

# Eye Gaze Optical Mouse using Computer Vision

Dhananjay Mewati  
 School of Computing Science & Engineering  
 Galgotias University  
 Greater Noida, India  
[djmewati@gmail.com](mailto:djmewati@gmail.com)

Mr. S. Jerald Nirmal Kumar  
 Assist. Prof. School of Computing Science & Engineering  
 Galgotias University  
 Greater Noida, India  
[gerald.kumar@galgotiasuniversity.edu.in](mailto:gerald.kumar@galgotiasuniversity.edu.in)

**Abstract**— This paper proposes a technique of using the movement of eyes to control the movement of cursor on monitor screens. Thereby, creating new ways of Human Computer Interaction (HCI) and also helping physically handicapped people to interact with computer devices more efficiently. Earlier eye gaze optical mouse comprised of a head gear which had an eye motion sensor attached and were more hardware based. The input gathered through these sensors helped in cursor movement on screen. With the advancement in the field of Image Processing Techniques and Artificial Intelligence, a simple web camera attached with computer can be used to perform this task. In this paper, pupil of the eye is detected. The coordinates gathered by tracking pupil movement are mapped with the coordinate of display monitor. Based on this mapping the mouse cursor can be moved on the screen.

**Keywords**— eye gaze optical mouse, human computer interaction (HCI), digital image processing, computer vision.

## I. INTRODUCTION

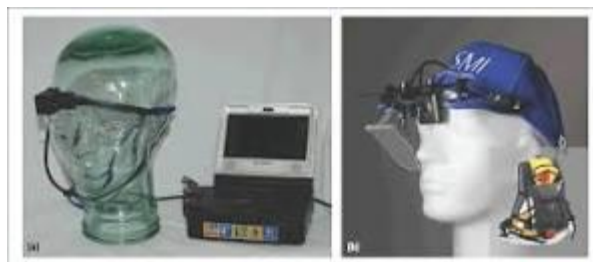
The advancement in technology has led to the creation of new ways of Human - Computer Interaction(HCI). During the last decade a wired mouse and keyboard were the primary sources to accomplish this task. Later on, as the wireless technologies evolved, the wireless mouse, keyboard, speakers and other peripherals came into existence. The development in the field on artificial intelligence lead to the creation of new ways of HCI. like using gestures of hand to control the screen displays.

This paper focuses on one such field of HCI known as eye-gaze optical mouse or eye tracker based HCI. In general, the main focus of these devices is to detect the pupil in the eye of users. Several different approaches are applied so far for detecting the pupil present in the eye. These devices track the movement of eyes [1]. The eye movements lead to the generation of some coordinates. These coordinates are then mapped with the coordinates of the computer's display so as to make the computer interact with the movements of eyes. The mapping of coordinates is done in a way such that if we look at the left side of the computer screen, the mouse cursor would move towards that position.

## II. RELATED WORK

The earlier devices consisted of a head mounted gear. These gears consist of sensors that detect the position of eyes.




The position of eyes is then converted into coordinates. These coordinates are then analysed by the computer to map them with the coordinates on monitor screens. Once the coordinates are mapped, one can easily move the cursor over the screen. IR sensor based which used the corneal based reflection of IR waves were also introduced to improve the accuracy of head mounted eye gaze optical mouse.



[2] Earlier approaches of eye gaze HCI

The major drawbacks situated with these devices is that they are all head mounted systems. Although this type of devices came into existence couple of decades ago, a standard user is still unable to access such devices for daily use. Moreover, they are more hardware based.

To overcome these issues, the concepts of computer vision and image processing can be used. A simple web camera present on the display screen of computer or laptop can be used to accomplish this task. Since, a web camera is always facing the user of computer, the pupil in the eyes of the user facing the camera can be detected and its movement can be tracked. By mapping these tracked coordinates with computer screen, the movement of mouse can be controlled. Also, the blinking of individual eyes can be used for the simulation of left and right mouse clicks.

 Right Eye Wink	Right Click
 Left Eye Wink	Left Click
 Squinting Eyebrows	Activate / Deactivate Scrolling

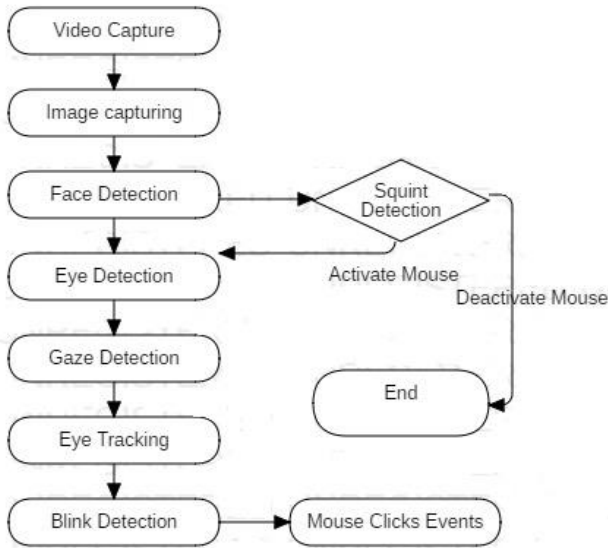
Demonstration of the actions to be performed using eye movements

A similar approach, that uses eye movements for controlling mouse [3]. It used the concept of calculating distance of gaze from the centre of eye to map with the coordinates of monitor screen with an approach of window to viewport mapping. Another paper is such that on looking to the left side of the screen, the mouse cursor moves to the left [4]. The main focus of this approach was on starburst algorithm [5]. The working of all the previous approaches are different from the approach proposed in this paper. This paper is based on the implementation of image processing and computer vision concepts that enable the eye tracking and carry out mouse actions using the web camera present on the computer screen.

