Eye Gaze Detection Using Optical Flow Concept for Man Machine Interaction

Leema Roselin.G [1], Sudharsan. N. S [2]

[1]P.G Student, [2]Assistant Professor, Department of CSE,
T.S.M.Jain College of Tecchnology
[1] gleemaroselin@gmail.com,[2] sudharsan.n.s@gmail.com

Abstract— **A Man-Machine interaction system that is designed for persons with harsh disabilities to reproduce control of a conventional personnel machine is introduced. In this Eye-tracking play a major role, it will be highly beneficial to severely disable person. In that the eye gazed data is analyzed in the image liberty that gaze-coordinates were recorded with the help of distinct region of interest. Such analysis, which spotlight primarily on where the screen users are looking, require significant manual input and are not feasible for studies involving many subjects, long sessions, and heavily interactive visual stimuli. On the other hand, it is feasible to collect and analyze eye-tracking information on data space. Exclusively, the visual layout can be used to speak about gaze-coordinates to visualization and data objects that users view in real time. In that the system demonstrates the efficiency of this approach from help of usual but eminent blob detection in the necessary region which is defined as region of interest rather than giving importance of least important region for calculating gaze and blink detection than other existing system. Moreover, this novel approach could translate gaze-coordinates into viewed objects of greater accuracy than simply binning gazes into dynamically.**

Keyword- Chain code, Map Reducing, Social media Support Vector Machine, Un-labelled image, Weakly Labelled image

I. INTRODUCTION

In recent years, there has been an effort to design assistive technology that provides individuals with severe disabilities a tool for communication and access to the computer. Such technology may augment traditional human-computer interfaces like the keyboard and mouse. These traditional human-computer interfaces demand good manual dexterity and refined motor control, which may be absent or unpredictable for people with severe disabilities.

The motivation of our research is to provide an alternative communication tool for non-verbal individuals whose motor abilities are extremely limited by conditions ranging from traumatic brain injuries to degenerative diseases such as multiple sclerosis (MS), muscular dystrophy (MD), or amyotrophic lateral sclerosis (ALS). These individuals may only be able to control their head and eyes. Our goal was to develop a computer vision system that replaces the traditional computer mouse with a system that can be completely controlled with the head and eyes.

The proposed algorithm allows a user to interact with the computer by using their eyes to simulate clicking a traditional mouse. The algorithm is able to automatically locate the user's eyes and learn the appearance of the user's open and closed eyes. Online learning provides a level of robustness that allows the algorithm to work consistently for various individuals and has also shown success for individuals wearing glasses.

Work on camera-based blinks detection has focused on specific c tasks such as human-computer interaction [4, 7] or fatigue detection [5]. Blink detection modules have been part of more general systems on eye motion analysis [1]. Some research e orts in camera-based blinks detection use infrared lighting [8, 13]. The advantage of an infrared system is that the pupils of the user are highlighted when exposed to infrared lighting. While infrared systems make the problem of detecting the eyes easier, the typical user does not have access to infrared lighting and there are safety concerns about long-term exposure to infrared lighting. Our system uses standard lighting with a typical USB camera that is easily available to users. Other systems [1] use active appearance models to locate and track the eyes of the user and make assumptions of the shape, colour, and lighting of the user's eyes. The online templates for the user’s eye that our system automatically captures eliminate the need for us to make such assumptions.

Previous interaction systems for people with disabilities [4, 7] interpret a user blink as a trigger for binary switch applications. By tracking and interpreting both eyes of the user, our system allows interaction with a computer on a level that is closer to using a traditional mouse. Our system enables users to move the mouse pointer towards the screen and issue mouse-clicking commands hands-free. For individuals that are not able to control the muscles around their eyes to a degree that they can wink, the system still enables them to simulate the left-click command of a traditional mouse. This is an improvement in current assistive mouse-replacement systems such as Camera Mouse [2, 3], which limits the user to left-click commands by hovering over a certain location for a predetermined amount of time. This is counterintuitive as the lack of action on the part of the user causes a click to occur. It can lead the system to issue a click command that was not intended by the user if the user is not moving the mouse within the threshold of hovering time. Our system provides a more intuitive method of controlling the mouse, as it requires a specific action by the user to simulate a mouse click.

II. STATE OF ART

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 of locating the eyes in static images was developed by Kanade in 1973[1] and has been improved by other people over the years. Most of these researchers have based their methods of Yuille's deformable templates [2] to locate and track eye-features .This method looks for the valleys and peaks of 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 [3] 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 of 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 [4] 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 for both the iris and the eye contour. Tian has [5] shown that such an approach is prone to error if the eyes blink in the image sequence.

