Input Interface Using Eyes from Webcam Stream

Sanjay Dahiya[1] Bhartendu Kumar [2]

[1,2] *Department of Computer Science & Engineering, Chaudhary Devi Lal State Institute of Engineering & Technology,* Panniwala –Mota, (Sirsa),Haryana, INDIA-125077

Email: - sanjaydahiyakkr@gmail.com[1], bhartendukumar1998@gmail.com[2]

Abstract: ***Real time detection and tracking of eyes is now possible due to the culmination of research and experiments of many researchers over the year and also the advancement in computational capacity of modern computers. Thus, it opens up a new opportunity for all of us to consider "Natural Human-Computer Interface" as a reality and treat eyes as input interface to get information from the users. Eyes have been experimentally and intuitively thought of conveying a large information as well as presenting opportunity to have seamless and natural interaction with the system. This paper proposes a framework for the same using only webcam. As most computing systems have a decent webcam these days, we propose here a graphical feature centered method. Further, after the localization of the user eyes, we can extract geometrically the features of the eyeball to estimate gaze direction, blinks, attention level, open and closed eyes. The two defining achievements of this paper are: firstly, till now, no practical framework has been developed to get user interactivity through webcam, and secondly, all the eye tracking commercial products in the market use external equipment like head-mounts, glasses, etc. The proposed functionality achieved by the paper includes: video play-pause by verifying user gaze towards screen or not, natural scrolling while reading text based on the pupil conditions, long blink screenshots, etc. The application areas include, driver fatigue intelligent automobiles, intelligent camera adjustment, face recognition, natural HCI through mobile devices. The framework can run utilizing 5% of CPU resources and having an accuracy of 98%.***

***Keywords*:** Eye tracking, Eye detection, Natural HCI, Real-time eye localization, Image normalization, Harr-cascade, Optical flow, Geometric features, Gaze, Pupil.

I. Introduction

The paper is divided into four main steps of: (1) detecting eyes as fast as possible by "harr-cascade classifier" (which is the most popular and most widely used algorithm to detect objects in real time), (2) and after detection we use "Lucas-Kanade optical flow" to have track of eyes, (3) then we extract the eye pupil features based on geometric-analysis (simple and fast) and at last (4) we stimulate desired input that the eye is transferring into the operating system input stream. So, in all these 4 steps read the eye behavior and the information abstracted in it to transfer input to the input stream and interact with applications and programs. This transfer of information to computer from eyes has to be in real time and the computations performed have to be bare minimum. The proposed system for the step **one** uses Harr-Cascade. Anytime as the frame from the video stream is chosen for evaluation, it is *preprocessed* into a new image representation called the "Integral Image" [1] which makes all the further computations fast. Then the detection step is modified AdaBoost *learning algorithm*. It is a degenerate decision tree like classifier that starts on the pre-processed frame image and in successive steps finds interesting areas in the image to focus on [2]. Each iteration of classification represents some features that it has learned from training examples and can go to 38 steps, each one more aggressive than previous. This is because in training examples the successive stage classifier works on the images that the previous stage classifiers passed and it has to filter among them to the next level. So, after training the AdaBoost algorithm stores in it feature classifiers (many successive stages) and each classifier looking for some feature that it has learned. Increasingly more *classifiers* are "cascaded" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The step two is tracking which is done by continuous image analysis. In this step we at fixed interval of time analyze the live video to get the state of the eyes. The present state is then compared with the previous states and this kind of adaptive algorithm that also mainly focuses on only the important aspects of the image is very computationally efficient. Then in third step, we use geometric detailed analysis on the detected eye region to get full information of gaze, open or close state, etc. We can get data for detailed analysis for drowsiness, attention level, emotions displayed through eyes. These inputs may be used for some advanced interaction system. The last step, fourth, is the bridge between the extraction of exact state of eyes in the video frame and informing the computer system by sending appropriate input signals in the stream. The least computationally intense and quickest step is this which could be a simple and fast program code that maps the eye states to accurate inputs for simple program requirements. Like having gaze not in direction of the screen may introduce command for pause if video is playing. This approach maps continuous input class to binary or multi class outputs using a "parameter" value for mapping. For example, pupil size in the detailed analysis is if less than the "parameter", then eye is considered closed, and thus a continuous input class is mapped to discrete output cases. These algorithms have the potential to revolutionize the future computer-man interaction. Eye and lips are two of the most interactive organs to communicate many things and if we successfully make a bridge for computer to interpret inputs directly, then many new and unheard-of areas will be unlocked.

