the speed of the algorithm is reduced.

In this work, the detection and assessment of pupil diameter is achieved by employing a low-cost eye tracking system. The scope of the system development is to detect pupillary responses over a period of time. The features of the proposed framework are eye and pupil-localization, pupil area and diameter calculation. For this purpose, human eye images are acquired by a webcam and processed offline. The hardware includes a Logitech®C920webcam with a USB data transfer capability to computer. The components of algorithm development include; eyelash recognition through parabolic Hough transform, iris and pupil localization using elliptic Hough transform and pupil area and diameter extraction. Moreover, a high performance implementation of the proposed algorithm is done using MATLAB® parallel processing toolbox [28]. The framework is based on simple computer vision techniques with easy to implement setup and has the ability of pupil-tracking with translational and rotational invariance. A performance evaluation is conducted on the CASIA Iris database [29] to show the pupil detection capability of the proposed program. Finally, the single core methodology is compared with multicore as well as GPU based implementation on a standalone machine to assess computational efficiency.

II. MATERIALS AND METHODS

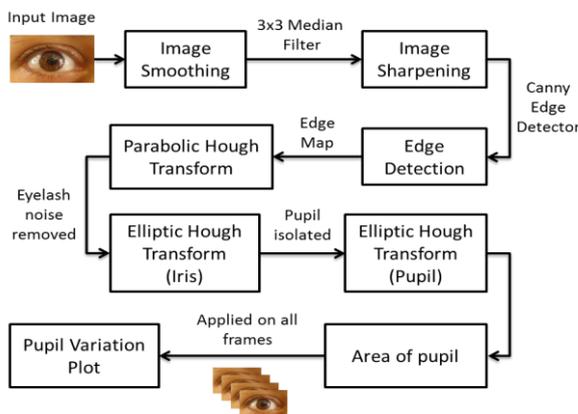


Figure 1. Flow Diagram for the proposed methodology

Segmentation is defined as classification of an image into its component regions. In this paper, a method for pupil extraction is presented using a noisy input from a low cost webcam. Initially the Region of Interest (ROI) is extracted from the full frame. After removing noise, the edges of eye-lashes are detected. The same procedure is

adopted for iris and pupil detection. The objective of the proposed algorithm is to systematically remove redundant edges and isolate the pupil region. The constraint of the algorithm is that the subject head should remain stationary during the video. Fig. 1 shows the methodology flowchart of the proposed algorithm:

The Hough transform is a classic computer vision technique used for the detection of arbitrary shapes in an image such as circles and ellipses. A voting method is carried out in a parametric domain, from which objective candidates are acquired as local maxima in the accumulator space that is constructed by the algorithm for computing the Hough transform. As a first step in preprocessing, the raw input data, image is manually cropped by identifying the region of interest in the input image. In this case, the region of interest constitutes the whole eye area. Hence, a 600x900 pixel region is cropped from the 1920x1080 input images as shown in Fig. 2. It can be observed that the pupil is the part of highest intensity. Therefore, thresholding is applied to separate the regions with high intensity close to 255 i.e. pupil, iris and eyelashes from regions of low intensity that is made up of skin.

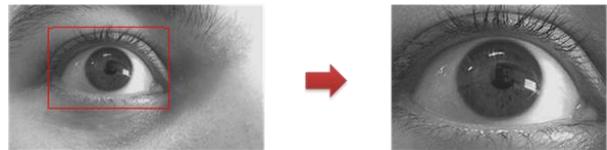


Figure 2. Original Image from camera (left), Cropped Image (right)

Image smoothing is done to smoothen the abrupt noisy jumps in the image called ‘dots’ or ‘speckles’. A large kernel size filter causes more blurring which can result in the smearing out of the value of a given pixel over a larger area of the image. A small kernel size causes less blurring and allows detection of small and sharp lines. Hence, a median filter of kernel size 3x3 is applied to fine-tune the edges. The filter replaces the noisy pixels by the median value of all its neighboring pixels.

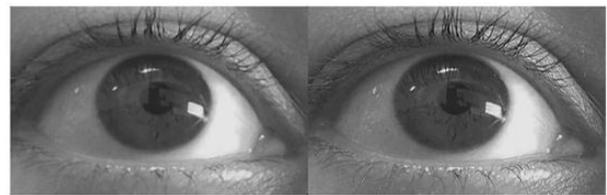


Figure 3. Input image (left) and sharpened image (right)

Edge-based segmentation is applied afterwards using Canny edge detector [30] as shown in Fig. 4. Canny edge detection is a four phase process that includes a Gaussian filter applied to remove the speckle noise. A gradient operation is applied for obtaining the intensity and direction. Non-maximum suppression is carried out to determine if any pixel is a better nominee to be considered for an edge than its neighboring pixels. Hysteresis thresholding is applied finally to set the beginning and end of edge intensity. Two thresholds with

hysteresis are used to allow more flexibility than a single-threshold approach. A threshold set too high can skip important information. Alternatively, a threshold set too low will identify false and irrelevant edges. In this case, the threshold values that produce most accurate results are dependent on lighting conditions and therefore adjusted on ad-hoc basis.



