Mouse Control Via Live Eye Gaze Tracking

Najla P R
Department of Electronics and communication
KMCT College of Engineering,
Calicut, India

Jayasree T C
Department of Electronics and communication
KMCT College of Engineering,
Calicut, India

Abstract— **The field of Human-Computer Interaction (HCI) has witnessed a tremendous growth in the past decade. The invention of tablet PCs and smart phones allowing touch-based control has been hailed warmly. The researchers in this field have also explored the potential of eye-gaze as a possible means of interaction. Some commercial solutions have already been launched, but they are as more expensive and offer limited usability. This paper strives to present a low cost real time system for eye-gaze and also concentrates on a human computer interaction application based on eye-gaze tracking. Human eyes carry much information which can be extracted and can be used in many applications. Eye gaze reflects a person's point of interest means it is possible to say that what they are thinking based on where they are looking. Eye gaze tracking is aimed to keep track of human eye-gaze. Eye movements can also be captured and used as control signals to enable people to interact with interfaces directly without the need for mouse or keyboard input. This can be achieved by employing computer vision and image processing algorithms. In the proposed method, first track the human face in a real-time video sequence to extract the eye regions using a webcam. Then, find the facial feature characteristics to extract the eye region to obtain the gaze point. Then corresponding to each gaze point the cursor movement on the monitor screen is done.**

Keywords—Desktop environment,webcamera, point of gaze, head pose estimation, calibration

I. INTRODUCTION

Nowadays personal computer systems are playing a huge role in our everyday lives. They are used in areas such as work, education and entertainment. In all these applications have in common is that the use of personal computers is mostly based on the input method via mouse and keyboard. While this is not a problem for a healthy individual, this may be discomfort for people with limited freedom of movement of their limbs, like people with disabilities or suffering from brainstem strokes, and other handicapped problems. In these cases it would be preferable to use input methods which are based on motor abilities of the head region such as head or eye movements.

So in order to provide such alternative input methods a system was made which follows a low-cost approach to control a mouse cursor on a computer system. It consists of an eye tracker and a head tracker which are both attached to a head mount for locating gaze point corresponding to the eye movement. The eye tracker is based on the images which are recorded by a modified webcam to acquire the eye movements. These eye movements are then mapped to a computer screen to position a mouse cursor accordingly.

Face is the index of mind and eyes are the window to the soul. Eye movements provide a rich and informative window into a person's thought and intentions. So the study of eye movement may determine what people are thinking based on where they are looking. Eye tracking is the measurement of eye movement or activity and gaze (point of regard) tracking is the analysis of eye tracking data with respect to the head or visual scene. Eye tracking is mostly used in the applications like drowsiness detection, diagnosis of various clinical conditions or even iris recognition and also it will help the physically disabled people to use the computer without any external help.

Eye tracking is the process of measuring either the point of gaze that is where one is looking or the motion of an eye relative to the head. An eye tracker is a hardware device for measuring eye positions and eye movement. Eye trackers are used in research on the visual system, in psychology, marketing, as an input device for human computer interaction, and in product design. There are a number of methods for measuring eye movement. The most popular method uses video images from which the eye position is extracted.

II. RELATED STUDY

There are several methods to track the motion of the eyes. The most direct method is the fixation of a sensor to the eye. The fixation of small levers to the eyeball belongs to this category, but is not recommended because of high risk of injuries. A safer way of applying sensors to the eyes is using contact lenses. The big advantage of such a method is the high accuracy and the nearly unlimited resolution in time. For this reason, medical and psychological research uses this method.

Another method is electrooculography (EOG) where sensors attached at the skin around the eyes measure an electric field. Originally, it was believed that the sensors measure the electric potential of the eye muscles. It turned out that it is the electric field of the eye which is an electric dipole. The method is sensitive to electro-magnetic interferences but works well as the technology is advanced and exists already for a long time. The main advantage of the method is its ability to detect of eye movements even when the eye is closed, e.g. while sleeping.

Both methods explained so far are obtrusive and are not suited well for interaction by gaze. The last and preferred method for eye-gaze interaction is video. The central part of this method is a video camera connected to a computer for real-time image processing. The image processing takes the pictures delivered from the camera and detects the eye and the pupil to calculate the gaze's direction. The main advantage of video-based eye tracking is the unobtrusiveness.

Consequently, it is the method of choice for building eye-gaze interfaces for human-computer interaction. Especially for physically disabled people to control the mouse with the movement of their eyes.

*1) Video-Based Eye Tracking:*The task of a video-based eye tracker is to estimate the direction of gaze from the image delivered by a video camera. All video-based eye-tracking methods need the detection of the iris center in the camera image. This is a task for image recognition, typically edge detection, to estimate the elliptical contour of the pupil. As the cornea has a nearly perfect sphere shape, a glint stays in the same position for any direction of gaze while the pupil moves. There are eye trackers which also track the rotational movement of the eye but such systems are not very common. A nice application for such eye trackers is a head camera controlled by the eye movements using the motion stabilization of the eye to get a motion stabilized camera image.

