International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 08, 2020
ISSN: 1475-7192

EYE TRACKING AND VOICE CONTROL FOR COMPUTER APPLICATION

[1]Pindikanti Vivek, [2]Akshay Singh, [3]Mrs. A.L. Amutha

***Abstract** -- The concept of digital image processing and speech recognition can be used for tracking the eye movement which can help in controlling the mouse on the screen and performing the specified function using voice commands. To track eye a macro lens camera is used, which will be placed close in front of the user's eye. A human eye has 3 visible regions – Sclera, Iris and Pupil. Sclera is white in color, color of Iris varies from person to person, but Pupil is black in color. This allows us to track the pupil's movement on the eye. The second camera will be positioned on the user's head which will track the screen edges and determine the distance between user and screen, it will also track the head rotation to determine the exact cursor position. The pupil's position can be then mapped on to the screen so that the mouse moves relative to the eye. The mapping is done by resizing the screen frame to fit the eye frame and tracking the center of the eye relative to the screen edges. After finding the cursor coordinates it is important to remove noise. Noise can be in the form of non-continuous cursor movement or cursor moving back and forth even if the user is idle. The voice recognition system will process the user's voice and converts it into text which can be used as commands for computer operation. These operations involve clicking, double-clicking, right-clicking, typing, etc. The main idea of this project is to minimize the number of input devices and increase the ease of human computer interaction. It will be useful for people who are suffering from amputation or who want an ergonomic work space, as it minimizes the use of hands for interacting with the computer.*

***Keywords** -- Digital Image Processing, Voice Recognition, Human Computer Interaction*

I. INTRODUCTION

Eye tracking is a sensor technology which enables a device to know where exactly a person's eye is focused. It determines one's presence, attention, focus, drowsiness, conscious-ness or other mental states. This information can be used to gain insights into consumer behavior or to design revolutionary new user interfaces across various devices. Eye-tracking analysis is also a research tool which is used to measure visual attention. Eye-tracking can also be used for measuring physiological responses to visual stimuli which has the advantage that responses can be recorded in real time. Speech recognition is a subfield of computational linguistics (CL) which helps in developing technologies and methods that enables the translation as well as recognition of the spoken language into text by computers. It is also known as Speech to Text (STT), computer speech recognition or Automatic Speech Recognition (ASR). It also incorporates knowledge and research in the linguistics, computer science and engineering, and electrical engineering fields. Voice recognition is a software program or a hardware device programmed with certain specification so as to convert human voice into text. Voice recognition is generally used to perform certain commands, operate a device, or write without having to use a, mouse, keyboard or press any buttons. Eye tracking and voice recognition as a combination will ease human computer interaction, help people who suffer from amputation and provide an ergonomic work environment.

Tobii Eye Tracker is an existing system which is used for eye tracking. It consists of 2 cameras which are used to track the pupil position. By combining eye tracking with other input modalities, for example, keyboard, mice, joystick, touchpad and voice commands, it is possible to create more intuitive user experiences. The use of high-performance sensor and camera requires high computational power (minimum 2.4Ghz quad core processor). It only works on

[1] Dept. of CSE, SRM IST, KTR Campus
[2] Dept. of CSE, SRM IST, KTR Campus
[3] Dept. of CSE, SRM IST, KTR Campus

Received: 27 Mar 2020 | Revised: 20 Apr 2020 | Accepted: 05 May 2020

screens which support pixel shader 2.4 or higher. These requirement makes it expensive ($250 to $300) and out of reach for common people.

As per the 2011 Census, in India, out of the 121 Cr population, about 2.68Cr persons are 'disabled' which is 2.21% of the total population. The five types of disabilities on which data has been collected are: speech (7.5%), mental (10.3%), movement (27.9%), sight (48.5%), and in hearing (5.8%). More than 50% of the disabled are below the poverty line. The main objective of this project is to minimize the number of input devices, increase the ease of human computer interaction, reduce the overall hardware cost, make work place ergonomic, make eye tracking faster and decrease the delay due to lag, reduce the calibration time for mapping the eye to screen.

