AUTOMATIC EYE TEMPLATE GENERATION AND TRACKING

RICHA MEHTA¹ & MANISH SHRIVASTAVA²

¹²Department of Information Technology, Laxmi Narain College of Technology, Bhopal, India
Email: richa07mehta@gmail.com, Maneesh.shreevastava@yahoo.com

Abstract- Eye blinking is a physiological necessity for humans. This method automatically locates the user’s eye by detecting eye blinks. A system is the improvement of driver carefulness and accident reduction. The driver’s face is tracked while he is driving and he is warned if there seems to be an alerting fact that can result in an accident such as sleepy eyes, or looking out of the road. Furthermore, with a facial feature tracker, it becomes possible to play a synthesized avatar so that it imitates the expressions of the performer. For a user who is incapable of using her hands, a facial expression controller may be a solution to send limited commands to a computer. Eye blinking is one of the prominent areas to solve many real world problems. The process of blink detection consists of two phases. These are eye tracking followed by detection of blink. The work that has been carried out for eye tracking only is not suitable for eye blink detection. Therefore some approaches had been proposed for eye tracking along with eyes blink detection. In this thesis, real time implementation is done to count number of eye blinks in an image sequence. At last after analyzing all these approaches some of the parameters we obtained on which better performance of eye blink detection algorithm depend. This project focuses on automatic eye blink detection in real time. The aim of this thesis is to count the number of eye blinks in a video. This project will be performed on a video database of the facial expressions.

Keywords-Eye Tracking; Blink Detection; Eye template.

I. INTRODUCTION

There has been a growing interest in the field of facial expression recognition especially in the last two decades. The primary contribution of this research is automatically initializing the eye blink tracking and detection in an image sequence for real time eye blinking and tracking applications. Tracking and blinking the eye parameters and detecting eye states is more difficult than just tracking and blinking the eye locations because the eyes occupy a small region of the face. Most eye trackers work well for open eyes. However, blinking is a physiological necessity for humans. Moreover, for applications such as facial expression analysis and driver awareness systems, we need to do more than tracking the locations of the person’s eyes but obtain their detailed description. We need to recover the state of the eyes (whether they are open or closed), and the parameters of an eye model (e.g. the location and radius of the iris, and the corners and height of the eye opening). We develop a model based system of tracking eye features that uses convergent tracking techniques and show how it can be used detect whether the eyes are open or closed, and to recover the parameters of the eye model. Eye tracking has received a great deal of attention. As blinking is a physiological necessity for humans. An example of such a system is the improvement of driver carefulness and accident reduction. The driver’s face is tracked while he is driving and he is warned if there seems to be an alerting fact that can result in an accident such as sleepy eyes, or looking out of the road. Furthermore, with a facial feature tracker, it becomes possible to play a synthesized avatar so that it imitates the expressions of the performer. Human-Computer Interaction (HCI) systems may also be enriched by a facial feature tracker. For a user who is incapable of using her/his hands, a facial expression controller may be a solution to send limited commands to a computer. For many people with physical disabilities, computers form an essential tool for communication, environmental control, Security, education and entertainment. However access to the computer may be made more difficult by a person’s disability. A number of users employ head-operated mice or joysticks in order to interact with a computer and to type with the aid of an on-screen keyboard. Head-operated mice can be expensive. The implementation of a system such as the proposed one presents several areas of difficulty:

1. Automatic Frame generation in video image sequence
2. Identifying and tracking the location of the eyes.
3. Eye Template generation in image sequence.
4. Being able to process the information in real-time using a moderately priced processor that will be running other applications in the foreground (e.g., Microsoft Word).

II. LITERATURE REVIEW

Much of the eye-detection literature is associated with face detection and face recognition. Direct eye-detection methods Search for eyes without prior information about face location, and can further be classified into passive and active methods.