Figure 4. Edge map obtained after applying Canny edge detector and thresholding

After obtaining a clean edge map, Hough transform [31] is used to detect and isolate features of parabolic and elliptic shapes within the image. Votes are stored in the Hough space that is two dimensional for the Hough transform in lines. The size of the Hough space is determined by the size of the input image and the required accuracy for the parameterization of the lines. The advantage of Hough transform is its tolerance towards occlusions. The gaps in the edges are reasonably unaffected by redundant edges using this approach. Hough transform is insensitive to rotation and translation of the input images as the voting procedure depends on the highest number of votes and not the order of edges. Moreover, it requires that the desired edges be specified in mathematical (algebraic or polar) form. For eyelash detection, parabolic Hough transform is applied to the upper half of the input eye image, using pixel coordinates on x and y-axis and the stretch defining factor r in the input 2-D image to generate the Hough space for parabolas. The algorithm uses the Cartesian equation for the parabola in vertex form:

$$y = r(x - x_o)^2 + y_o \quad (1)$$

where x_o and y_o are the coordinates of the parabola vertex. The upper left corner of image is considered as the origin of coordinate system. r is the factor defining the shape and extent of opening of the parabola. For positive r , the parabola opens upwards. Size for 'x', 'y' and 'r' are 900 pixels, 600 pixels and 96 respectively. Therefore the Hough-Array becomes of the size 600x900x96. The function takes edge image as an input and returns detected parabola points as output (Fig. 5).

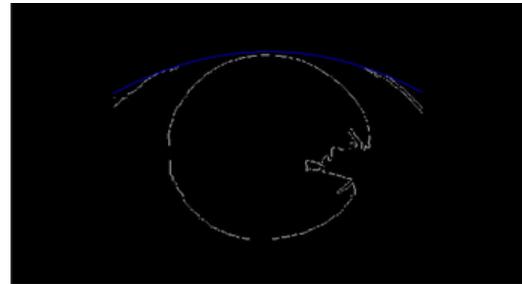
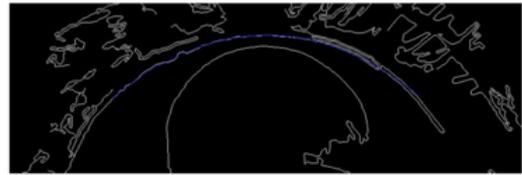


Figure 5. Detected parabola (top) and filtered extra edges (bottom)

A total of fifteen parabolas are found in this case. For each edge point in the original image and non-zero x, y , the algorithm calculates the first dimension value (x_o, y_o) and votes in the accumulator. The algorithm then finds all the possible parabolas in the given range, votes for them in the Hough space, and finds out the parabola containing the largest number of votes or points. Subsequently the peak point in the Hough transform space is found.

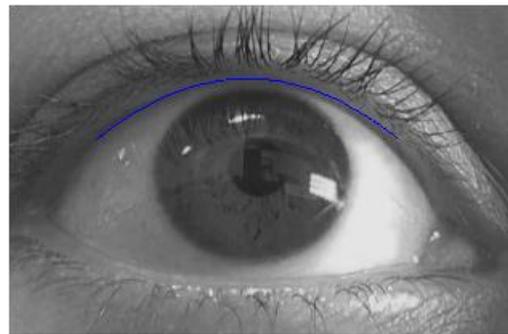


Figure 6. Detected eyelash parabola on the grayscale image

For iris and pupil detection, elliptic Hough transform is applied to the rest of the edge eye image, to generate points in the Hough space for all possible ellipses. The algorithm uses the Cartesian equation for the parabola in vertex form as:

$$\frac{(x - x_o)^2}{a^2} + \frac{(y - y_o)^2}{b^2} = 1 \quad (2)$$

where, x_o and y_o are the coordinates of the ellipse center, a and b represent the length of the major and minor axis of the ellipse respectively. If $a > b$, then a is considered as the major axis and vice versa. Size for x is 900 pixels, size for y is 600 pixels and for a and b , the size of the window is 20. Therefore, the accumulator array becomes of the size 600x900x400. The upper left corner of image is considered as the origin of coordinate system. The function takes edge image as an input and returns detected ellipse points as output. About 400

ellipses are checked i.e. 400 combinations of a and b for the entire image are generated. The accumulator votes are stored for each combination in a separately indexed array. For each edge point in the original image, where x and y are both not zero, the algorithm calculates the first dimension value and votes in the accumulator. The algorithm then finds all possible combination of ellipse in the given range, votes for them in the Hough space, and finds out the ellipse having the largest number of votes or points and maps ellipse over the iris. The detected iris from the program is shown in Fig. 7 and is superimposed on the grayscale image as well.