### III. METHODOLOGY

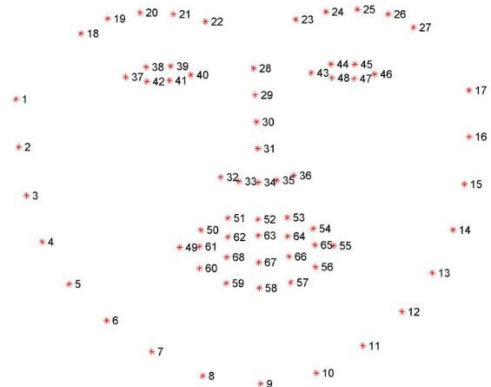
The proposed eye gaze optical mouse is based on the implementation of the concepts of image processing and computer vision. These concepts are applied to enable the eye motion detection through the web camera placed on the computer monitor screen. For this purpose, the python enabled library pyautogui is used. This library helps to get the coordinate points of monitor screen, mouse as well as the eye position.

In this approach, the left eye gaze is primarily used for mouse cursor movement. The implementation of this model is divided into the following -



1. *Image Capturing* – The first step is to capture the image of the user. This is done using the opencv library in python that captures the video of the user [6]. From this video each image frame is captured.
2. *Face Detection* - For this task, first the input image is converted into a monochrome image. Then frontal face detection present in the haar-cascade of opencv is used for fast detection of the face of user.

3. *Eye detection* – The third step is to detect eyes and eyebrows of the user. For this task, the imutils library in python is used that detects all the facial landmarks of the face. Rectangular boxes are used to bound the detected eyes and eyebrows of the user.



[7] Facial landmarks used for eye and eyebrow detection

4. *Gaze detection* – For gaze detection, the extracted images of left is transformed into monochrome image as it the primary eye for mouse cursor movement. Since, pupil is always darker sclera is always white, the gaussian blur filter is used to get the centroid position of the pupil. These coordinates are marked as (x0, y0).

The distance between the facial landmarks 22 and 23 is initialized with the first photo frame of the input video.

5. *Squint detection* – Squinting of eyebrows is used for the activation and deactivation of mouse. The measured distance between facial landmarks 22 and 23 helps to perform this task. The first captured image of user helps to initialize the distance between eyebrows.
6. *Blink detection* – for this purpose the distance between landmarks 39 and 41 is noted for the left eye. For the right eye, 44 and 48 is noted.

Here, we are using XOR operation for winks. When the distance between landmarks 39 and 41 is minimum, the left-wink is considered and left click is performed. Similar action is implemented for the right eye. When both the eyes wink at the same time, no operation is performed.

Left Eye	Right Eye	Action
0	0	No action
0	1	Right click
1	0	Left click
1	1	No action

0 – No wink  
1 – Wink

Truth table representing the actions performed using blink detection

7. *Eye tracking* – The detected coordinated (x0, y0) are used for this task. A simple mapping between monitor coordinates (x1, y1) and (x0, y0) is performed as:

$$x1 = x0 * (Cx/Ex)$$

$$y1 = y0 * (Cy/Ey)$$

Where, (x1, y1) are mapped coordinates of eye (x0,y0) on monitor display. (Cx, Cy) is the maximum size of monitor display and (Ex, Ey) is the maximum size of rectangular bounding box of eye.

#### IV. CONCLUSION

The eye gaze mouse proposed in this paper is implemented using opencv library in python. This model enables the system to detect and track the eye movement using a web camera. Based on the gazing point on the screen, the cursor is moved over the screen. Using the individual blinks of left and right eye, the left and right mouse clicks are simulated on the monitor screens. The system proved to be promising in terms of cursor movements based on eye gaze. The system is cheap to implement and is able to help a majority of physically disabled population to interact with the computer system.

#### V. FUTURE WORK

The model proposed in this paper performs as well and eye movements are performing well to simulate the mouse clicks.

However, the accuracy and speed of eye tracking are not very fast, thus an improvement to increase the responsiveness of eye tracking can be made to make this model more reliable. The implementation of such system in different field like gaming, augment reality, virtual reality and controlling electronic devices are a scope of development of this system.

#### REFERENCES

- [1] Fatima, Ramsha & Usmani, Atiya & Zaheer, Zainab. (2016). Eye movement based human computer interaction. 489-494. 10.1109/RAIT.2016.7507950.
- [2] Andreas Bulling and Hans Gellersen "Toward Mobile Eye-Based Human-Computer Interaction" Published by the IEEE CS, 15361268/10/\$26.00 © 2010 IEEE. October–December 2010.
- [3] Ghani, Muhammad Usman & Chaudhry, Sarah & Sohail, Maryam & Geelani, Muhammad. (2013). GazePointer: A Real Time Mouse Pointer Control Implementation Based On Eye Gaze Tracking. 2013 16th International Multi Topic Conference, INMIC 2013. 10.1109/INMIC.2013.6731342.
- [4] [https://www.academia.edu/40162316/Eye\\_Gaze\\_Movement\\_Detection\\_for\\_Controlling\\_On-Screen\\_Cursor\\_in\\_Real\\_Time](https://www.academia.edu/40162316/Eye_Gaze_Movement_Detection_for_Controlling_On-Screen_Cursor_in_Real_Time)
- [5] Dongheng, Li, and J. Derrik Parkhurst. "Starburst: A robust algorithm for video-based eye tracking." Proceedings of the IEEE Vision for HumanComputer Interaction Workshop. 2005.
- [6] "OpenCV," [Online]. Available: <https://opencv.org/about/>. [Accessed april 2, 2020].
- [7] "pyimagesearch," [Online]. Available: <https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/>. [Accessed April 2, 2020].