II. SYSTEM DESCRIPTION

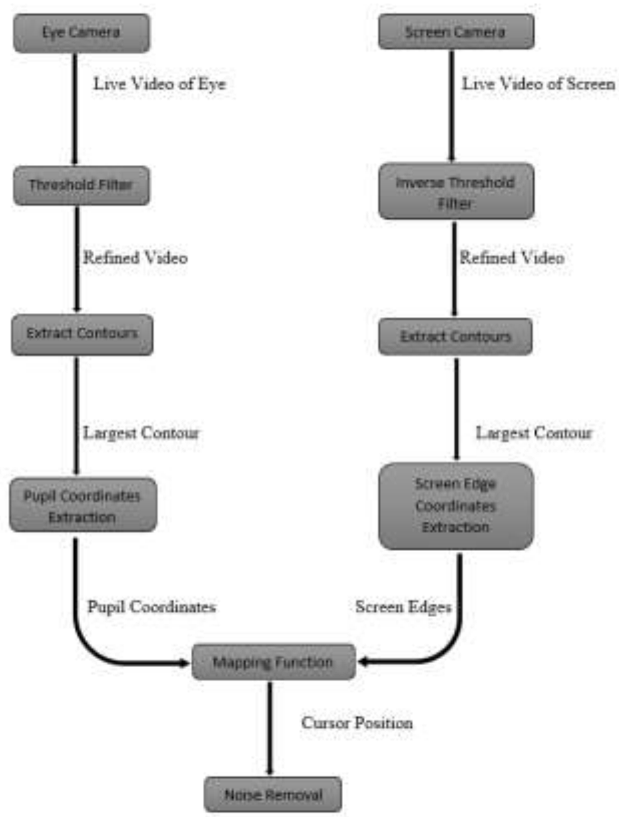


Fig. 1. ARCHITECTURE DIAGRAM

The architecture diagram is represented in Fig. 1. There are 6 major modules of the system namely Video Capture Module, Threshold Check Module, Contours Module, Calibration Module, Noise Removal Module, Voice Command Module.

1. MODULES

A. Video Capture Module

This module Captures the video sent by the eye and screen camera. Then it divides the video into frames each containing the RGB values of every pixel in a list format. Then each frame is rotated to a certain angle so that the eye's horizontal movement matches with the cursor's horizontal movement. Each frame can then be processed for extracting the location of pupil and screen edges. This module also flips the screen camera's frame horizontally and vertically to counteract the head movement.

B. Threshold Check Module

This module converts each frame to gray. Then the frame is blurred to a point where only pupil is visible as total black. Threshold value is selected by changing the grey value and observing the accuracy. The value which provides higher precision in tracking is taken as the final value. Then threshold is applied to the blurred frame. This module converts the gray frame to black and white frame. This makes it easier to identify the pupil and screen.

C. Contours Module

This module finds the contours in threshold frame (i.e. areas with white color). Contours are the edge coordinates of each white block which is then used to calculate the area. Then the contour with largest area is selected by sorting the area in the descending order. The center of this contour will be returned which the reference point for tracking the eye. For screen edges, the edge points are selected from the largest contour.

D. Calibration Module

This module calibrates the pupil coordinates and the screen edge coordinates together. Calibration is done by getting the eye coordinates when the user is looking at the edges of the screen. Then these edge coordinates are used as the reference point for 4-point calibration. At every frame the screen camera will give the coordinates of screen edges which are then passed on to the thread handling the eye camera. The screen edges are then mapped to the eye camera to get the position of cursor relative to the screen.

E. Noise Removal Module

This module gets the calibrated cursor coordinates and then move the cursor accordingly. This module also tries to reduce the noise in cursor movement so that the cursor movement feels smoother and sturdier. This is done by delaying the cursor and reading its position before pointing the cursor to the specified coordinate, then it compares the cursor position between 2 frames, if the cursor movement is very less between the 2 frames then it will ignore the frame and move on to the next frame.

F. Voice Command Module

This module listens to user's speech and send the audio to google speech recognition API. The API returns the text corresponding to user's speech. The text is used to trigger commands like click, double-click, right-click, type, etc. The type command listens to user's speech and type it as text using the same API. This module can eliminate the need of keyboard.