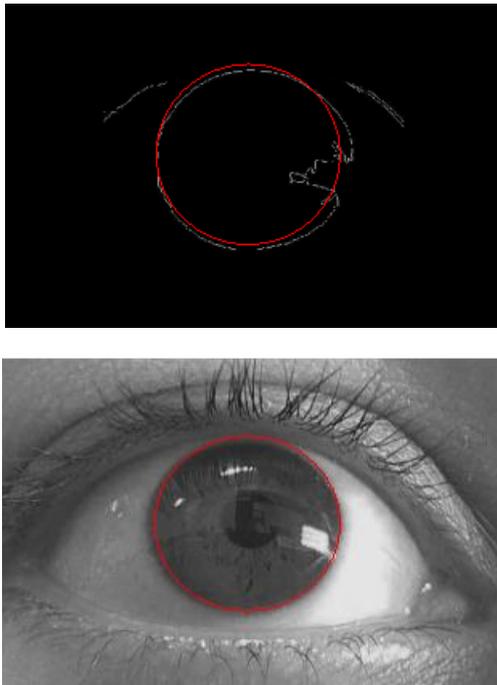


Figure 7. Detected iris on the edge-map (top) and on the original grayscale image (bottom)

The iris detection helps in the isolation of the pupil from the rest of the image known information as it is located inside the iris. Fig. 8 shows that all points on and outside of the perimeter of the iris are deleted and the remaining edge image is searched for pupil.

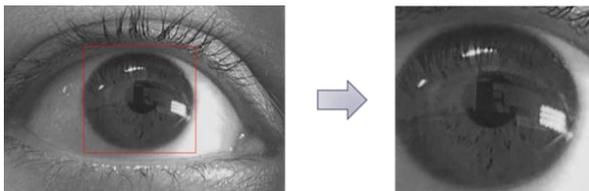


Figure 8. Isolated iris region cropped

Same process is repeated as of iris detection for the pupil localization with the input range for major and minor axes decreased accordingly. Elliptic Hough transform with array accumulator is used to identify the pupil location as shown in Fig. 9. Finally, the area of eye's pupil is calculated using area of detected pupil

ellipse i.e. $Area = \pi ab$ to measure the pupil response under a series of cognitive stimuli in a digital video.

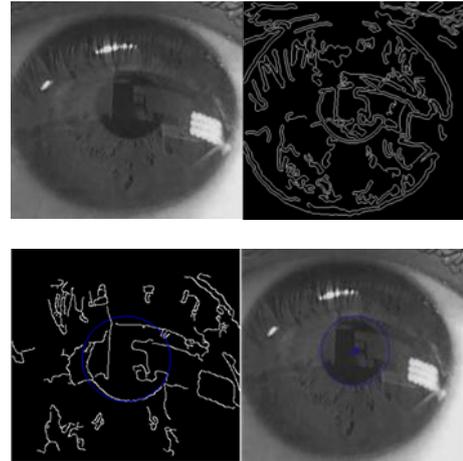


Figure 9. Isolated pupil region (top-left), pupil edge-map (top-right), located pupil through Hough ellipse transform on the edge-map (bottom-left) and detected pupil on the original grayscale image (bottom-right)

The complete eye pattern detected through the proposed methodology is shown in Fig. 10.

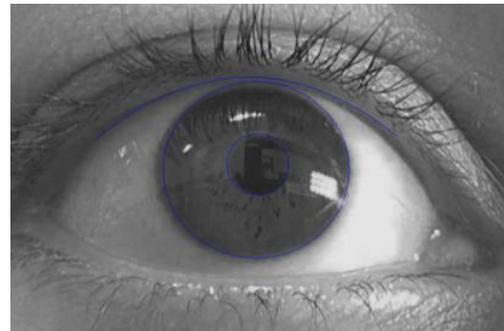


Figure 10: Complete eye pattern detected through the proposed methodology

III. PARALLELIZING HOUGH TRANSFORM

In current pupil detection algorithms, Hough transform is used for both parabola (eyelash) and ellipse (iris and pupil) detection. The most time consuming part of these algorithms is the calculation of Hough voting array for both Hough parabola and ellipse. The voting procedure takes almost 98% of the total processing time. The preliminary experiments indicate that processing 30 sec at 25 fps required approximately 6 hours on a single Pentium III 2.6-GHz system. The amount of time and memory the algorithm takes are computed using MATLAB® built-in tools. As the size of the problem increases, delay in computation of voting array impairs the performance of pupillometry applications.

The first step for parallelization of eye-segmentation algorithm is partitioning that involves distribution of workload among available processing resources. In classical theory of parallel computing, partitioning of a computationally intensive problem can be carried out

either by dividing data (i.e. data parallelism) or by dividing tasks (i.e. task parallelism) among processors. Figure 11 elucidates our parallel approach in which the input data comprising of input edge map is distributed to workers, where most of the computation is carried out. The worker processes gather votes from an accumulator array using a simple data parallel partitioning approach and determine maximum number of votes using an evaluation function. Each worker process only processes a subset of the accumulator array, computes a score and finally winner 2D slice gives the detected ellipse combination.

