Real Time Eye-Tracking using Web Camera

A Review Paper on an Ongoing Project

Riddhi Chavda, Amit Doshi, Madhura Barve
Student, Computer Engineering Dept.
D.J. Sanghvi College of Engineering
Mumbai, India

Ruhina Karani
Assistant Professor, Computer Engineering Dept.
D.J. Sanghvi College of Engineering
Mumbai, India

***Abstract*— This paper describes an ongoing project that has the aim to develop a low cost application to replace a computer mouse for people with physical impairments. The application is based on an eye tracking algorithm: Fabian Timm implemented using a conventional web camera. The system to be designed aims to detect the user's eye movements. As an advancement to the research conducted, we aim to implement detecting blinks and analyse the nature and timing of blinks, which would in turn be used as an input to the computer as a mouse click. An initialization of the system should occur automatically after the user blinks involuntarily in the initial few seconds of use, the eye should be tracked using correlation with the online template. Also if the user's position changes to a large extent or very fast, then the system is supposed to reinitialize.**

Keywords— Human Computer Interaction (HCI); eye-tracking; Fabian Timm algorithm; pattern recognition.

I. INTRODUCTION

Many devices which exploit the remained abilities of people, who are physically challenged to operate computers, have been invented in the recent years. Speaking of which, computer vision has made considerable research under the category of object-tracking. Thus, now the requirement of direct contact with the cornea has been eliminated to replace it with more non-invasive techniques. The Tobii eye-tracker is an example of this. However, the high-cost factor has hampered its popularity in terms of usage. Eye-tracking is in fact a very challenging task due to aspects like significant pupil reflectivity and different shapes and openness of the eye. Calibration issues and hardware setup are also very critical for system design. In recent years, economical web cameras like those of Logitech are easily available, which facilitates incorporation of these systems on a larger scale, thus annulling the requirement of rather expensive equipment and high end video cameras. Considering the software part, we have the tracking algorithm whose aim is to locate and track the user's eye in consecutive frames of the video stream. The intended input is the region of interest, where the search procedure takes place, i.e. only the eye image.

The following considerations need to be kept in mind while implementation:

- Controlling the mouse cursor
- Minimal hardware complexity
- Adaptability with different users
- Minimal cost

II. REVIEW OF LITERATURE

Different techniques for eye-tracking, its detection, movement, etc. have been used for different implementations.

One of the papers provides a robust reimplementation of the system described by Grauman *et al.* implemented in the BlinkLink blink detection system that is able to run in real time at 30 frames per second on readily available and affordable webcams. [5]

The other experiments involve the use of neural networks for eye-detection. The intensity of each pixel in the image of the eye was used as an input to the neural network. The network's two outputs corresponded to the X and Y locations of the user's gaze on the screen. Assuming linear separability would not pose a problem, we employed a feedforward, two-layer neural network. [7]

The prototype of yet another thesis is subdivided into two parts—hardware and software implementation. Eye-GUIDE focuses on video-based gaze tracking, which consist Hannah camera that records the eye of the user and a computer. The webcam can be used with or without infrared (IR) illumination. For the software implementation the proponents used three different softwares for eye gazing, eye clicking and messaging system. The eye gazing software extracts eye features like pupil and iris centre from the image captured by the web camera. For clicking purposes, the Eye GUIDE Clicker software was introduced. This tool allows user to click templates, keyboard letters and supporting buttons on the Eye GUIDE Messenger provided by this study by hovering the cursor several seconds over the desired button or icon. Finally, software for this messaging system allows the user to select from the provided templates, what the user wanted to say. [1]

One of the research involves the Houghman circle detection algorithm for eye-tracking. This basic algorithm processes the input video frames from the camera to detect the cornea. Then this position is compared to the centre point calibrated initially using a square grid on which the algorithm is applied. This calculates the angle and the speed at which the mouse should move [4].

Based on the evaluation of the above techniques, along with the other evaluations, we have decided to setup using a simple web camera and implement the Fabian Timm algorithm, the details of which are mentioned in the following section.

