

## Real Time Eye Template Detection and Tracking

Richa Mehta<sup>1</sup>, Manish Shrivastava<sup>2</sup>

Department of Information Technology, LNCT, Bhopal, India<sup>1,2</sup>

### Abstract

*There has been a growing interest in the field of facial expression recognition especially in the last two decades. 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 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 are 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 to 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. Identifying and tracking the head location.
2. Identifying and tracking the location of the eyes.

3. Automatic Eye blinking and detecting blinks of the eyes.
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 has shown that such an approach is prone to error if the eyes blink in the image sequence.

### **A. 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 center in the previous frame and find the center of mass of the eye region pixels. Then we search a 5 x 5 window around the center of mass and look for the darkest pixel, which corresponds to the pupil. If this estimate produces a new eye center close to the previous eye center 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-skin 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 comers. The eye centers we select are the centroids that have the closest slant to that of the lip comers. 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 center in the previous frame, we warp the sub image of the current frame to find the new eye center. Thus, we have two estimates for the eye centers, 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 center. Later, we discuss how we detect rotation.

### **B. 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.

### **C. Pose of the Head**

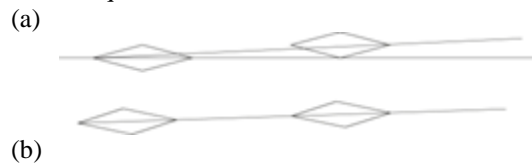
Since most systems use single fixed camera setups, constraints are often imposed on the position and orientation of the head relative to the camera to ensure the input image has the face in frontal view or near frontal view. However, in reality, head rotations occur frequently, pose invariant expression recognition methods need to be developed. In-plane rotations and limited out-of-plane rotations of the head may be partly handled by normalization before facial feature extraction.

#### **D. 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. In experiments, we found that our tracker works well after this intensity normalization for those images with dark eye regions.

#### **E. 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 1: (a) Detected corners in the eye-blink pair;  
 (b) Corners adjusted according to the slope  
 between the outer corners.**

#### **F. 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.

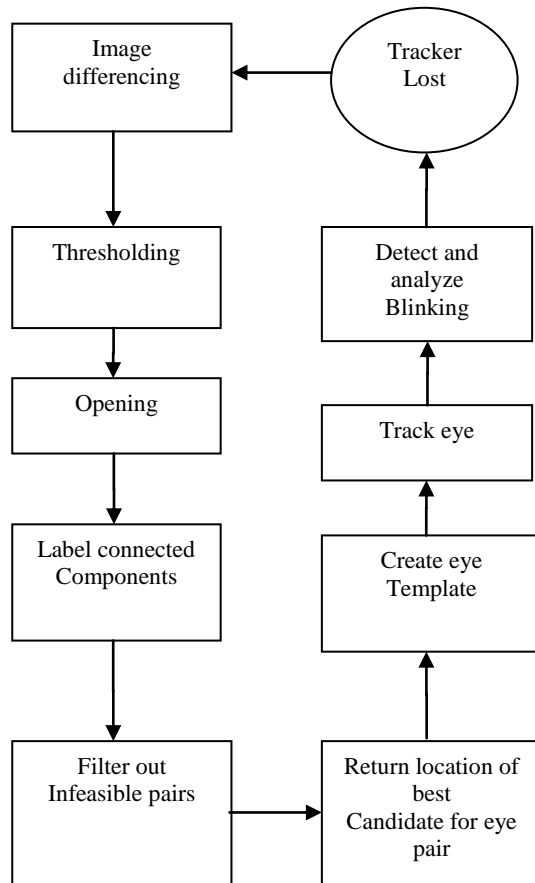
Passive eye detectors work on images taken in natural scenes, without any special illumination and therefore can be applied to movies, broadcast news, etc. One such example exploits gradient field and temporal analysis to detect eyes in gray-level video [5]. Active eye-detection methods use special illumination 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 the would generate pupil-like regions in the image.