Most eye trackers require manual initialization of the eye location before they can accurately track eye-features for real-time applications. A method for locating the eyes in static images was developed by Kanade in 1973 and has been improved by other
people over the years. Most of these researchers have based their methods on Yuille’s deformable templates to locate and track eye-features. This method looks for the valleys and peaks in the image intensity to search for the eyes. Once the location of the eyes is determined, its position information is used as a priori knowledge to track the eyes in succeeding frames. But it requires the eye template to be initialized manually at or below the eye otherwise it detects the eyebrow instead of the eye. Hallinan has tried to build an automated model for deformable templates the best candidates for the eye pair, but in order to make his method invariant to scaling, the template is initialized at different sizes at various places and the best candidates for the eyes are selected. Chow et al. make use of the Hough transform in combination with the deformable templates to extract eye-feature points, but this approach is also time consuming as the Hough Transform for various scales had to be applied prior to detecting the iris, unless the approximate radius of the iris is known in advance. Deng and Lai presented the concept of regional forces and regional torque to accurately locate and resize the template on the iris even when the iris is in an extreme position, and for the correct orientation of the template. But their method also requires hand initialization to the position of the eye window before it can successfully locate and track the eyelids. All these methods track the eyes from frame to frame by readjusting the template of both the iris and the eye contour. Tian et al. have shown that such an approach is prone to error if the eyes blink in the image sequence.

A. Eye Authentication Processing

An image may be defined as a two-dimensional function, f(x, y) where x and y are spatial coordinates, and the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements and pixels. Pixel is the term most widely used to denote the elements of a digital image Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. There is no general agreement among authors regarding where image processing stops and other related areas, such as image analysis and computer vision, start. Sometimes a distinction is made by defining image processing as a discipline in which both the input and output of a process are images. We believe this to be a limiting and somewhat artificial boundary. [9]

![Fig. 1: Image acquisition](image1.png)

**Eye Authentication Processing** Two broad categories can be defined methods whose input and output are images, and methods whose inputs may be images, but whose outputs are attributes extracted from those images. This organization is summarised in Fig. 1. The diagram does not imply that every process is applied to an image. Rather, the intention is to convey an idea of all the methodologies that can be applied to images for different purposes and possibly with different objectives. [10] Image Acquisition is the first process shown in figure 1. Acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling. Image enhancement is one of the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image.

B. Eye Tracking and Detection

Video-based eye tracking has become one of the most popular and successful eye-tracking techniques. A multi-stage eye tracker with similar constraints to the multi-stage lip tracker. For the first stage, the eye centre in the previous frame and find the centre of mass of the eye region pixels. Then we search a 5 x 5 window around the centre of mass and look for the darkest pixel, which corresponds to the pupil. If this estimate produces a new eye centre close to the previous eye centre then we take this measurement [1]. If this stage fails, we run the second stage, where we search a window around the eyes and analyze the likelihood of each non-skinned connected region being an eye. We limit the 69 search space to a 7 x 20 window around the eye. We find the slant of the line between the lip corners. The eye centres we select are the centroids that have the closest slant to that of the lip corners. Still, this method by itself can get lost after occlusion. For simplicity in our description, we refer to these two stages together as the eye black hole tracker. The third stage, which we call the affine tracker, runs in parallel with the first two stages. Since automatic initialization yields the eye centers, we construct windows around them, and then in subsequent frames, consider a second window centered on the same point. We compute the affine
transformation between the windowed sub-images and then, since we know the eye centre in the previous frame, we warp the sub image of the current frame to find the new eye centre. Thus, we have two estimates for the eye centres, one from the eye black hole tracker and one from the affine tracker. When there is rotation or occlusion or when the eye black hole tracker produces an estimate that is too far away from the previous frame, we use the affine tracker slowly. In all other cases we take an average of the two trackers to be the eye centre. Later, we discuss how we detect rotation.

1) Eye position initialization

We assume the initial location of the eye is given in the first frame. The purpose of this stage is to get the initial eye position in the first frame of the image sequence. Some literature about eye locating has been published.

2) Eye Region intensity Normalization

For some image sequences, the eye region is very dark because of eye makeup or poor illumination. We therefore normalize the intensity of the image sequence. After the eye positions are initialized, a fixed size window is taken around the eye region. The intensities in this region are linearly stretched to fill the 0 - 255 range. For color image sequences, the R, G, B channels are stretched separately.

3) Eye Corner Tracking

It is assumed the initial location of the eye is given in the first frame. The purpose of this stage is to get the initial eye position in the first frame of the image sequence. It is found that eye inner corners are the most stable features in a face and relatively insensitive to facial expressions. Using an edge based corner detector, the inner corners can be detected easily. However, due to the low intensity contrast at the eye boundary and the wrinkles around the eye, some false corners will be detected as well as the true corners. Instead of using the corner matching method, we therefore use a feature point tracking method to track the eye inner corners for the remaining images of the sequence.