*2) Types of Video-Based Eye Trackers:*The most common mechanical setup is a stationary eye tracker. Such systems are commercially available as a laboratory tool and are typically used in medical research or marketing research. These systems comprise of as a desktop computer with integrated eye tracker and with a software package for analyzing and visualizing eye gaze data.

To accomplish the task of gaze tracking, a number of approaches have been proposed. The majority of early gaze tracking techniques utilize intrusive devices such as contact lenses and electrodes, requiring physical contact with the users; such a method causes a bit of discomfort to users. Tracking the gaze with a head-mounted device such as headgear is less intrusive, but is inconvenient from a practical viewpoint. In contrast, video-based gaze tracking techniques that could provide an effective nonintrusive solution are more appropriate for daily usage.

The video-based gaze approaches commonly use two types of imaging techniques: infrared imaging and visible imaging. The former needs infrared cameras and infrared light sources to capture the infrared images, while the latter usually utilizes high resolution cameras for images. As infrared-imaging techniques utilize invisible infrared light sources to obtain the controlled light and a better contrast image. Eye trackers are the devices which are used to track the eye movement and the position of eye. There are mainly two types of eye trackers, devices are connected to the human body, and devices have no contact with the body.

III. PROPOSED METHOD

The hardware based equipment used for the eye gaze tracking made a discomfortness to the people who were physically disabled**.** Because some devices need a physical contact with the user. So a method has been proposed to develop an eye gaze tracking system that helps the physically disabled people to interact with computer with a better accuracy. That's why it help operate the mouse effectively, access the internet and mails, entertain with games etc. The basic block diagram for this mouse control is shown in figure below. First take the real time video sequences of the person who was sitting in front of the camera. Then transform it from RGB to gray. The most notable gaze features in the face image are the iris center and eye corner. The eye ball moves in the eye socket when looking at different positions on the screen.

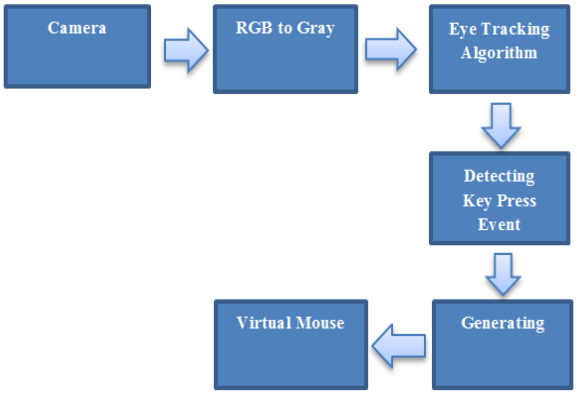

Fig.1. Basic block diagram

The eye corner can be viewed as a reference point, and the iris center in the eyeball changes its position that indicates the eye gaze**.** Therefore, the gaze vector formed by the eye corner and iris center contains the information of gaze direction, which can be used for gaze tracking. However, the gaze vector is sensitive to the head movement and produces a gaze error, while the head moves. Therefore, the head pose should be estimated to compensate for the head movement.

The three-phase feature-based eye gaze tracking approach uses eye features and head pose information to enhance the accuracy of the gaze point estimation shown in Fig. 2. In Phase 1, extract the eye region that contains the eye movement information. Then, detect the iris center and eye corner to form the eye vector. Phase 2 obtains the parameters for the mapping function, which describes the relationship between the eye vector and the gaze point on the screen. In Phases1and 2, a calibration process computes the mapping from the eye vector to the coordinates of the monitor screen. Phase 3 entails the head pose estimation and gaze point mapping. It combines the eye vector and head pose information to obtain the gaze point. The basic block diagram for representing the proposed method is shown below. It consist of mainly three phases that operating effectively to obtain the final gaze point.

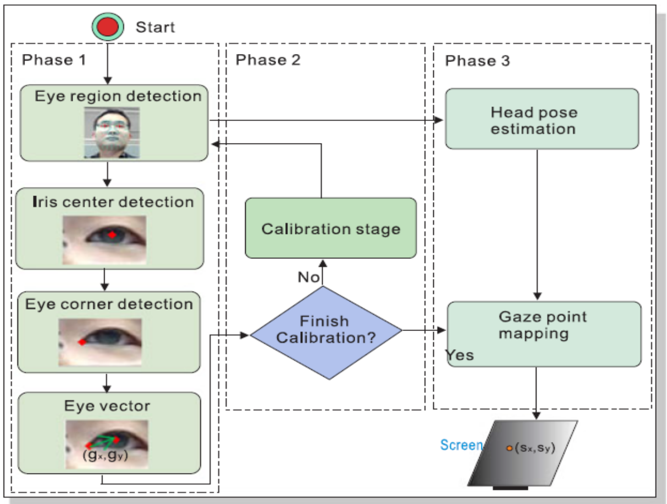

Fig.2. Three-phase feature-based eye gaze tracking method.

The figure above shows the proposed method for obtaining gaze point on the monitor screen. This having three phases based on the feature characteristics of eye. That is the iris center and eye corner. In this inner eye corner is taken as the reference because it is insensitive to the eye movement. And also when there is a head movement occurs it also introduces some errors on the gaze point. So in order to avoid this problem head pose estimation is also taken into consideration.

```
Algorithm1: Eye Gaze Tracking System

Initialization:
-Extracting facial features using ASM
-Extracting the head pose estimation
-Get calibration mapping function
Tracking the gaze through all the frames:
Input:      Video from camera
Step 2:     Extract the eye region
Step 3:     Detect the iris center $p_{iris}$
Step 4:     Detect the eye inner corner $p_{corner}$
Step 5:     Eye vector is obtained: $g = p_{corner} - p_{iris}$
Step 6:     Get static gaze point ($u_x$, $u_y$) by mapping function
```

A. *Eye region detection*

To obtain the eye vector, the eye region should be located first. Traditional face detection approaches cannot provide accurate eye region information in uncontrolled light and with free head movement. Therefore, an efficient approach should be needed for illumination and pose problems. Here, present a two-stage method to detect the eye region. In the first stage, utilize local sensitive histograms to cope with various lighting problems. Compared with normal intensity histograms, local sensitive histograms embed spatial information and decline exponentially with respect to the distance to the pixel location where the histogram is calculated.

Examples of the utilization of local sensitive histograms are shown in Fig.3, in which three images with different illuminations have been transformed to ones with consistent illumination with the local sensitive histograms.

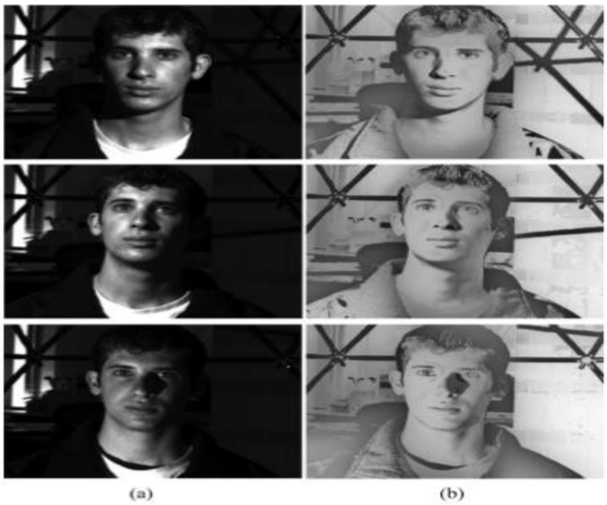

Fig.3. (a) Input images . (b) Results using local sensitive histograms

It operates in a way similar to conventional image histograms. However, instead of counting the frequency of occurrences of each intensity value by adding ones to the corresponding bin in normal histogram, a floating-point value is added to the corresponding bin for each occurrence of an intensity value. The floating-point value declines exponentially with respect to the distance to the pixel location where the locality sensitive histogram is computed. Thus, the proposed histogram is more suitable for applications such as visual tracking for cursor control which assigns lower weights to pixels further away from the target center.

In the second stage, use the active shape model (ASM) to extract facial features on the image, through which the illumination changes are eliminated. Active Shape Models (ASMs) are aimed at automatically locating landmark points that define the shape of any statistically modelled object in an image. When modeling faces the land mark points of interest consist of points that lie along the shape boundaries of facial features such as the eyes, lips, nose, mouth and eyebrows. At the testing stage, the Viola Jones face detector is used for locating the face in an image. Once the face has been detected, the mean face is scaled, rotated and translated using a similarity transform to roughly fit on top of the face in the test image.

*1) Shape model:*An object is described by points, referred to as landmark points. The landmark points are determined manually in a set of training images. From these collections of landmark points, a point distribution model is constructed as follows. The landmark points $(x_1, y_{1,} .... x_n, y_n)$ are stacked in shape

$$X = (x_1, y_{1,} .... x_n, y_n)^T \qquad (1)$$

Principal component analysis (PCA) is applied to the shape vectors X by computing the mean shape.

$$\overline{X} = 1/s1 \sum_{i=1}^{s} x_i \qquad (2)$$

The covariance,

$$S=1/(s-1) \sum_{i=1}^{s} (x_i - \overline{X})(x_i - \overline{X})^T \qquad (3)$$

The eigenvectors corresponding to the ‘ t ‘ largest eigenvalues are retained in a matrix $\phi=(\phi_1\ \phi_2 \ldots \phi)_t$. A shape can now be approximated by

$$X \approx \overline{X} \pm \phi b \qquad (4)$$

where ‘b’ is a vector of elements containing the model parameters, computed by

$$b = \phi^T (X - \overline{X}) \qquad (5)$$

When fitting the model to a set of points, the values of are constrained to lie within the range of $\pm 3\sqrt{\lambda_i}$, for the purpose of generating a reasonable shape. Then make model shapes fit the new input shape by translation t, rotation θ, and scaling s, that is

$$y = T_{X,t,s,\theta}(\overline{X} + \phi b) \qquad (6)$$

Where y is a vector containing the facial features. The eye region can be extracted using an improved version of ASM. In Fig.4, the eye region in each image is detected under the different illumination and head pose.

B. Eye features detection

In the eye region, the iris center and eye corner are the two notable features, by which to estimate the gaze direction. Accordingly, the following two sections focus on the detection of iris center and eye corner.