2. EQUATIONS

A. Screen Edge Detection

The screen edge is calculated from the largest contour. A contour holds the list of all white pixels whose adjacent pixel is black (black and white frame is obtained from the threshold filter). So, the largest contour contains all the boundary pixels of the screen. Each pixel has 2 values the X-axis coordinate and the Y-axis coordinate. To find the top-left corner of the screen, each pixel is compared with top-left corner of the frame i.e. (0,0). Whichever pixel has the least distance from the frame corner is selected as the top-left corner of the screen. The same process is applied to find each corner.

$$min(\sum_{x_1,y_1}^{x_n,y_n}(x_i - x + y_i - y))$$

B. Noise Removal

Noise can be in the form of non-continuous cursor movement or cursor moving back and forth even if the user is idle. To eliminate this, we have to delay the cursor movement by 2 frames so that we can identify if there is any noise in the cursor movement and eliminate it. When the cursor is moving back and forth the first and third position of cursor are same and the second position is different. After identifying this pattern, we can ignore the second frame so that the cursor remains idle.

$$if(x_1 == x_3 \cap y_1 == y_3) \Rightarrow (x_2 = x_1, y_2 = y_1)$$

3. HARDWARE SPECIFICATIONS

A. Eye Camera

The eye camera is used to get the video feed of the eye. This camera is placed just below the user's eye. The eye camera has a macro lens which makes it easier to focus on the eye from a very short distance.

TABLE I. EYE CAMERA SPECIFICATIONS

Resolution	1920x1080
FOV	120°
FPS	60
Connection	USB 2.0

Fig. 2. EYE CAMERA VIEW

B. Screen Camera

The screen camera is used to get the video feed of the screen. This Camera is placed on the user's head and faced towards the screen. This camera will track the user's head movement.

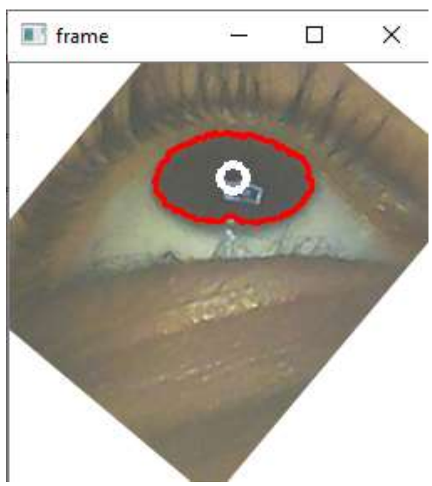
TABLE II. SCREEN CAMERA SPECIFICATIONS

Resolution	1280x720
FOV	60°
FPS	30
Connection	USB 2.0

Fig. 3. SCREEN CAMERA VIEW

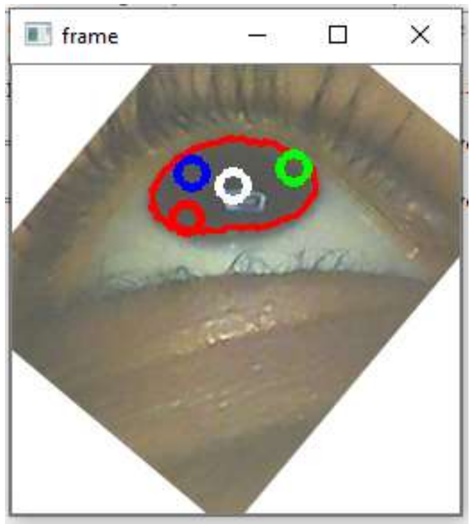
III. RESULTS

The algorithm developed in this project finds the pupil position and the screen corners, then the algorithm will map both of them together to get the cursor position. The cursor position is then passed through a noise removal filter so that the final cursor feels sturdier and smoother. The result for finding the pupil position is shown in Fig. 4(a). The result is obtained by processing Fig. 2. The result for finding the screen corners is shown in Fig. 4(b). The result is obtained by processing Fig. 3. The result for mapping screen corners to pupil is shown in Fig. 4(c). The red, blue and green circles represent the screen corners and the white circle represents the eye's center.