![Fig 2](image)

(a) Detected corners in the eye-blink pair; (b) Corners adjusted according to the slope between the outer corners.

C. Eye Blink Detection

Eye detection, the task of finding and locating eyes in images, is used in a great number of applications. Blink Detection, Blinking is defined as the rapid closing and opening of the eye lid [2]. The average duration of an eye-blink is 0.5 to 0.6 seconds, with a frequency varying from once every two seconds up to several tenths of a second. The blinking rate can also be affected by external stimulus such as fatigue, eye injury, medication or disease. Much of the eye-detection literature is associated with face detection and face recognition see, e.g. [3, 4]. Direct eye-detection methods search for eyes without prior information about face location, and can further be classified into passive and active methods.

Passive eye detectors work on images taken in natural scenes, without any special illumination and therefore can be applied to movies, broadcast news, etc. One such example exploits gradient field and temporal analysis to detect eyes in gray-level video [5]. Active eye-detection methods use special illumination and thus are applicable to real-time situations in controlled environments, such as eye-gaze tracking, iris recognition, and video conferencing. They take advantage of the retro-reflection property of the eye, a property that is rarely seen in any other natural objects. When light falls on the eye, part of it is reflected back, through the pupil, in a very narrow beam pointing directly towards the light source. When a light source is located very close to a camera focal axis (on-axis light), the captured image shows a very bright pupil [6, 7]. This is often seen as the red-eye effect when using a flashlight in stills photography. When a light source is located away from the camera focal axis (off-axis light), the image shows a dark pupil. This is the reason for making the flashlight units pop up in many camera models. However, neither of these lights allow for good discrimination of pupils from other objects, as there are also other bright and dark objects in that would generate pupil-like regions in the image.

D. Face Resolution

Most systems proposed in the literature attempt to recognize facial expressions from high resolution faces (face regions are always greater than 200x200 pixels). However, for real-life applications, face resolutions can be affected by the quality of camera or the distance of user to camera, high resolution input cannot be guaranteed. Since facial images with coarse resolution can provide less information about facial features, algorithms that work well for high resolution face images can be expected to perform poorly when the resolution of input degrades.

E. Environment Variation

The variations of recording environment such as complex background pattern, presence of other people and uncontrolled lighting conditions have a potentially negative effect on expression recognition. As discussed above, in most of the training data sets, background of the images is neutral or has a consistent pattern and only a single person is present
in the scene. When input images are captured in a clustered scene, face detector trained by data set without corresponding variations are difficult to perform reliably [8]. Similar to low resolution input, images acquired in low lighting conditions may also provide less information about facial features.

III. METHODOLOGY

To make an automatic eye blink tracking and detection system for a video, we require extracting and tracking the eyes movements in an image sequence. For making such type of system, we have included three distinct phases: First, eyes are detected in each frame of a video. Motion analysis techniques are used in this stage, followed by online creation of a template of the open eye to be used for the subsequent tracking and template matching that is carried out at each frame. A flow chart depicting the main stages of the system is shown in Figure 3.

A recursive labelling procedure is applied next to recover the number of connected components in the resultant binary image. Under the circumstances in which this system was optimally designed to function, in which the users are for the most part paralyzed, this procedure yields only a few connected components, with the ideal number being two (the left eye and the right eye). In the case that other 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 computation of six parameters for each component pair: the width and height of each of the two components and the horizontal and vertical 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.

IV. RESULTS

A large volume of data was collected in order to assess the system accuracy. For experiment, total 30 videos are used in different lighting condition using inbuilt USB camera of Dell 5050 laptop. The size of each frame is 480 x 640. The result may be tested for more number of videos. The overall theme of our experiment was to capture the original blinks with the help of web cam, then converting these original blinks into automatic blinks with the help of Matlab code and thus calculated the missed blinks by comparing the original blinks with that of automatic. This experiment has various applications which can be used for security and up gradation of various technical sectors. From each person different states of the eyes were chosen, i.e. closed, open, half open, gazing and diverse head poses.