III. METHODOLOGY

A. Set-up

1) Hardware set up

a) An economical web camera

Currently used cameras for video chat applications support high-definition (HD) images with high resolutions up to 1920x1080 pixels. Smartphones and tablets now have an in-built camera which faces the user, and have gained considerable popularity in the recent years. They can provide the necessary baseline for our proposed system. This baseline is defined by resolutions of 1280x720 (iPad FaceTime camera resolution) and 640x480 (the common VGA cameras resolution) [5]. We intend to consider all of these and also the differences in the camera placements and camera qualities while finalizing for one model. In order to find an apposite camera that meets the requirements such as good resolution, frame rate, quality of image, etc., we have tested the following cameras: Microsoft HD-3000, Logitech (c270, c615), and other cheaper models of Trust, NIUM, and Hercules. As expected, it was observed that the expensive models give good HD quality images available at acceptable frame rates (10 fps). After a compromise to match our requirements, finally we have decided to proceed with Logitech c270 HD.

Fig 1: Logitech c270 HD.

b) Display

The display is obviously a crucial component for this implementation. We intend to have the display screen at pre-known distances away from the subject's face appropriate for the camera's point of view also.

2) Software baseline

a) Fabian Timm algorithm

The algorithm used by the system for detecting and analyzing blinks is automatically initialized and it is only dependent upon the involuntary blinking of the user. For doing the same, motion analysis techniques are used, along with creation of a template of the open eye online which is used for further tracking and template.[2]

B. Steps

1) Initialization

Naturally, the first step is locating the eyes. To accomplish this, the difference image of each frame and the previous frame is created and then thresholded finally giving a binary image which shows the movement regions that had occurred in between the two frames.

Now, we pass a 3x3 kernel over the binary difference image which eliminates noise and naturally-occurring jitter that may have been present due to lighting conditions or camera resolution, and also the possibility of background movement. In addition, this Opening operation also produces fewer and larger connected components in the vicinity of the eyes (when a blink happens to occur), which is crucial for the efficiency and accuracy of the next phase.

A recursive labelling procedure is applied next to recover the number of connected components in the resultant binary image. In the case that other movement besides the eye movement has occurred, producing a much larger number of components, the system discards the current binary image and waits to process the next involuntary blink in order to maintain efficiency and accuracy in locating the eyes.

Given an image with a small number of connected components output from the previous processing steps, the system is able to proceed efficiently by considering each pair of components as a possible match for the user's left and right eyes. The filtering of unlikely eye pair matches is based on the calculation of six parameters for each of the component pairs which are the height and width of each of the two components, vertical and horizontal distance between the centroids of the two components. A number of experimentally-derived heuristics are applied to these statistics to pinpoint the exact pair that most likely represents the user's eyes. For example, if there is a large difference in either the width or height of each of the two components, then they likely are not the user's eyes. As an additional example of one of these many filters, if there is a large vertical distance between the centroids of the two components, then they are also not likely to be the user's eyes, since such a property would not be humanly possible. Such observations not only lead to accurate detection of the user's eyes, but also speed up the search greatly by eliminating unlikely components immediately [5].

2) Template creation

The size of the template which will be created is directly proportional to the size of the chosen component. So, the larger of the two locations tracked in the previous step is chosen so that we get more brightness information, and thus more accurate tracking and correlation scores.

Since the system will be tracking the user's open eye, template shouldn't be taken for when the user was blinking. So, once the eye is located, we trigger a timer. Now after a small number of frames have elapsed, that is about the approximate time for the user's eye to be again open after an involuntary blink, the user's open eye template is created. Thus, during initialization, we assume that the user will be blinking at a normal rate of one involuntary blink every few moments. No offline templates are required as the creation of the online template is independent of past templates created during the run of the system [5].

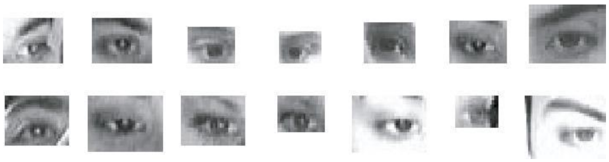

Fig 2: Open eye templates

3) Eye-tracking

The use of template matching is necessary for the desired accuracy in analyzing the user's blinking since it allows the user some freedom to move around slightly. Though the primary purpose of such a system is to serve people with paralysis, it is a desirable feature to allow for some slight movement by the user or the camera that would not be feasible if motion analysis were used alone.

The result of this computation is a correlation score that should the similarity between the open eye template and all points in the search region of the video frame. The value of the scores can indicate if it is an open or closed eye template of loss of precision. A major benefit of using this similarity measure to perform the tracking is that it is insensitive to constant changes in ambient lighting conditions.

Since this method requires an extensive amount of computation and is performed 30 times per second, the search region is restricted to a small area around the user's eye. This reduced search space allows the system to remain running smoothly in real time since it drastically reduces the computation needed to perform the correlation search at each frame [5].