The implementation of the proposed parallel approach has been carried out using Matlab®. The parallel computing Toolbox is part of Matlab that allows writing parallel codes for multicore and GPU platforms using special directives. To measure the performance of both sequential and parallel implementations for Hough Transform, a number of benchmark tests are performed. The CPU used in these benchmark tests is an Intel® Xeon E5520 with eight cores running at 2.27 GHz. The CPU implementations use parfor to utilize all cores and calculate the Hough array by iterating over all pixels in the binary input image. If a pixel value is one, this pixel value is used in the voting process, otherwise it is discarded. In all implementations, the trigonometric functions are pre-calculated and stored in an array.

The reduced computational runtime is achieved by distributing the voting calculation among 2, 4, 6 and 8 worker processes. The GPU used in this setup is NVIDIA® TESLA T10 running at 1.2 GHz and has 1280 MB of off-chip global memory.

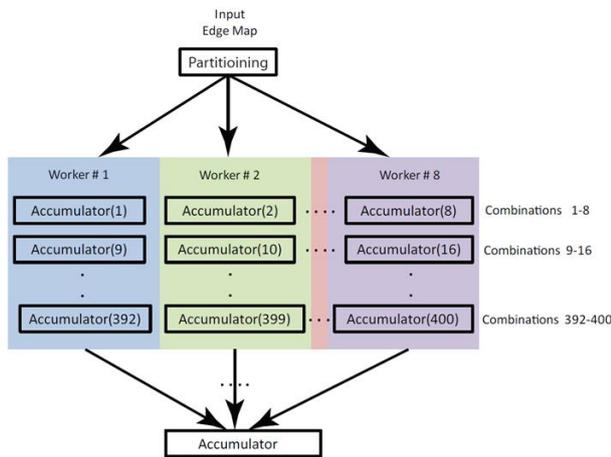


Figure 11. Distribution of workload for voting procedure using data parallelism

IV. RESULTS AND DISCUSSION

In order to measure the efficiency of the pupil localization, the proposed algorithm is tested on CASIA database. Out of the 500 images processed from 250 subjects 436 show correctly segmented pupils, which gives the proposed algorithm 87.2% pupil detection accuracy. Due to the use of infra-red light for

illumination, images in the CASIA iris dataset do not contain specular reflections. The proposed method was tested to remove reflection noises from the noisy input image as well. Additionally, other noises can be accurately detected and removed by the proposed method. Fig. 12 shows the segmentation results of a few samples in the CASIA iris dataset. The detected pupil is shown by blue ellipse superimposed on the images.

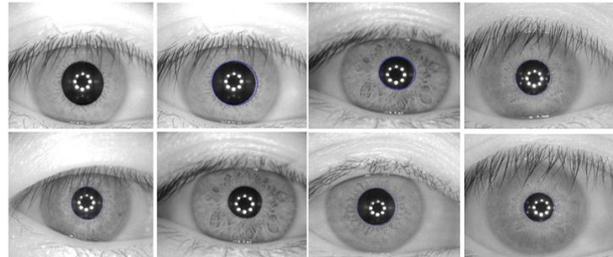


Figure 12. Segmentation results of some samples in the CASIA iris dataset

The accuracy of the proposed algorithm is compared with the results reported by Fernandez [32] and Libor Masek [33]. Fernandez used connected component analysis to detect irregular pupil boundaries. Masek used circular Hough transform to detect the pupil boundary. The proposed method has higher detection rate compared to existing techniques (Table I).

TABLE I. COMPARISON OF SEGMENTATION ACCURACY

	Method	Success Detection Rate
Fernandez	Connected component analysis	82.3 %
Masek	Circular Hough transform	83.92 %
Proposed	Parabolic and Elliptic Hough transform	87.2 %

The parallelization results discussed ahead are clustered in three segments. The first part discusses the advantages of parallelizing only parabolic Hough Transform. The next part discusses the effect of parallelizing elliptic Hough Transform alone. Finally, the speedup analysis by parallel implementation of the whole algorithm is performed on a single eye image.

A. Parallel Parabolic Hough Transform (PPHT)

The sequential results of Parabolic Hough Transform are compared with the Parallel Parabolic Hough Transform (PPHT) in Table II and plotted in Fig. 13. The number of parallel nodes considered in the study are two, four six and eight. It can be observed that using eight cores of a CPU for PPHT resulted in a speedup by almost a factor of two without any performance loss in terms of accuracy. Moreover, the CPU runtime almost follows a linear reduction with the increase in processing power.

TABLE II. EXECUTION TIMES AND SPEEDUPS FOR CPU, CPU (8-CORES) AND GPU FOR PARABOLIC HOUGH TRANSFORM

	Serial	Number of cores			
		2	4	6	8
CPU Runtime (s)	84.2331	70.8776	56.0168	45.0513	42.1716
CPU Speedup (x)	1x	1.1884x	1.5037x	1.8697x	1.9974x

In the next step, GPU accelerators are added into the computational realm to exploit the full computing power of the standalone machine. The GPU implementation is done to take a step further. The total number of cores in the GPU system plays an important role in determining the performance of the program. As explained before, Hough array computation is the identified bottleneck in achieving performance.