1) Iris Center Detection: Once the eye region is extracted using the steps discussed above, the iris center will be detected in the eye region. First estimate the radius of the iris. Then, a combination of intensity energy and edge strength information is utilized to locate the iris center. In order to estimate the radius, first smooth the eye region, which can remove noisy pixels and preserve the edges at the same time. Subsequently, the iris center can be obtained by the colour intensity. The eye region obtained by canny edge detection. Then by using this edge and center the radius of iris will be calculated.

Finally, combine the intensity energy and edge strength to locate the iris center. And denote the intensity energy and the edge strength by E1 and E2, respectively

$$E_1 = \sum (I * S_r) \qquad (7)$$

$$E_2 = (g_x^2 + g_y^2)^{(1/2)} \qquad (8)$$

where I is the eye region, and S_r is a circle window with the same radius as the iris. g_x and g_y are the horizontal and vertical gradients of the pixel, respectively. Fig. 4 illustrates the results of iris center detection, in which Fig.4 (a)–(c) are in the same video sequence.

That is, Fig. 4(a) is the first frame, where the iris center could be accurately detected using the proposed algorithm. Accordingly, supposing the radius of the iris does not change with respect to the large distance between the user and the computer screen, then detect the iris center of eye images as shown in Fig. 4(b) and (c).

2) Eye corner detection: Usually, the inner eye corner is viewed as a reference point for gaze estimation because it is insensitive to facial expression changes and eye status , and is more salient than the outer eye corner. Therefore, it is better to detect the inner eye corner to guarantee the gaze direction accuracy**.** A template matching record has been used for our eye corner detection. Having a number of different positions of eye corner, then by template matching can determine the exact eye corner corresponding to our gaze point.

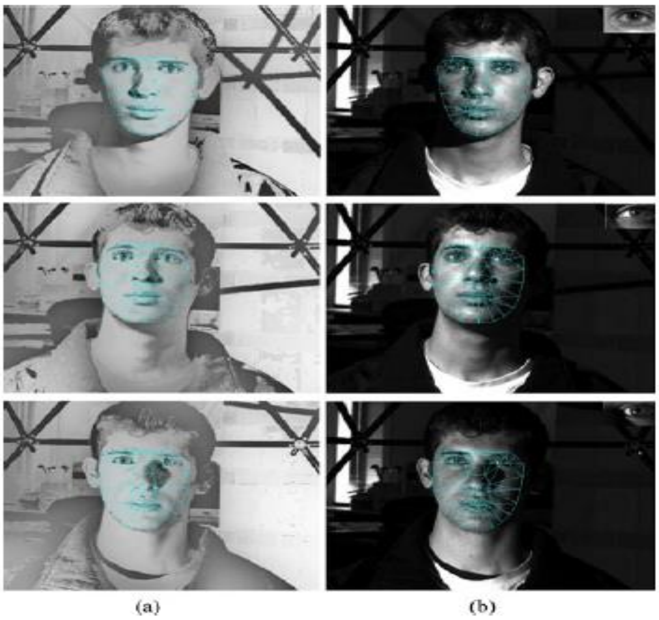

Fig. 4. (a) ASM results on the gray image. (b) Mapping ASM results to the original images and extracting the eye region.

C. Eye vector and calibration

By looking at the different positions on the screen plane, while keeping our head stable, the eye vector is defined by the difference between iris center p_{iris} and the eye corner p_{corner}. That is $g = p_{corner} - p_{iris}$. It provides the gaze information for the cursor control to obtain the screen coordinates by a mapping function. A calibration procedure is to present the user with a set of target points at which to look, while the corresponding eye vectors are recorded. Then, the relationship between the eye vector and the coordinates on the screen is determined by the mapping function. The second-order polynomial function is a good compromise between the number of calibration points and the accuracy of the approximation for the gaze mapping. In our calibration stage, the second-order polynomial function is utilized and the user is required to look at nine points shown in Fig.6. Then the eye vectors are computed and the corresponding screen positions are known. Then, the second-order polynomial can be used as mapping function, which calculates the gaze point on the screen, i.e., scene position, through the eye vector. Then the eye vector coordinates can be represented as,

$$u_x = a_0 + a_1 g_x + a_2 g_y + a_3 g_x g_y + a_4 g_x^2 + a_5 g_y^2 \tag{9}$$

$$u_y = b_0 + b_1 g_x + b_2 g_y + b_3 g_x g_y + b_4 g_x^2 + b_5 g_y^2 \tag{10}$$

Where (u_x, u_y) is the screen position, (g_x, g_y) is the eye vector, (a_1,……..,a_5) and (b_1,..........,b_5) are the parameters of mapping function that can be solved using the least square method

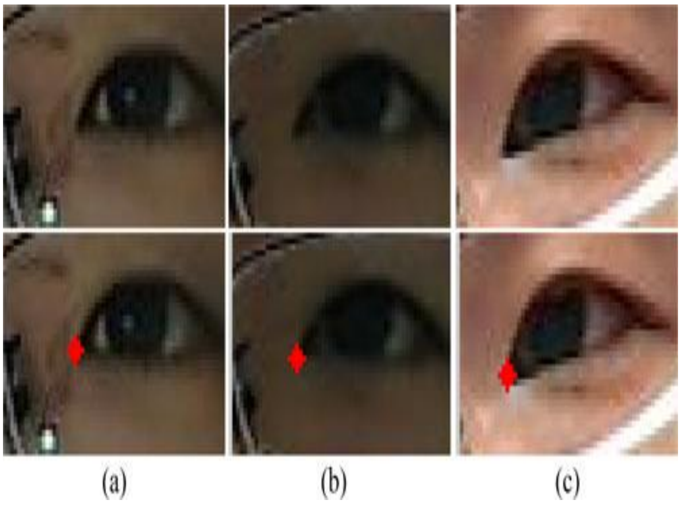

Fig.5.: First row: eye regions, and the second row: eye corner detection results.

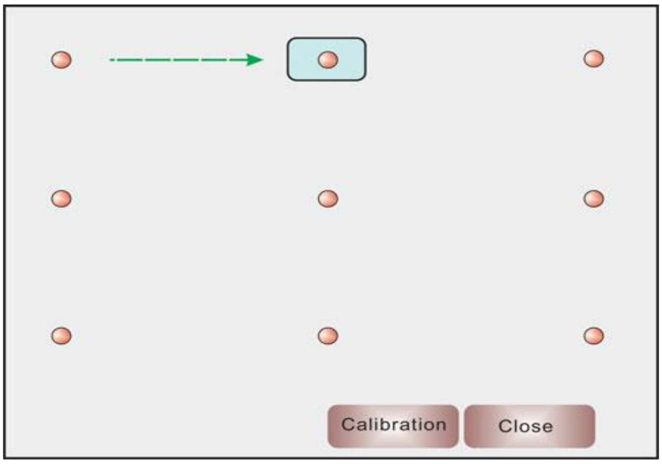

Fig.6: Nine positions on the screen.

D. Head pose estimation

This section discusses about the facial feature tracking and the head pose estimation algorithm in video sequences. Usually, the human head can be modeled as an ellipsoid or cylinder for simplicity, with the actual width and radii of the head for measures.

To improve the estimation of head pose, utilize an Sinusoidal Head Model (SHM) to simulate the 3-D head shape because the ellipsoid and cylinder do not highlight facial features. The SHM can better approximate the shape of different faces with different facial features. 2-D facial features can be related to 3-D positions on the sinusoidal surface. When the 2-D facial features are tracked in each video frame, the 2-D– 3-D conversion method can be utilized to obtain the head pose information. Pose from orthography and scaling with iterations (POSIT) is a 2-D–3-D conversion method to get the pose (i.e., rotation and translation) of a 3-D model given a set of 2-D image and 3-D object points. For a better estimation of the head pose, propose the AWPOSIT algorithm because the classical POSIT algorithm estimates the head pose of the 3-D model based on a set of 2-D points and 3-D object points by considering their contribution or significance uniformly. As for the 2-D facial features, they have different significance with respect to reconstructing the pose due to their reliability. If some features are not detected accurately, the overall accuracy of the estimated pose may decrease sharply with the classical POSIT algorithm. The proposed AWPOSIT can obtain more accurate pose estimation using key feature information. The implementation details follow.

Algorithm 2-AWPOSIT

```
Input: $P_{2D}$, $P_{3D}$, w, f
n=size ($P_{2D}$, 1); c=ones (n, 1)
u=$P_{2Dx}$/f; v=$P_{2Dy}$/f
H=[$P_{3D}$,c]; O=pinv(H)
Loop
    J=O.u; K=O.v
    $Lz 1/(1/\|J\|).(1/\|K\|))^{1/2}$
    $M_1$=J.Lz; $M_2$=K.Lz;
    $R_1$=$M_1$(1:3); $R_2$=$M_2$(1:3)
    $R_3 = (R_1/\|R_1\|) \times (R_2/\|R_2\|)$
    $M_3$=[$R_3$;Lz]
    c=H.$M_3$/Lz
    uu=u; vv=v
    u=c.w.$P_{2Dx}$; v=c.w.$P_{2Dy}$
    $e_x$=u-uu; $e_y$=v-vv
    I f $\|e\| < \epsilon$ then
        $M_4$=[$M_1$(4),$M_2$(4),Lz,1]$^T$; Exit loop
    End if
end loop
Output M
```

The SHM assumes that the head is shaped as a 3-D sine wave in Fig.7 and the face is approximated by the sinusoidal surface. Hence, the motion of the 3-D sine is a rigid motion that can be parameterized by the pose matrix M at frame F_i .

The pose matrix includes the rotation matrix R and translations matrix T at the i^th frame, i.e

$$M = \begin{pmatrix} R & T \\ T^{T} & 1 \end{pmatrix} = [M_1 M_2 M_3 M_4] \qquad (11)$$

where R ∈ $R^{3\times3}$ is the rotation matrix, and T ∈ $R^{3\times1}$ is the translation vector with T =(t^i_x, t^i_y, t^i_z)^T , and M_1 to M_4 are column vectors. The head pose at each frame is calculated with respect to the initial pose, and the rotation and translation matrix can be set at 0 for the initial frame (i.e., standard front face). The ASM model is used on the initial frame to obtain the 2-D facial features. Since these facial features are related to the 3-D points on the sinusoidal model, the movements of which are regarded as summarizing the head motion, and utilize the perspective projection through the camera model for establishing the relation between the 3D points on the

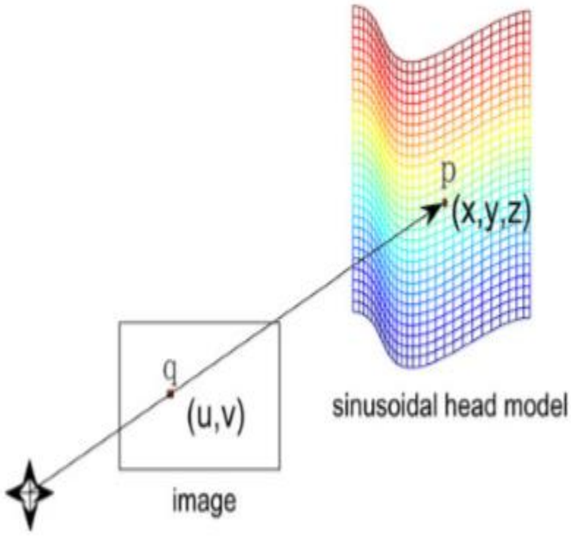

Fig.7. Perspective projection of 3-D point p onto the image plane

Sinusoidal surface and their corresponding projections on the 2-D image plane.Fig.7 shows the relation between the 3-D point p =(x,y,z)^T on the sinusoidal surface and its projection point q =(u,v)^T on the image plane, where u and v are calculated by

$$u = f(x/z) \qquad (12)$$

$$v = f(y/z) \qquad (13)$$

With f being the focal length of the camera. The Fig.8 shows the locations of facial features. The 2-D facial points are denoted as P2D and the 3-D points on the sinusoidal model are denoted as P3D. When the head pose algorithm is available, it estimates the head pose and computes the corresponding displacement (Δu_x, Δu_y) caused by the head movement. Suppose that the initial 3-D coordinate of the head is denoted as(x_0,y_0,z_0), and its position of projection on the image plane is (u_0,v_0).The coordinate of the head is (x', y', z') when head movement occurs. The corresponding parameters R and T are estimated by the AWPOSIT.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + T \qquad (14)$$

Therefore, the displacement (Δu_x,Δu_y) can be calculated by

$$\Delta u_x = f(x'/z') - u_0 \qquad (15)$$

$$\Delta u_y = f(y'/z') - v_0 \qquad (16)$$

The eye vector is extracted and the calibration mapping function is adopted to obtain the gaze point (u_x,u_y) on the screen. Finally combining the gaze direction from the eye vector and the displacement from the head pose, the final gaze point (s_x,s_y) can be obtained with

$$s_x = u_x + \Delta u_x \qquad (17)$$

$$s_y = u_y + \Delta u_y \qquad (18)$$

The implementation procedure has been summarized in below algorithm.

Algorithm 3: Pseudocode Of Eye Gaze Tracking System

```
Initialization:
– Extracting 2-D facial features using ASM
– Initialize the 3-D SHM for the head pose M
– Get calibration mapping
Tracking the gaze through all the frames:
Input from video
Extract the eye region
Detect the iris center p iris
Detect the eye inner corner p corner
Eye vector is obtained: g = p corner −p iris
Get gaze point ($u_x$, $u_y$) using mapping function
Track the face features $P_{2D}$
Obtain the head pose M = AWPOSIT($P_{2D}$, $P_{3D}$, w, f)
Get the displacement ($\Delta u_x$,$\Delta u_y$)
Obtain the final gaze point ($s_x$,$s_y$)
end
```

E. Point Of Gaze Calculation Algorithm

Point of gaze can be referred to as the point of interest of user in Test Area where he/she looking at or gazing at. User's point of interest i.e. PoG can be calculated by extracting some important eye features. Firstly it is necessary to find the reference point. It can be helpful in PoG calculations because fewer calculations will be required to translate pupil movements in eyes into cursor movements on screen. Centre of Eye can act as a desired candidate for reference point. Eyes movable region has already been computed during calibration stage, a simple averaging of x and y co-ordinates can result in Centre of Eye calculation. This is obtained by using equation (19) and (20).

$$COE_x = (TopRightCorner_x + TopLeftCorner_x)/2 \quad (19)$$

$$COE_y = (TopRightCorner_y + TopLeftCorner_y)/2 \quad (20)$$

Where, COE_x and COE_y denote x and y coordinates of center point of eye's movable region respectively. TopRightCorner, TopLeftCorner, BottomRightCorner and BottomLeftCorner construct a rectangular region which represent eye's movable region.

*1) Calculating The Scaling Factor:*This is the first step in point of gaze calculation. In this step the cursor movements and pupil movements were interrelated i.e. it was to be found that how many pixels a cursor will traverse for a single pixel movement of the pupil. For this calculation width and height of eyes were associated with the width and height of the screen. Screen width and height is constant, but eye's movable region width and height is subject to change in different scenarios. Eye's movable region width and height can be computed using Equation (21) and Equation (22).

$$W_{eye} = TopLeftCorner_y - TopRightCorner_y \quad (21)$$

$$h_{eye} = TopRightCorner_y - BottomRigh\,tCorner_y \quad (22)$$

Where, W_{eye} and h_{eye} represent width and height of eye's movable region respectively. Now scaling factor is to be computed for x and y coordinates with help of Equation (23) and Equation (24).

$$R_x = W_{screen} / W_{eye} \quad (23)$$

$$R_y = h_{screen} / h_{eye} \quad (24)$$

Where, W_{screen} and h_{screen} denote width and height of Test Area. R_x and R_y represent scaling factor for x and y coordinates respectively.

*2) Computing The PoG:*This is the final step of POG calculation as well as Gaze Pointer algorithm. This stage will realize the significance of reference point. It translates the pupil movements in eyes into cursor movements in Test Area. Taking assumption that reference point in eye corresponds to center point in Test Area, pupil movements can be simply translated into cursor movements using Equation (25) and Equation (26).

$$PoG_x = (W_{screen}/2) + R_x \times r_x \quad (25)$$

$$PoG_y = (h_{screen}/2) + R_y \times r_y \quad (26)$$

Where, PoG_x and PoG_y represent x and y coordinates of Point of Gaze respectively and r_x denotes pupil distance in the x - direction from reference point and r_y denotes pupil distance in y direction from reference point and they can be computed by using Equation (27) and Equation (28)

TABLE I. EYE MOUSE CONTROL OPERATIONS

RIGHT EYE	LEFT EYE	ACTION
TRUE	FALSE	RIGHT CLICK
FALSE	TRUE	LEFT CLICK
TRUE	TRUE	MOVING MODE

$$r_x = COI_x - COE_x \quad (27)$$

$$r_y = COI_y - COE_y \quad (28)$$

Where, COI represents pupil location.

IV. EVALUATION

An evaluation of eye feature detection, head pose estimation, gaze estimation and point of gaze calculation is presented

A. Eye Center Detection

The detection of the eye center is a difficult task in eye features detection. The accuracy of eye center detection directly affects the gaze estimation.

All the three phases of eye gaze tracking has been done on a video sequence and the corresponding gaze point of each frame has been determined perfectly. Then the result that representing the gaze point of a single frame and its corresponding changes in x and y axis along with average accuracy plot is given in the following figure.

At first the gaze point estimation of a single video frame is shown in the figure below. Then by using this gaze point the corresponding point on the monitor screen are detected and then move the cursor by moving our eyes.