#### **G. 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. Proposed Method

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.



**Fig 2: Overview of the main stages in the system**

The analyzing the blinking of the user is to locate the eyes. To accomplish this, the difference image of each frame and the previous frame is created and then threshold, resulting in a binary image showing the regions of movement that occurred between the two frames. Next, a 3x3 star-shaped convolution kernel is passed over the binary difference image in an Opening morphological operation. This functions to

eliminate a great deal of noise and naturally-occurring jitter that is present around the user in the frame due to the lighting conditions and the camera resolution, as well as 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 labeling 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. Application

The first application demands high image acquisition rate and adequately fast response from the image

analyzing system. This motivates the requirements of the output:

1. Accurateness for the application purposes
2. Real-time response

The basic two application of the proposed blink detection are following:

### **1) Computer input with human eyes only**

One of the examples of application of the proposed method for computer input with human eyes only is introduced. Mouse is one of computer input devices which allows input the desired key into computer.

Basically, eye-mouse has the same functions as those of general mouse. It, however, does work with human eyes only without any direct touch with computer. Eye-mouse is getting more important since it can be used for disable and handicapped persons as well as elderly persons who are not good at typing with keyboard to use computer. Then the eye blink can be used as a computer input device.

### **2) Identify the Drowsiness of during driving**

An active computer vision system is proposed to extract various visual cues for drowsiness detection of drivers. Drowsiness and increased tendency to fall asleep during daytime is still a generally underestimated problem. An increased tendency to fall asleep limits the efficiency at work and substantially increases the risk of accidents. Reduced alertness is difficult to assess, particularly under real life settings.

## **V. Conclusion**

According to this paper we can propose an accurate and fast method for locating and tracking the eyes of a computer user situated in front of the monitor. Once blinking is detected, measures like blinking rate, number of eye closures and eye positions are easily obtained. Blink detection is a viable technique for automatically initializing the eyes locations. We have demonstrated that eye-feature points obtained from the blink detection process can be successfully tracked in the succeeding frames of an image sequence. The approach is real-time and is robust with respect to variations in scaling and lighting conditions.

## **References**

- [1] Michael Chau and Margrit Betke "Real time eye tracking and blink detection using USB camera", Boston University Computer Science Technical Report No. 2005-12.
- [2] D.O. Gorodnichy. On importance of nose for face tracking. Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002), pages 188–196, Washington, D.C., May 20-21 2002.
- [3] A. L. Yuille, D. S. Cohen & P. W. Hallinan. Feature Extraction from Faces Using Deformable Templates. Proceedings Computer Vision and Pattern Recognition, pages 104-109, 2008.
- [4] J. Lombardi and M. Betke. A camera-based eyebrow tracker for hands-free computer control via a binary switch. Proceedings of the 7th ERCIM Workshop, User Interfaces For All (UI4ALL 2002), pages 199–200, Paris, France, October 2002.
- [5] J. Deng and F. Lai. Region based template deformation and masking for eye feature extraction and description. Pattern Recognition, pages 403–419, 1997.
- [6] M. Betke, W. Mullally, and J. Magee. Active detection of eye scleras in real time. Proceedings of the IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis (HMAS 2000), Hilton Head Island, SC, June 2000.
- [7] D.O. Gorodnichy. Towards automatic retrieval of blink-based lexicon for persons suffered from brainstem injury using video cameras. Proceedings of the CVPR Workshop on Face Processing in Video (FPIV2004), Washington, D.C., June 28 2004.
- [8] Y.-I. 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.
- [9] G. Chow & X. Li 1993. Towards a System for Automatic Facial Feature Detection. Pattern Recognition 26, 17391755.



**Richa Mehta** pursuing M.tech degree in Information Technology from Rajiv Gandhi Proudhyogiki Vishwavidyalaya (RGPV), Bhopal, India. Her research area is Image Processing.