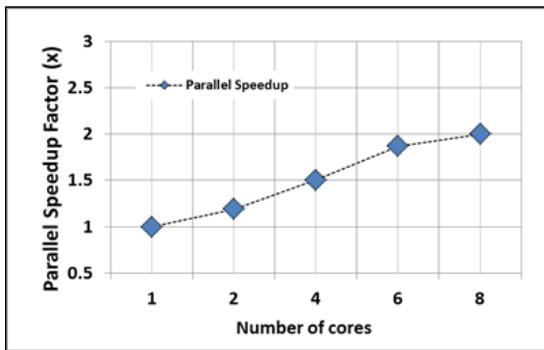
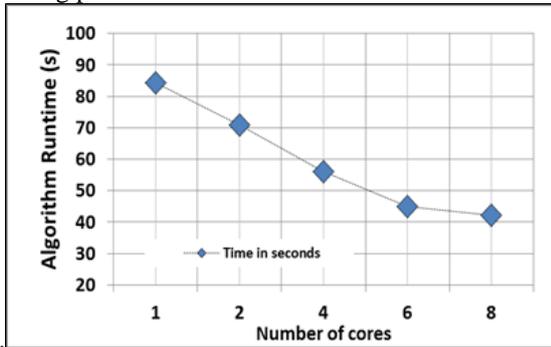


Figure 13: The algorithm runtime (top) and speed up factor (bottom) against the number of processors for PPHT

Accumulator calculation for parabolic Hough transform is performed via GPU as well. Table 3 gives the execution times and speedups achieved for the parabolic Hough implementation for sequential, multicore (8-core), and GPU programs.

TABLE III. EXECUTION TIMES AND SPEEDUPS FOR CPU, CPU (8-CORES) AND GPU FOR PARABOLIC HOUGH TRANSFORM

CPU Runtime	CPU Speedup	CPU Runtime 8-core	CPU Speedup 8-cores	GPU Runtime	GPU Speedup
84.2331	1x	42.1716	1.9974x	32.4628	2.5948x

It should be noted that in case of parabolic Hough transform the problem size is relatively small i.e. 96 possible combinations are to be checked. The speedup achieved for the parallelized version is 2.59 times more than the sequential program as shown in Fig. 14.

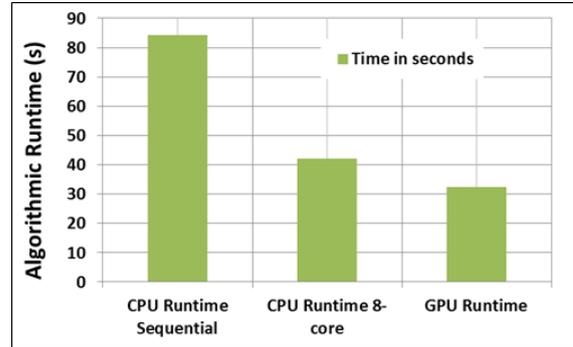


Figure 14. GPU runtime for Parabolic Hough Transform on Tesla T10 processor

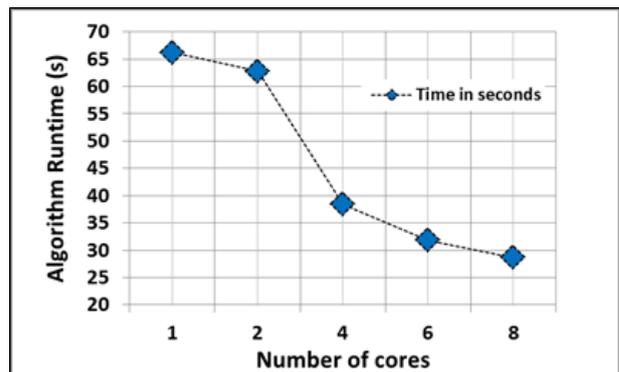
B. Parallel Elliptic Hough Transform (PEHT)

In Parallel Elliptic Hough Transform (PEHT), all eight processors on the machine are used to calculate 400 threads for the ellipse combinations to detect the pupil. In case of elliptic Hough implementation, the problem size is large hence a significant speedup is achieved through parallelization. Table 4 shows the execution time and speedup results achieved for elliptic Hough array computation.

TABLE IV. EXECUTION TIMES AND SPEEDUPS FOR CPU, CPU (8-CORES) AND GPU FOR ELLIPTIC HOUGH TRANSFORM

	Serial	Number of cores			
		2	4	6	8
CPU Runtime (s)	66.1799	62.7804	38.4122	31.8428	28.6614
CPU Speedup (x)	1x	1.0541x	1.7229x	2.0783x	2.309x

Fig. 15 shows that an overall increase of 2.31x speedup is achieved.