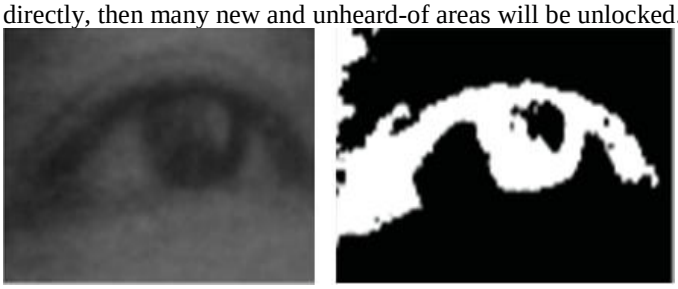

Fig.-1 Detailed geometric analysis to get gaze and pupil size [3]

II. Literature Review

There are two main types of analysis in the eye detection field. One is the INFRARED-OCULOGRAPHY which includes equipment to be worn by user like head gears, glasses, special fixed cameras, etc. These are the most accurate methodologies and are the most widely implemented in commercial products. In the biometric or other face detection equipment, the IR light is used. The advantages include: (1) approximately 100% accuracy and (2) no major computation so that supported by embedded electronic chips with low computation power. The disadvantages include: (1) some harmful radiations in IR spectrum that if used for prolonged durations could damage the eyes, (2) restricted movement of the head and (3) uncomfortable head gears. The other class of algorithms are VIDEO-OCULOGRAPHY, it is further classified into graphical analysis and texture analysis. The color-texture analysis means the color scheme and the texture map of the image is analyzed for eyes. Thilak et al. [8] proposed an algorithm which by three levels detects eyes. First, they localized eye candidates by simple thresholding on HSV color space and normalized RGB color space sequentially. The geometric features such as eyeball is round and darker than surrounding are widely used to detect eyes. Lin and Yang [7] propose a Dark Pixel Filter (DPF) applied in the upper half of the face region obtained by Face Circle Filtering (FCF). Then after dark pixel filtering, eye verification and eye localization, accurate eye locations are determined.

A notion similar to the cascade appears in the face detection system described by Rowley et al. in which two detection networks are used [12]. Rowley et al. used a faster yet less accurate network to prescreen the image in order to find candidate regions for a slower more accurate network. Though it is difficult to determine exactly, it appears that Rowley et al.'s two network face system is the fastest existing face detector. The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of Amit and Geman [1]. Unlike techniques which use a fixed detector, Amit and Geman propose an alternative point of view where unusual co-occurrences of simple image features are used to trigger the evaluation of a more complex detection process. Fleuret and Geman have presented a face detection technique which relies on a "chain" of tests in or- der to signify the presence of a face at a particular scale and location [10]. The image properties measured by Fleuret and Geman, disjunctions of fine scale edges, are quite different than rectangle features which are simple, exist at all scales, and are somewhat interpretable.

Further an iterative image registration technique with an application to Stereo Vision has been proposed by of the full framework, Lucas and Kanade [5]. This technique is faster because it examines far fewer potential matches between the images.

III Proposed Interface System

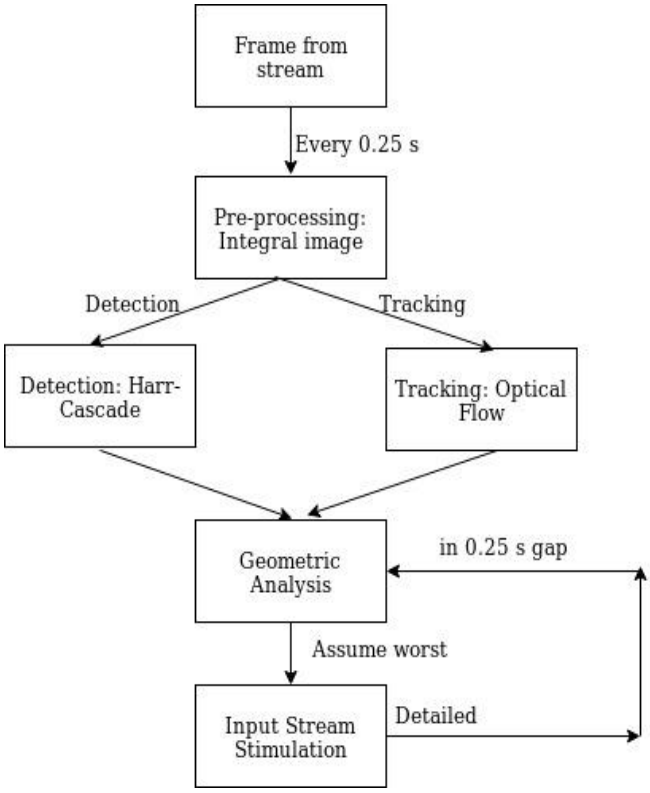

Fig- 2. Block Diagram of Proposed Interface System

(A) Analyzing video stream 4 times in a second:

The implementation includes the experimentally verified assumption that the time frame of ¼ of a second is the tradeoff for continuously tracking the eye in the video stream. The only algorithm that is sensitive to the time difference between two states under examination is the tracking algorithm "optical flow", which can work on this parameter and for failure of the "Optimal Flow Algorithm" due to abruptly large motion, we move forward by verifying it instantly by "Face-Detection. In this framework of eye-detection for verification in the case of swift motion, we could use eye-detection in the frame to localize the eye after the rash motion. But using face-detection at this step has several benefits: (1) after failure of "Optimal-Flow", we move forward assuming eye gaze is not directed to screen and so the exact conclusion should be as fast as possible, and face detection is faster than eye, (2) in case of large abrupt movement (as optical-flow fails) there is a great probability that there is no face in the video frame and (3) if a face is successfully detected, eye position prediction is too fast combining geometric features of face and face template from previous frames. The time duration between two successive frame analysis could be utilized for further computations on detected eyes by graphical methods for detailed status of the gaze and pupil if approximate results is sketchy or is invalidated by successive frames.

(B) Harr feature-based Cascade Classifiers

Fig-3 shows some examples of Haar-Like features, which are represented as the sum of pixel intensities in black region minus that in white region. The algorithm first trains on a large number of positive (which contain faces) and negative (which do not contain faces) images, to select some relatively effective Haar-Like features. For each feature, there is a corresponding weak classifier that outputs 1 for face, and 0 for nonface. Its performance is weak, that is, with high false positive or false negative rates. Thus, [1] proposes a boosting algorithm that boosts multiple weak classifiers into a strong classifier, and uses a sliding window to search the entire query image. At each stage, if the detection result of a weak classifier is below a certain threshold, the window is discarded immediately and will never be considered later. After passing all the stages, the remaining windows are considered to be face regions. The cascade classifier is shown in Figure 4.

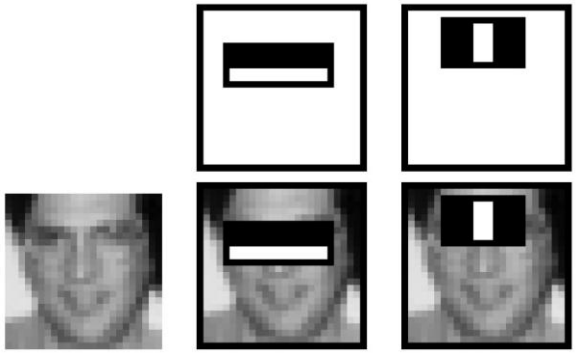

Fig-3: Harr-Like Features [4]

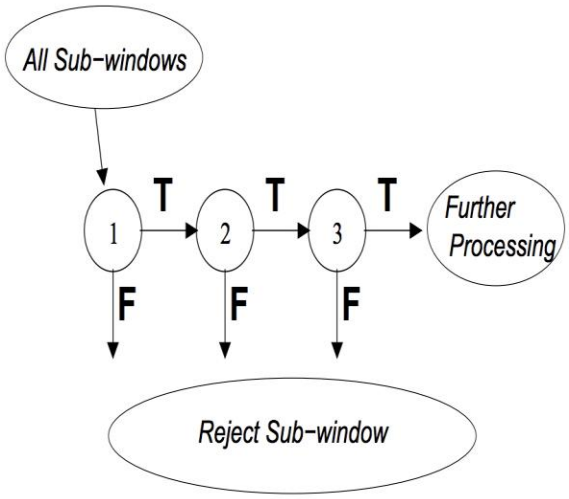

Fig- 4: Cascade Classifier [4]

(C) Optical Flow based Tracking

Optical flow is defined as spatio-temporal image brightness variations. To track an eye across a sequence of frames, we use Lucas-Kanade optical flow algorithm[5] to estimate the motion of extracted facial features between adjacent frames in the sequence. This technique is adaptive and speed increases if "integral image" are used.

The algorithm has three assumptions:

1. Small motion. Points only have a small movement between two neighboring frames.
2. Brightness constancy. Projection of the same point in every frame has almost the same brightness.
3. Spatial coherence. Points should have almost the same movements as their neighbors.

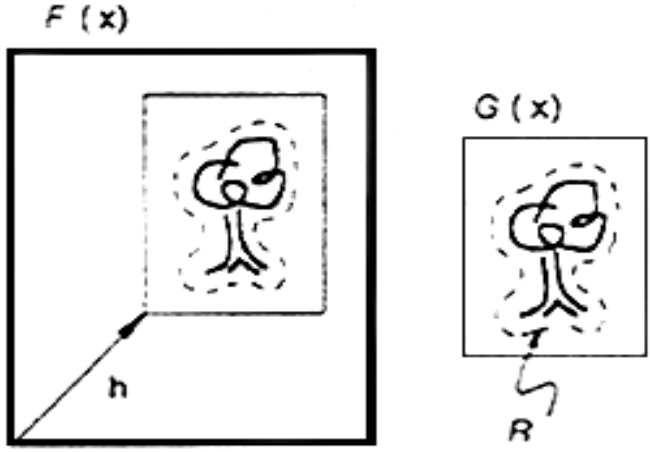

Fig.-5: The image registration problem [5]

Here in Fig.-5, as indicated by the Region of Interest (ROI): R, the two images are compared based on only this significant region, which makes this algorithm win over others as it limit the computation within the area "R" and compare the spatial intensity gradient of the images.