Video-based eye tracking has become one of the most popular and successful eye-tracking techniques. A multi-stage eyes tracker with similar constraints on the multi-stage lip tracker. For the first stage, the eye center of 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, the system run the second stage, where it searches a window around the eyes and analyze the likelihood of each non-skin connected region being an eye

The system limits the 69 search space to a 7 x 20 window around the eye. The system finds the slant of the line of 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 is called as 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.

The system will compute the affine transformation between the window sub-images and then, since we know the eye center in the previous frame, it warps the sub image of the current frame to find the new eye center. Thus, the system has 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 the system will take an average of the two trackers to be the eye center.

Most systems proposed to the literature attempt to recognize facial expressions of high resolution face (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 of 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.

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 occurs 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.

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.

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 of a face and relatively sensitive 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 of the remaining images of the sequence.

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 are 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 from 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 allows 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.

The variations on 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. SYSTEM OVERVIEW

A. Problem Definition

In the growing technology artificial intelligence plays major role in man-machine interaction. The idea is about developing the interaction more (e.g) from ordinary mobile used people are more compact with using smart phones with touch features. Likewise from developing computer interaction from touch to the next level, this could be achieved by controlling the machines through our eye movement. The proposed system involves in tracking the eye gaze movement, it pays ways for controlling any system through eye movement.

The proposed eye-blink detection algorithm consists of six steps:1) image acquisition, 2) face detection, 3) eye localization, 4) eye tracking, 5) eye extraction, 6) eye-blink detection and analysis as shown in Fig 1.

Face detection is performed by Haar-like features and a cascade of boosted tree classifiers [5]. The Haar-like features are computed similar to the coefficients in Haar wavelet transform and each feature is represented by the template, its size and location in the search window. The decision is made by a cascade of boosted tree classifiers. The simple “weak” classifiers are trained as the face images of the fixed size 24×24 pixels. Face detection is done by sliding the search window of the same size as the face images used for training as the test image.

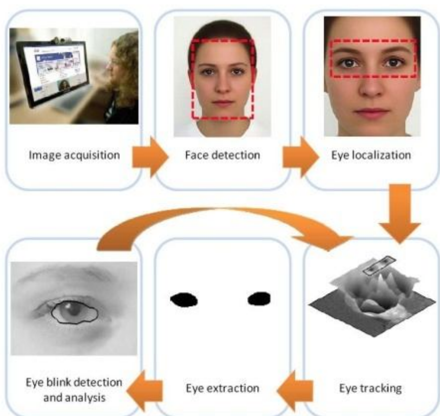

Fig.1. Scheme of eye-blink detection

The position of the eyes in the face image is found on the basis of certain geometrical dependencies observed in a human face. The traditional rules of proportion show to face divided into six equal squares, two by three [6]. According to these rules the eyes are located about 0.4 of the way for the top of the head to the eyes.

The located eye region is extracted from the face image and used as a template for further eye tracking by means of template matching. The extraction of the eye region is performed only at the initialization of the system and in cases when the face detection procedure is repeated. Template matching is realized using normalized cross-correlation method (1) . Normalization is required to reduce the influence of variations in brightness due to lighting and exposure conditions.

The final image processing block of the developed system for eye-blink monitoring is eye-blink detection and an analysis using skin color segmentation and active contour model. Skin color segmentation is performed in YCC color space based on pixel values in the two chrominance channels (Cb and Cr). For working out the threshold values the algorithm based on the Otsu threshold method [7] has been developed.

The detection of eye-blinks in doing on the binary image using an active contour model approach [8]. The active contour model approach is based on matching the computer-generated curve (snake) to object boundaries. In the iterative process of energy function minimization the curve becomes deformed and follows the shape of the object's boundary.

The initial shape of the snake in the proposed method was an ellipse since it is the most natural outline of the human eye. For eye tracking two contours, one for each eye, are employed. The resulting approximated elliptical contours for both eyes are presented in fig. 2. This approximation also facilitates the calculation of the area of the visible part of an eye, which is used for eye-blink detection. When the area A is lower or equal to 20% of the initial area of the ellipse [9] and lasts for at least 100ms the eye-blink is detected.

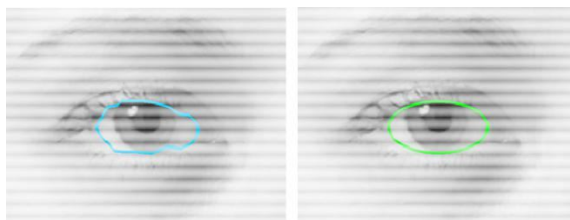

Fig.2 . Resulting active contours for one eye and its elliptical approximation

B. Direction Control Using Optical Flow

Once the eye pattern is identified, it has to subject to optic flow detection in order to control the mouse pointer. Motion estimation has been extensively studied by both computer vision and human vision researchers for more than 25 years. It is also commonly referred to as optical flow computation especially in the computer vision literature. The algorithm that system use here was given by Watson & Ahumada (WA) [10].