4) Blink detection

Blink detection and the analysis of the duration of the blink are solely based on the correlation scores generated by the tracking done using the online template of the user's eye. As the user's eye closes as the user blinks, the current template's similarity to the open eye template decreases. Correspondingly, it regains its similarity to the template when the user's eye becomes fully open again and the blink ends. This decrease and increase in similarity corresponds directly to the correlation scores returned by the template matching procedure.

Given these ranges of correlation scores and knowledge of what they signify derived from experimentation and observation across a number of test subjects, the system detects voluntary blinks by using a timer that is triggered each time the correlation scores fall below the threshold of scores that represent an open eye. If the correlation scores remain below this threshold and above the threshold that results in re-initialization of the system for a defined number of frames that can be set by the user, then a voluntary blink is judged to have occurred, causing a mouse click to be issued to the operating system [5].

IV. EVALUATION OF PARAMETERS

Based on the reviews of the various papers studied, the following are the shortlisted parameters that need to be considered for implementation:

- Camera resolution: A trade-off between cost and effective detection is being done while choosing the camera resolution.
- Camera position: The distance of the camera from the subject's eye is very crucial for eye detection. Easier and more accurate detection can be obtained by reducing the distance.
- Stability: Based on the evaluation of other experiments as well as a set-up of our own, it can be averred that the use of a chinrest improves the performance by 32% horizontally as compared to the setup without it. This reduces the variance among the people or subjects, thus increasing the experiment's reliability.
- Subjects: Different people have different eye structures causing differences in their eye detections. Depending if they use eyeglasses or not, their head movements during the experiment, their involuntary blinks, the output may vary. The aim is to create a model which is minimally affected by these factors.
- Lighting conditions: This is an important factor as it affects not only the image quality but also the camera frame rate. Based on our research, we have perceived that this factor may have an effect on corner detection algorithms and part detectors, as disturbances like shadows and noise can appear in the webcam image.

V. CONCLUSION

We have shown that it is plausible for an unmodified web camera to be used for eye tracking. If further research helps us to achieve the specified goals and precautions, a usable eye tracking interface could be implemented which requires no special hardware or setup costs and involves a simple software too. Furthermore, it could also implement blink detection.

ACKNOWLEDGMENT

We have great pleasure in presenting this review paper on "Real time eye tracking using web camera". We take this opportunity to thank all those who have contributed in successful completion of this report.

We wish to express sincere thanks and deep sense of gratitude to respected mentor Asst. Prof. Ruhina Karani**,** in Computer Engineering department of D.J. Sanghvi College of Engineering, for her support with all of her kindness. We are also thankful to our parents for their continuous encouragement to pursue higher studies and our friends for the help and support they provided.

REFERENCES

[1] Rommel Anacan, James Greggory Alcayde, Retchel Antegra, Leah Luna, “Eye-GUIDE (Eye-Gaze User Interface Design) Messaging for Physically-Impaired People”, International Journal of Distributed and Parallel Systems (IJDPS) Vol.4, No.1, January 2013.

[2] Fabian Timm, Erhardt Barth, “Accurate Eye Centre Localisation By Means Of Gradients”, Institute for Neuro- and Bioinformatics, University of L¨ubeck, Ratzeburger Allee 160, D-23538 L¨ubeck, Germany, Pattern Recognition Company GmbH, Innovations Campus L¨ubeck, Maria-Goeppert-Strasse 1, D-23562 L¨ubeck, Germany.

[3] Weston Sewell, Oleg Komogortsev, “Real-Time Eye Gaze Tracking With an Unmodified Commodity Webcam Employing a Neural Network”, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI), Atlanta, GA, 2010.

[4] Reji Mathews, Nidhi Chandra, “Computer Mouse using Eye Tracking System based on Houghman Circle Detection Algorithm with Grid Analysis”, International Journal of Computer Applications (0975 – 8887) Volume 40– No.13, February 2012.

[5] Michael Chau, Margrit Betke, “Real Time Eye Tracking and Blink Detection with USB Cameras”, Boston University Computer Science Technical Report No. 2005-12, May 12, 2005.

[6] Onur Ferhat, Fernando Vilariño, “Eye-Tracking with Webcam-Based Setups: Implementation of a Real-Time System and an Analysis of Factors Affecting Performance, Master in Computer Vision and Artificial Intelligence Report of the Master Project Option: Computer Vision”.

[7] Erna Demjén, Viliam Aboši, Zoltán Tomori, “Eye Tracking Using Artificial Neural Networks For Human Computer Interaction”, Physiological Research Pre Press Article.