(D) Detailed geometrical analysis of pupil

The detailed analysis of pupil is done graphically and thus the only computations are of distances and shapes for knowing the exact state of the pupil for gaze direction and pupil size. Using graphical calculations focus on the black spot of pupil surrounded by the white cornea. Some fast algorithms of computer graphics like flood fill algorithm (or boundary fill), and also as the computation at this stage is only black and white, the boundary of the pupil could be constructed in the detected eye region. The pupil boundary could then be analyzed for its shape and size. The shape of the pupil is given by the averaged maximum and minimum points in direction parallel to the line connecting both the eyes and the averaged maximum and minimum points in the direction perpendicular to it. These calculations could give the estimate of the pupil shape. The size could be calculated as the ratio of the pupil averaged area to the eye averaged area. The ratio if less than the threshold would

signal eyes closed and if ratio is more than threshold, indicates eyes open.

(E) Input Stream according to detected state of eyes

There are many input APIs particularly according to the platform and they could very easily mimic the input stream that is controlled by eye. The result of the approximate eye state (or if detailed graphical calculations) is send to the operating system or the applications working on, this makes webcam here as an input device.

IV Conclusion

The proposed system works in real time due to ¼ of a second continuous checking, and experiments show that major portion of the algorithm just verifies that the position of eye is nearly static, thus saves a large number of computations. And when computations are needed, we do minimum to get accurate **results,** like Harr-Cascade which takes constant time for each classification and Optical flow which takes as minimum computation as possible in related images. Further "integral-image" preprocessing before any algorithm makes computations fast. The geometric calculations are the only one to make detailed analysis but the trade-off we adopted for accuracy in gaze detection and time of computation is to have the computation only approximate, and move further by quick evaluation which should be supported by successive frames, it worked well in real systems as any wrong approximation was detected by next frames and user is not bothered by 0.25 or 0.5 second lag in correction. So, averaging over most of the personal computer sin the market, the system would take only 1/10 of the CPU resource in worst case, but could be easily optimized below it.

Reference

[1] Viola, P., Jones, M., "Rapid Object Detection using a Boosted Cascade of Simple Features," in: CVPR, pp. 511–518. IEEE Computer Society (2001)

[2] Y. Li, X. Xu, N. Mu and L. Chen, "Eye-Gaze Tracking System By Haar Cascade Classifier," in: 11th Conference on Industrial Electronics and Applications (ICIEA), IEEE, Hefei, China, 5-7 June 2016.

[3] Hanif Fermanda Putra and Kohichi Ogata, "Development of Eye-Gaze Interface System and Its Application to Virtual Reality Controller," in: 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)

[4] Qi Cao and Ruishan Liu, "Real-Time Face Tracking and Replacement"

[5] Bruce D. Lucas Takeo Kanade , "An Iterative Image Registration Technique with an Application to Stereo Vision," in: Proceedings of Imaging Understanding Workshop, pp. 121-130 (1981).

[6] A. Haro, M. Flicker and 1. Essa, "Detection and tracking eyes by using their physiological properties. dynamics and appearance," in: Proceedings o/lEEE CVPR 2002.

[7] D.T. Lin and C.M. Yang, "Real-time Eye Detection Using Face-circle Fitting and Dark-pixel Filtering," in: IEEE International Conference on Multimedia and Expo, vol. 2, pp. 1167–1170, 2004.

[8] R. Thilak, S. Kumar Raja, and A.G. Ramakrishnan, "Eye detection using color cues and projection functions," in: Proceedings 2002 International Conference on Image Processing ICIP, volume 3, Rochester, New York, USA, 2002.

[9] Barnea, Daniel I. and Silverman, Harvey F, "A Class of Algorithms for Fast Digital Image Registration," in: IEEE Transactions on Computers C-21.2 (February 1972), 179-186.

[10] F. Fleuret and D. Geman, "Coarse-to-fine face detection," in: Int. J. Computer Vision, 2001.

[11] Y. Amit, D. Geman, and K. Wilder, "Joint induction of shape features and tree classifiers," in: 1997.

[12] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," in: IEEE Patt. Anal. Mach. Intell., volume 20, pages 22–38, 1998.