![Manually Eye Blinks Detection](image326x85to506x160)

![Fig: 4.1 Manually Eye Blink Detection](image501x174to514x232)
A. Manually Eye Blink Detection: Here the dotted line represent the original eye blinking which were taken in the different videos with different values, the time taken in the video was about 6 to 8 second, through the blue dots shows the value of original eye blink tracking. In Fig. 4.1 the graph represents two dimensions ie; X and Y, In which X co-ordinate shows the total number of records and Y-coordinates shows the number of eye blinks in each video. The blinks are detected manually through web cam, the videos used here are all based on real time. Thus the graph is plotted here is for keeping a record to compare the blinks with that of the automatic blink, thus we can say that this graph is basically plotted to keep a record of natural blinks.

B. Automatic Eye Blink Detection: Dotted line represent the automatic eye blink detection taken in the different values; video timing is 6 to 8 second, through the blue dotted line shows the value of automatic eye blink tracking. In Fig. 4.2 the graph represents two dimensions ie; X and Y, In which X co-ordinate shows the total number of records and Y-coordinates shows the number of eye blinks in each video. The blinks are detected automatically through matlab coding. In some video the natural blinks are same as that of the automatic blink in the video, while in some the count was less and more then the natural blink. Thus in this graph we are able to compare the blinks with regard to automatic and natural.

VI. CONCLUSION

The system proposed in this thesis provides an automatic eye blink tracking and detection for people with disabilities similar to the one presented by Grauman et al. [9]. However, some significant improvements and contributions were made over such predecessor systems. The automatic initialization phase (involving the motion analysis work) is greatly simplified in this system, with no loss of accuracy in locating the user’s eyes and choosing a suitable open eye template. Given the reasonable assumption that the user is positioned anywhere from about 1 to 2 feet away from the camera, the eyes are detected within moments. As the distance increases beyond this amount, the eyes can still be detected in some cases, but it may take a longer time to occur since the candidate pairs are much smaller and start to fail the tests designed to pick out the likely components that represent the user’s eyes. In all of the experiments in which the subjects were seated between 1 and 2 feet from the camera, it never took more than three involuntary blinks by the user before the eyes were located successfully.

After studying and analyzing results of above technique following points is concluded:
1. A good accuracy is achieved in different illumination conditions.
2. The initialization technique is efficient and gives good results.
3. The system responds slowly and requires more work for real time implementation.
4. Testing must be done on large database of videos.

REFERENCES

[1] Farhan Zaidi, T.Morris, P.Blenkhorn, “Blink detection for real time eye tracking”, Journal of Network and Computer Applications (2002) pages 129-143 February 2002.
[2] Michael Chau and Margit Betke “Real time eye tracking and blink detection using USB camera”, Boston University Computer Science Technical Report No. 2005-12.
[3] D. G. Evans, R. Drew, P. Blenkhorn, “Controlling Mouse Pointer Position Using an Infrared Head Operated Joystick”, IEEE TRANSACTIONS ON REHABILITATION ENGINEERING, VOL. 8, pages 107-116, MARCH 2000.
[4] Y.-L. Tian, T. Kanade & J. Cohn 2000. Dual-State Parametric Eye tracking. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, pages 26-30 March 2000.
[5] J.-Y. Deng & F. Lai. Region-Based Template Deformation and Masking for Eye-Feature Extraction and Description. Pattern Recognition 30, pages 403-419, MARCH 1997.
[6] C. Fagiani, M. Betke, and J. Gips. Evaluation of tracking methods for human-computer interaction. Proceedings of the IEEE Workshop on Applications in Computer Vision (WACV 2002), pages 121–126, Orlando, Florida, December 2002.
[7] Y. Tian, T. Kanade & J. Cohn 1999. Multi-State Based Facial Feature Tracking and Detection.Robotics Institute, Carnegie Mellon University, Technical Report CMU-RI-TR-99-18.
[8] X. Xie, R. Sudhakar & H. Zhuang. On Improving Eye Feature Extraction Using Deformable Templates. Pattern Recognition 27, pages791-799, 1994.
[9] K. Grauman, M. Betke, J. Gips, and G. Bradski. Communication via eye blinks - detection and duration analysis in real time. Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR 2001), Vol. 2, pages 1010–1017, Kauai, Hawaii, December 2001.
[10] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G. Bradski. Communication via eye blinks and eye brow raise: Video-based human-computer interfaces. Universal Access In The Information Society, 2(4), pages 359–373, November 2003.