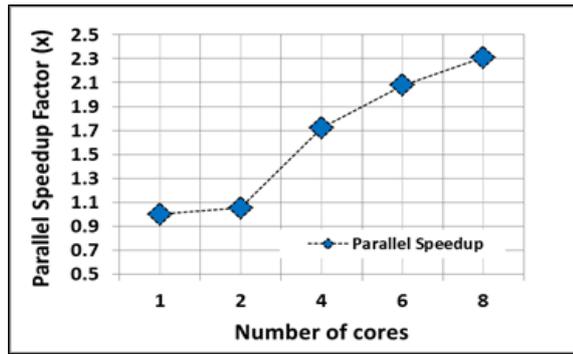


Figure 15: The algorithm runtime (left) and speed up factor (right) against the number of processors for PEHT

TABLE V. EXECUTION TIMES AND SPEEDUPS FOR CPU, CPU (8-CORES) AND GPU FOR ELLIPTIC HOUGH TRANSFORM

CPU Runtime (s)	CPU Speedup	CPU Runtime 8-core (s)	CPU Speedup 8-cores	GPU Runtime (s)	GPU Speedup
66.1799	1x	28.6614	2.309x	26.5183	2.4956x

The GPU implementation runs 2.49 times faster as compared to the sequential CPU implementation (Table 5). However, when compared to the multicore program the speedup reaches only 1.08 times, which is almost the same as multicore implementation as shown in Fig. 16.

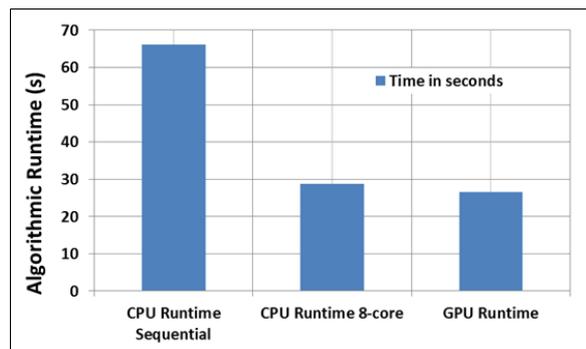


Figure 16: GPU runtime for Elliptic Hough Transform as compared to sequential and CPU-8 core processing

C. Parallelized Pupil Detection on a Single Eye Frame

The effect of parallelizing elliptic and parabolic Hough codes on the whole pupil detection program is given for a single frame in Table 6. The CPU runtimes and associated speedups with varying processors are plotted in Fig. 17. The multicore implementation of the proposed program reduces the runtime of the pupil detection on a single eye frame from 151.95 s to 81.44 s. Moreover, the speedup achieved is 1.86 times higher than that of the sequential implementation.

The effect of parallelizing elliptic and parabolic Hough codes on the whole pupil detection program is given for a single frame in Table 6. The CPU runtimes and associated speedups with varying processors are plotted in Fig. 17. The multicore implementation of the proposed program

reduces the runtime of the pupil detection on a single eye frame from 151.95 s to 81.44 s. Moreover, the speedup achieved is 1.86 times higher than that of the sequential implementation.

TABLE VI. EXECUTION TIMES AND SPEEDUPS ACHIEVED BY THE PROPOSED PUPIL DETECTION METHODOLOGY APPLIED ON A SINGLE EYE IMAGE

	Serial	Number of cores			
		2	4	6	8
CPU Runtime (s)	151.9486	132.8531	90.0382	83.5424	81.4382
CPU Speedup (x)	1x	1.1437x	1.6876x	1.8188x	1.8658x

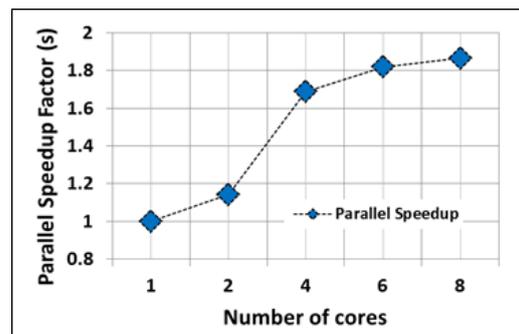
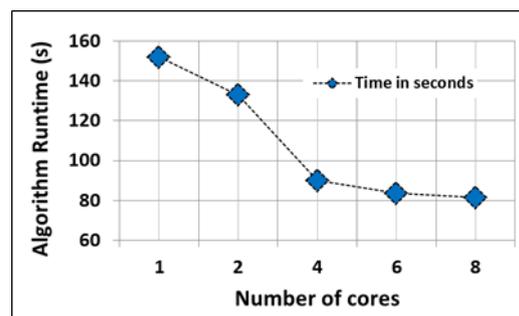


Figure 17. The algorithm runtime (left) and speed up factor (right) against the number of processors for the complete parallel framework

The GPU-based implementation made the algorithm performance even better. Table VII shows a comparison of runtimes and speedups achieved with sequential, multicore and GPU-based executions for the proposed program. The multi core implementation gives us a 1.86x increase in speed whereas the GPU-based code provides a 3.56x increase in the speedup of the pupil detection program.

TABLE VII. EXECUTION TIMES AND SPEEDUPS ACHIEVED FOR CPU, CPU-8 CORES AND GPU-BASED IMPLEMENTATION FOR A SINGLE EYE FRAME ON NVIDIA® TESLA T10 GPU

CPU Runtime (s)	CPU Speedup (x)	CPU Runtime 8-core (s)	CPU Speedup 8-cores (x)	GPU Runtime (s)	GPU Speedup (x)
151.9486	1x	81.4382	1.8658x	42.634876	3.5639x

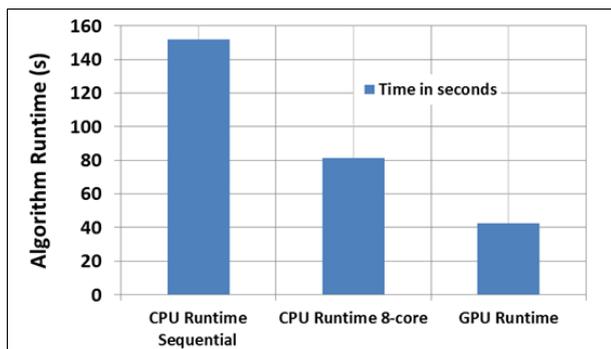


Figure 18. A comparison of CPU, CPU-8 core and GPU runtimes for proposed algorithm for a single eye frame on NVIDIA® Tesla T10 processor

V. CONCLUSIONS

In this paper, a parallel approach is proposed to increase the performance of segmentation algorithm in pupillometric applications. The automatic segmentation algorithm localizes the pupil region from an eye image and isolates eyelid and eyelash areas. The automatic segmentation is achieved through the use of Parabolic and Elliptic Hough Transform for localizing the iris and pupil regions respectively. The Parabolic Hough Transform is used for detecting occluding eyelash region. The pupil detection method according to this approach detects pupil boundary with almost 87.2% accuracy which is approximately 4-5% higher than the existing approaches. For computational efficiency, the parallel framework is developed and analyzed for the standalone machine. Specifically, the performance of multi-core CPU computations with and without GPU accelerators is compared. It is identified that the primary computational cost is of Hough Transform. Therefore, parabolic and elliptic Hough transforms are parallelized separately as well as combined for the detection and measurement of pupil dilation over time. For the proposed pupil detection and area calculation method applied on a single input eye image, a maximum speed up of 1.86x and 3.56x on multi-core CPU framework and GPU is achieved respectively. The proposed approach can be used to develop efficient future pupillometry applications for multicore and GPU based HPC platforms.

ACKNOWLEDGMENT

The work was supported by National Internship Program of Ministry of Federal Education & Professional Training, Government of Pakistan.

REFERENCES

- [1] J. L. Andreassi, *Psychophysiology, in Human Behavior and Physiological Response*, Erlbaum Associates, Incorporated, Lawrence, 2000.
- [2] E. H. Hess and J. M. Polt, *Pupil Size as Related to Interest Value of Visual Stimuli*, Science, 1960.
- [3] E. H. Hess and J. M. Polt., *Pupil Size in Relation to Mental Activity during Simple Problem-solving*, Science, 1964.
- [4] J. Beatty and B. L. Wagoner, *Pupillometric Signs of Brain Activation Vary with Level of Cognitive Processing*, Science, 1978, 199(4334), pp. 1216-1218.
- [5] J. Qiyuan, et al., "The pupil and stimulus probability," *Psychophysiology*, vol. 22, no. 5, pp. 530-534, 1985.
- [6] Klingner, J. A. K., Rakshit and Hanrahan, Pat. "Measuring the task-evoked pupillary response with a remote eye tracker," in *ETRA '08: Proc. the 2008 Symposium on Eye Tracking Research & Applications*, 2008. ACM.
- [7] J. Klingner, Tversky, Barbara. Hanrahan, Pat., "Effects of visual and verbal presentation on cognitive load in vigilance, memory, and arithmetic tasks," *Society for Psychophysiological Research*, 2010.
- [8] J. Zhai and B. Armando, "Stress detection in computer users based on digital signal processing of noninvasive physiological variables," in *Proc. Engineering in Medicine and Biology Society, EMBS, 28th Annual International Conference of the IEEE*, 2006.
- [9] Bayer, M., Sommer, Werner., Schacht, Annekathrin., "Emotional words impact the mind but not the body: Evidence from pupillary responses," *Society for Psychophysiological Research*, 2011.
- [10] M. A. S. Pomplun, Sindhura., "Pupil dilation as an Indicator of cognitive workload in human-computer interaction," in *Proc. the International Conference on HCI*, 2003.
- [11] B. Laeng, S. Sirois, and G. Gredebäck, "Pupillometry A window to the preconscious? *Perspectives on Psychological Science*, 2012.
- [12] G. Gredebäck and A. Melinder., "Infants' understanding of everyday social interactions: A dual process account," *Cognition*, 2010.
- [13] Gredebäck, G., Linn Fikke, and Annika Melinder., "The development of joint visual attention: a longitudinal study of gaze following during interactions with mothers and strangers," *Developmental Science*, 2010.
- [14] I. Jackson and S. Sirois, "Infant cognition: Going full factorial with pupil dilation," *Developmental Science*, 2009.
- [15] F. Shic, C. Katarzyna, B. Jessica., Scassellati, Bradshaw., Autism, Eye-Tracking, Entropy, in 7th IEEE International Conference on Development and Learning, ICDL, 2008.
- [16] A. Iriki, M. Tanaka, and Y. Iwamura., "Attention-induced neuronal activity in the monkey somatosensory cortex revealed by pupillometrics," *Neuroscience Research*, 1996.
- [17] Palinko, O. A. K., Andrew, "Exploring the influence of light and cognitive load on pupil diameter in driving simulator studies," in *Proc. the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2011.
- [18] A. L. Kun, Palinko, Oskar., Medenica, Zeljko., Heeman, Peter A., "On the feasibility of using pupil diameter to estimate cognitive load changes for in-vehicle spoken dialogues," in *Proc. Interspeech*, 2013.
- [19] D. M. Bowler, S. B. Gaigg, and J. M. Gardiner, "Subjective organisation in the free recall learning of adults with Asperger's syndrome," *Journal of Autism and Developmental Disorders*, 2008.
- [20] L. Kuchinke, D. Schneider, S. A. Kotz, and A. M. Jacobs, "Spontaneous but not explicit processing of positive sentences impaired in Asperger's syndrome: Pupillometric evidence," *Neuropsychologia*, 2011.
- [21] C. Daluwatte, J. H. Miles, and G. Yao, "Simultaneously measured pupillary light reflex and heart rate variability in healthy children," *Physiological Measurement*, 2012.
- [22] C. Daluwatte, J. H. Miles, S. E. Christ, D. Q. Beversdorf, T. N. Takahashi, and G. Yao, "Atypical pupillary light reflex and heart rate variability in children with autism spectrum disorder," *Journal of Autism and Developmental Disorders*, 2013.
- [23] A. Parnandi and R. Gutierrez-Osuna., "Contactless measurement of heart rate variability from pupillary fluctuations," in *Proc. IEEE, Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, 2013.
- [24] X. Jianga, Z. Bin, T. Geoffrey, A. M. Stella, "Pupil response to precision in surgical task execution studies in health technology and informatics," 2012.
- [25] Y. K. Chen, W. L. Li, J. G. Li, and T. Wang, "Novel parallel hough transform on multi-core processors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2008.
- [26] Wu, S. A. L., Xiangjiao, "Parallelization research of circle detection based on Hough transform," *International Journal of Computer Science Issues*, IJCSI, 2012.

- [27] G. J. V. D. Braak, N. Cedric, M. Bart, and C. Henk, "Fast hough transform on GPUs: Exploration of algorithm trade-offs," in *Proc. Advances Concepts for Intelligent Vision Systems*, 2011.
- [28] G. Sharma and M. Jos, "MATLAB®: A language for parallel computing," *International Journal of Parallel Programming*, 2009.
- [29] P. T. Tan, *Institute of Automation*, Chinese Academy of Sciences (CASIA) 2004, Center for Biometrics and Security Research (CBSR), National Laboratory of Pattern Recognition (NLPR).
- [30] J. F. Canny, "Finding edges and lines in images," *Massachusetts Inst. of Tech. Report*, 1983. 1.
- [31] C. H. P. V., *Method and Means for Recognizing Complex Patterns*, 1962, Google Patents.
- [32] J. J. Fernandez and A. Mathew, "Irregular pupil localization using connected component analysis," in *Proc. Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, 2013, IEEE.
- [33] L. Masek, "Recognition of human iris patterns for biometric identification," 2003, University of Western Australia.



Ms. Amnah Nasim holds bachelor in electronics engineering and masters in computational science and engineering, both from National University of Sciences and Technology, Islamabad, Pakistan. She opted this area of research for her Master's thesis under the supervision of Dr. Adnan Maqsood at Research Centre for Modeling & Simulation (RCMS). She is currently pursuing PhD studies from Pakistan Institute

of Engineering & Applied Sciences (PIEAS), Islamabad. Her current research interests are associated with Computer Vision, Machine Learning and Parallel Computing.



Dr. Adnan Maqsood is working as Assistant Professor at National University of Sciences and Technology (NUST), Pakistan, since 2012. He received his Bachelor's degree in Aerospace Engineering from NUST, Pakistan in 2005 and PhD from Nanyang Technological University (NTU), Singapore in 2012. Currently, he is engaged in graduate teaching and research at Research Centre for Modeling & Simulation and heading a

Mechanics & Control (AFMC). He has done significant research work and published several top quality international conferences and journal papers. He has been often invited as a reviewer for a number of conferences, journals and book reviews. His current research interests are associated with Applied Aerodynamics, Flight Dynamics and Control, Unmanned Air Vehicle (UAV) Systems, Human Factors Engineering and Applied Ergonomics. This work is targeted towards development of physiological measures for stress assessment in complex workspaces such as Ground Control Stations for Unmanned Systems.



Mr. Tariq Saeed is serving as Assistant professor at Research Center of Modeling and Simulation (RCMS) Since 2011. He completed MS Information Security from NUST in 2008. He served as Lecturer at RCMS from 2008-2011. He has also served as Incharge Supercomputing Research and Education Center (ScREC) from April 2009 to October 2012.