1) Gaze Point

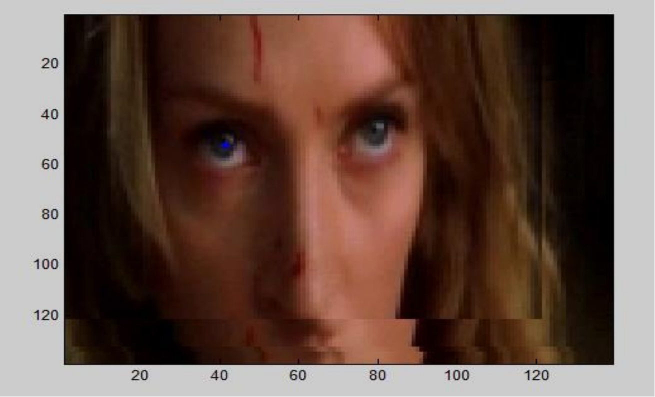

Fig.8. Eye gaze point corresponding to a single frame

2) Head pose estimation Plot: X-direction

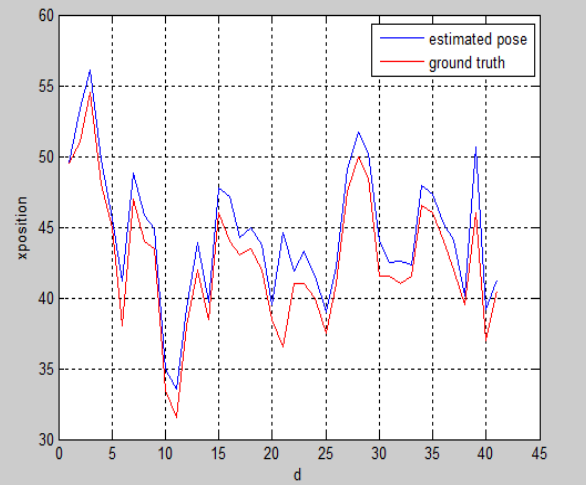

Fig.9. Head pose estimation in X-direction

3) Head pose estimation Plot: Y-direction

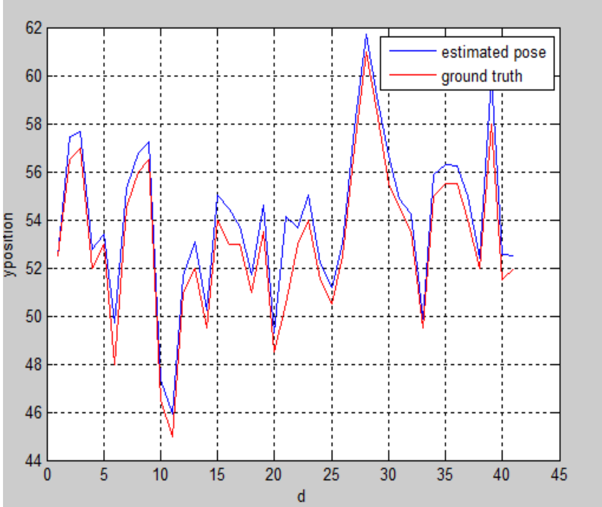

Fig.10. Head pose estimation: Y-direction

4) Average accuracy Plot

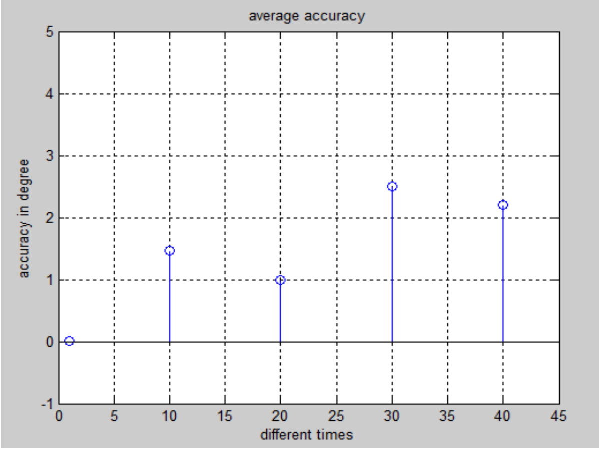

Fig.11. Average accuracy

5) Mouse control operations

TABLE II. MOUSE OPERATION RESULTS

OPERATION	TOTAL FRAMES	TRUE	FALSE
RIGHT CLICK	100	89	11
LEFT CLICK	100	88	12

V. CONCLUSION

A model for gaze tracking has been constructed using a web camera in a desktop environment. Its primary novelty is using intensity energy and edge strength to locate the iris center and eye corner detector to detect the eye vector. Further, have proposed an algorithm to improve the estimation of the head pose. Therefore, the combination of the eye vector formed by the eye center, the eye corner, and head movement information can achieve improved accuracy and robustness for the gaze estimation. After finding the gaze point corresponding to each eye movement our cursor has been moved into different positions on the screen, and is done by point of gaze algorithm. And also the mouse selection means right click, left click, double click can also done in accordance with our eye gaze. The experimental results have shown the efficacy of the proposed method.

VI. ACKNOWLEDGMENT

I am thankful to my project guide Asst. Prof Mrs Jayasree T C and Head of the Department Mrs. Nishidha for their support, remarks and suggestions, which were essential to carry out the survey. I am also grateful to all the staff members of Electronics & Communication Engineering Department of KMCT College of Engineering, Calicut for their assistance and support in improving the review paper significantly. I also thank my family and friends for all the support given by them.

REFERENCES

[1] Yiu-ming Cheung, Senior Member, IEEE, and Qinmu Peng, Member, IEEE" Eye Gaze Tracking With a Web Camera in a Desktop Environment" ieee transactions on human-machine systems, vol. 45, no. 4, august 2015

[2] Muhammad Usman Ghani, Sarah Chaudhry, Maryam Sohail, M. Nafees Geelani" GazePointer: A Real Time Mouse Pointer Control Implementation Based on Eye Gaze Tracking" Journal of Multimedia Processing and Technologies Volume 5 Number 2 June 2014

[3] Chaudhari.Sonali.A, Madur.Neha" Virtual Mouse Using Eye Tracking Technique" International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-4, Issue-2) February 2015

[4] Prajakta Tangade1, Shital Musale1, Gauri Pasalkar1 Miss. Umale M.D., Miss. Awate S.S."A Review Paper on Mouse Pointer Movement Using EyeTracking System and Voice Recognition" International Journal of Emerging Engineering Research and Technology Volume 2, Issue 8, November 2014,

[5] N. Ramanauskas Department of Electronics, Šiauliai University" Calibration of Video-Oculographical Eye-Tracking System" electronics and electrical engineering medicine technology

[6] A. T. Duchowski, "A breadth-first survey of eye-tracking applications,"Behavior Res. Methods, Instrum., Comput., vol. 34, no. 4, pp. 455–470, 2002

[7] E. D. Guestrin and E. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," IEEE Trans. Biomed. Eng., vol. 53, no. 6, pp. 1124–1133, Jun. 2006.

[8] R.Valenti,N.Sebe,andT.Gevers,"Combining head pose and eye location information for gaze estimation," IEEE Trans. Image Process., vol. 21, no. 2,pp. 802–815, Feb. 2012

[9] Ramesh R, Rishikesh M (2015) Eye Ball Movement toControl Computer Screen. J Biosens Bioelectron 6: 181.doi:10.4172/2155 6210.1000181