The algorithm works by resolving the input stimulus into its component sinusoidal gratings oriented in different directions and estimating the motion of those gratings. This approach work well because determining the motion of a sine grating is easy: there exist a simple relationship between the spatial and temporal frequencies of the grating and its velocity.

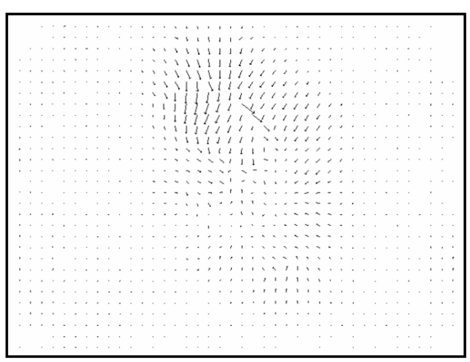

Fig. 3. Optic Flow direction Sensing

It is given by ($\omega x v x+\omega y v y+\omega t=0$), where $\omega x,\omega y$ are the spatial frequencies of the grating vx,vy is the velocity of the grating and ωt is the temporal frequency of the grating.

C. Control of Mouse Pointer

The tracking point located near the upper lip is used to control the location of the mouse. Movement towards the image is mapped to movement towards the mouse. Smoothing is applied to the sequence of mouse pointer locations, so that the mouse pointer is easier to control. The distance the pointer moves on the screen that one-pixel movement towards the tracking point maps to can was adjusted to the user's preference.

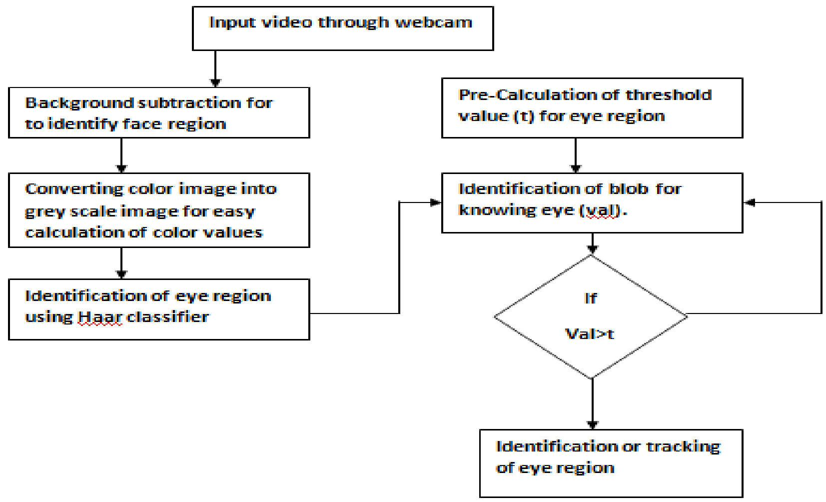

Fig. 4. Architecture diagram: eye region detection.

IV. EXPERIMENTAL RESULTS

The algorithm allows the user three modes of control for simulating mouse command. If a user is able to wink, the algorithm simulates a double-click when a user blinks and a left or right click when the user winks with their left or right eye. The algorithm simulates a left-drag action which occurs when the user closes their left eye and keeps it closed while they move the mouse pointer. When the user opens their left eye, the algorithm sends a left-release signal to the operating system. The movement towards the mouse pointer is made possible using optic flow algorithm. The Experimental result shows good accuracy in identifying the eye activities with 90.2% accuracy. The result is affected with background brightness, but it is negligible, it can be avoided using proper background lighting.

V. CONCLUSION AND FUTURE WORK

In visualizations that is open to instrumentation, gaze information provided by an eye tracker can be used to automatically detect what visual objects users are likely to be viewing. Such detection can provide results that are almost as accurate as annotations created by human coders, provided that detection is done “intelligently”, by using gazes together with a prediction of which objects are likely to be viewed at a given time. Data collected in this way is highly granular and has semantic content because it is linked to the data underlying the visualization. For this reason, and because it does not require any human pre-processing, object viewing data can be collected and analysed efficiently for many subjects, using interactive visualizations, for long analytic session, and could be used in studies that explore how analysts hypothesize about data using complex visual analytics systems.

The future scope of the work is involved in identifying the features that are more useful and extract those features which help in representing the object categories present in the images more consistently using global and local image features in detected eye for more accuracy.

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, June2000.

[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.

[10] Watson, A. B., & Ahumada, A. J., Jr. (1985). Model of human visual-motion sensing,Journal of the Optical Society of America A 2(2), 322-342.