(a)

(b)



(c)

Fig. 4. RESULTS

The test was conducted on a computer whose description is given in Table III. The first test was to find the delay between the eye movement and the cursor movement. This is done by recording the screen with a very high framerate video recorder and comparing each frame with the corresponding eye camera recording. The time taken by the computer to process a frame and move the cursor was 17.2ms average. The next test was to find the speed at which the video is processed. In other words, we are calculating how many frames are processed in 1 sec, it is called FPS (Frames Per Second). The frames processed each second were ranging in between 57 and 60 with an average of 58 FPS. The final test was to find the accuracy of eye tracking. To find the eye tracking accuracy, a continuous video of program was recorded. In this video each frame was checked to find whether the program was successful in tracking the eye or not. The accuracy achieved from this test was 96%.

TABLE III. COMPUTER SPECIFICATIONS

Processor	Intel i7 – 4790k
Clock Speed	4.4Ghz
Cores and Threads	4 Cores 8 Threads
GPU	Nvidia GTX 970
V-RAM	4 GB
RAM	16 GB

IV. CONCLUSION

In this system interfacing of the eye was done with cursor by using two different cameras, for computer screen and eye. The coordinates captured from the screen camera were mapped to eye camera by using screen to eye ratio which was found to be 1:10 for x-axis and 1:12 for y-axis. A microphone was used for recording user’s voice and triggering the action based on his/her voice command. From the results found after testing this system we can conclude that the system has achieved a fast and responsive eye tracking technique. Although the results look promising, there is a requirement for a bright video of eye. The accuracy of eye tracking decreases if the light is insufficient. The initialization time for mapping eye to screen is also decreased significantly. The project was done under a budget which is significantly less than the cost of existing system. This system has the potential to eliminate the use of mouse and keyboard.

V. FUTURE SCOPE

This system can be used in almost all computer applications. The requirement for a bright environment can be fulfilled by a light source. We can also use an infrared light, but so can result harmful to eyes. A technology called "Night Mode" can also be used, "Night Mode" is used by a few smartphone companies in their phones so that user can take picture in a dark environment. The technique behind this technology is that the camera takes a picture with very high ISO i.e. the camera lens gets light for a long period of time and then generate a picture. Then this picture is used with a dark picture taken by the user to convert it to a bright photo. This technology can record video in dark environment without the use of light source. This system has the potential to provide an ergonomic work place especially for the people suffering from amputation.

REFERENCES

[1] Veena N. Hegde, Ramya S. Ullagaddimath, S. Kumuda: "Low cost eye based human computer interface system" - 2016 IEEE Annual India Conference (INDICON).
[2] Tibor Ujbányi: "Examination of eye-hand coordination using computer mouse and hand tracking cursor control" - 2018 9th IEEE International Conference on Cognitive Infocommunications.
[3] Prashant Salunkhe, Ashwini R. Patil: "A device controlled using eye movement" - 2016 International Conference on Electrical, Electronics, and Optimization Techniques.
[4] Z. A. Sasinas Alias Haritha, P. V. Raveena, K. S. Arun, P. N. Nithya, M. Vishnu Balan, Sabareesh Krishnan: "Eye tracking system using isophote eye center detection with blink perception" - 2016 International Conference on Signal Processing, Communication, Power and Embedded System.
[5] Pawel Kasprowski, Katarzyna Harezlak, Michał Niezabitowski: "Eye movement tracking as a new promising modality for human computer interaction" - 2016 17th International Carpathian Control Conference.
[6] Eun Yi Kim, Sin Kuk Kang, Keechul Jung, Hang Joon Kim: "Eye mouse: mouse implementation using eye tracking" - 2005 Digest of Technical Papers. International Conference on Consumer Electronics, 2005. ICCE.
[7] Robert Gabriel Lupu, Florina Ungureanu, Valentin Siriteanu: "Eye tracking mouse for human computer interaction" - 2013 E-Health and Bioengineering Conference.
[8] Shahram Jalaliniya, Diako Mardanbeigi; Thomas Pederson, Dan Witzner Hansen: "Head and Eye Movement as Pointing Modalities for Eyewear Computers" - 2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops.
[9] Tomasz Kocejko, Adam Bujnowski, Jerzy Wtorek: "Eye mouse for disabled" - 2008 Conference on Human System Interactions.
[10] Chairat Kraichan, Suree Pumrin: "Face and eye tracking for controlling computer functions" - 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology.