JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Human Computer Interaction Based Eye Controlled Mouse

Mr. D. Srinivas Goud
Associate Professor, CSE
Vignana Bharathi Institute of Technology, Hyderabad.
goud@vbithyd.ac.in

G. Rakesh
Bachelor of Technology, CSE
Vignana Bharathi Institute of Technology, Hyderabad.
rakeshgaini1405@gmail.com

L.Suraj
Bachelor of Technology, CSE
Vignana Bharathi Institute of Technology, Hyderabad.
surajloni1122@gmail.com

K. Hemanth Reddy
Bachelor of Technology, CSE
Vignana Bharathi Institute of Technology, Hyderabad.
reddyhemanth6992@gmail.com

Abstract:

In today's digital age, advancements in technology continue to create opportunities for innovation in computing. There is a growing demand for hands-free computing, particularly to assist quadriplegics. This paper introduces a Human Computer Interaction (HCI) system designed to cater to the needs of amputees and individuals with hand-related impairments. The system utilizes an eye-based interface to function as a computer mouse, translating actions such as blinking, gazing, and squinting into cursor movements. It relies on a basic webcam and requires software components including Python, OpenCV, numpy, and other essential packages for facial recognition. The facial detection aspect of the system utilizes the Histogram of Oriented Gradients (HOG) feature in conjunction with a linear classifier and the sliding window technique. Notably, this system operates without the need for external hardware or sensors, offering a hands-free solution for users.

Keywords—: Python (3.6), Open CV, Human computer interaction, numpy, face recognition,

Histogram of Oriented Gradients (HOG), SVM

I. INTRODUCTION

In today's technology, the conventional method for moving the cursor on a screen involves using a computer mouse or finger movements. This system detects the movement of the mouse or finger to correspond it with the cursor's movement. However, individuals without functional arms, such as amputees, are unable to utilize this technology effectively.

To address this limitation, tracking the movement of the eyeball and determining the direction of the user's gaze can offer a solution. By mapping the movement of the eyeball to the cursor, individuals like amputees can control the cursor effortlessly.

The concept of an 'eye tracking mouse' holds significant promise for amputees. Currently, this technology is not widely available, with only a few companies having developed and released such products. Our goal is to develop an eye tracking mouse that replicates most functions of a traditional mouse, allowing users to control the cursor with their eyes.

We aim to accurately estimate the user's gaze direction and move the cursor accordingly. While pointing and clicking actions have long been standard with traditional mice, some individuals may find them uncomfortable or, in the case of those unable to move their hands, inaccessible. This underscores the importance of developing hands-free mouse alternatives that rely on eye and facial movements for control.

II. LITERATURE SURVEY

Generally speaking, eye tracking tracks an individual's gaze direction and eyeball location. Various technologies can be used to track an individual's eye movements. Four categories can be used to group it: scleral search coil technique and infrared-oculography (IROG) (VOG), electrooculography (EOG), and super-spatial coherence (SSC). At the moment, the majority of eye tracking studies in HCI are based on VOG since it has reduced user intrusiveness to some extent.

In order to create a cursor control system for computer users, Chin et al. combined two inputs: point-of-gaze coordinates generated by an eye-gaze tracking system and electromyogram signals from facial muscles. Even though it might make a trustworthy click operation, it was less accurate and slower than the control system that merely used eye tracking. Missimer and Betke created a system that mimics left and right mouse clicks by blinking left or right monocularly while using the head position to control the mouse pointer. This system used the user's head position to determine where to point the mouse pointer. Erroneous head movements can impact the precision of the click operation. A communication system for those with disabilities was proposed by Lupu et al. which used a specially created gadget consisting of a webcam put on a spectacles frame for picture processing and acquisition. The apparatus detects the eye movement, and the voluntary Eye blinking is associated with a chosen symbol or keyword that best represents the demands of the patient. The system's shortcoming is that the image processing algorithm is not robust to light intensity and is unable to reliably identify the obtained image of low quality. Subsequently, they suggested an eye tracking mouse system using video glasses and a new, robust eye tracking algorithm based on the adaptive binary segmentation threshold of the captured images in order to increase the dependability of the communication system.

Recently, researchers have also developed a number of comparable systems. The basic idea behind these systems is to take pictures using a camera that is either remotely placed or worn on the user's headgear, and then extract data from various eye attributes to ascertain the gaze's point. All of the above-mentioned eye tracking control solutions were suggested using custom hardware and software because the commercial eye trackers were too costly to utilize in HCI. The closed-source nature of the hardware and software made it difficult for these systems to become widely used. 5. This can be accomplished using a variety of techniques. Margrit Betke et al.[1] proposed the camera mouse as a tool for nonverbal quadriplegics. A camera tracks the user's movements, which can then be linked to the movements of the mouse pointer that's displayed on the monitor. Robert Gabriel Lupu, et al.[2] presented yet another approach to human-computer interaction that used a head-mounted device to capture eye movement and convert information onto a screen. Another method for eye tracking utilizing the Hough transform is presented by Prof. Prashant Salakhe et al. [3]. Many efforts are being made to enhance HCI's features.A study by Ghani, Muhammad Usman, and others [4] implies that the user may access interfaces without the usage of any additional hardware, such as a mouse or keyboard, by using the eye's movements as an input[5]. In order to do this, image processing methods and computer vision. The Haar cascade characteristic can be used as one method of eye detection.According to Vaibhav Nangare et al. [6], the eyes can be identified by comparing it with templates that are already saved. An infrared sensor can be used to obtain an accurate image of the iris. The orientation of the head can be determined using a gyroscope, as recommended by Anamika Mali et al. [7]. One way to perform the click operation is to "gaze," or fix your gaze on the screen. Additionally, Zhang et al.'s suggested implementation of the scroll function can be carried out by looking at a piece of either the upper or lower portion of the screen [8]. Together with eye movements, it gets simpler. If we additionally include a few delicate facial gestures. The blink action can be recognized using facial landmarks, as described by Tereza Soukupova and Jan ´ Cech [9] in a real-time eyeblink detection system. This is important since it takes blinking motions to convert it to clicking motions. OpenCV with Python with dlib can be used to detect eyes and other face components [10].

In a similar manner, blinks can be identified.Christos Sagonas et al.'s research addresses the difficulties associated with facial landmark localization. In a similar manner, Akshay Chandra suggests using the mouse cursor. making facial gestures.

III. METHODOLOGY

The project's methodology commences with the utilization of the webcam to access video frames, initiating a process wherein each frame undergoes individual processing to identify specific facial features such as the eyes, mouth, and nose. This process occurs in a continuous loop, with frames being extracted at a frequency of approximately 30 frames per second. The primary objective is to establish a connection between these detected facial features and cursor movements, essentially enabling hands-free control of the mouse cursor. This functionality is particularly beneficial for individuals with physical disabilities, as it eliminates the need for conventional mouse input methods. By leveraging eye movements as a means of controlling the mouse pointer, the system aims to enhance accessibility for users with limited or no hand functionality, thereby promoting inclusivity in computer usage. Additionally, the system incorporates real-time eye tracking and gaze estimation through eye-based human-computer interaction, facilitating seamless interaction with the computer interface solely through eye movements. Furthermore, the system is designed to simulate various mouse functions, including left-click, right-click, scrolling, and more, thereby offering a comprehensive handsfree computing solution that caters to diverse user needs and preferences. This approach not only addresses the specific requirements of individuals with disabilities but also contributes to advancing the field of human-computer interaction by introducing innovative and inclusive computing solutions.

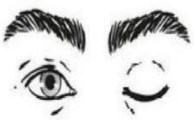
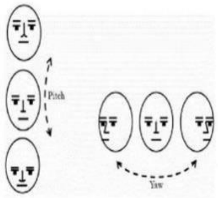
Action	Function
 Opening Mouth	Activate / Deactivate Mouse Control
 Right Eye Wink	Right Click
 Left Eye Wink	Left Click
 Squinting Eyes	Activate / Deactivate Scrolling
 Head Movements (Pitch and Yaw)	Scrolling / Cursor Movement

Fig. 1. Mouse Cursor Movements

A. Webcam Access:

The project's first step involves accessing the webcam to facilitate the capture of video frames. This pivotal task ensures the continuous flow of visual data necessary for subsequent processing. The webcam is initialized, enabling the extraction of frames at a consistent rate of approximately 30 frames per second. This steady stream of frames serves as the raw material for further analysis and manipulation within the system.

B. Image Processing:

Following the extraction of each frame, the system applies a series of image-processing functions to enhance its quality and facilitate the detection of facial features. These functions are crucial for optimizing the frame in preparation for feature detection. By highlighting and refining regions of interest within the frame, such as the eyes, mouth, and nose, the system ensures greater accuracy in subsequent analyses. This meticulous process of image enhancement lays the groundwork for the system's functionality in accurately detecting and mapping facial features to cursor movements, thus enabling precise and responsive interaction with the user interface.

The project's methodology was informed by a detailed exploration of software development requirements, distinguishing between functional and non-functional aspects, and undertaking a thorough requirement analysis in both system and software engineering domains to accommodate the varied needs of stakeholders. Functional requirements, a cornerstone in software and system engineering, delineated the fundamental functions of a system or component, encompassing inputs, behaviours, and outputs.

These encompassed a range of tasks critical to the project's objectives, including image resizing, converting images from BGR to grayscale, and detecting and predicting facial features through the utilization of prebuilt models. For instance, the system harnessed a prebuilt model for face detection, complemented by a 'predictor' function to pinpoint facial landmarks, subsequently storing them as coordinates for subsequent use. Integral to the project's success were the establishment of ratios for eye and mouth aspect ratios (EAR and MAR), pivotal for detecting actions such as blinking and yawing, and translating them into mouse actions.

The system further delineated specific actions performed by the face, including activating the mouse, left/right clicking, and scrolling, predicated on predefined thresholds and ratios. For instance, yawing (mouth opening vertically) triggered cursor activation, while closing one eye signified left or right clicking, and simultaneous squinting of both eyes facilitated scrolling. These functionalities were meticulously implemented using conditional statements and ratio comparisons within the program, with EAR and MAR being computed using the provided formulas. Ultimately, the project aspired to develop an inclusive system for hands-free cursor control predicated on facial movements, adeptly addressing diverse user needs and interactions within the digital environment.

These functionalities were implemented using conditionals and ratio comparisons within the program, with EAR and MAR calculated using the following formulas:

$$EAR=||p2-p6||+||p3-p5||/||p1-p4||$$

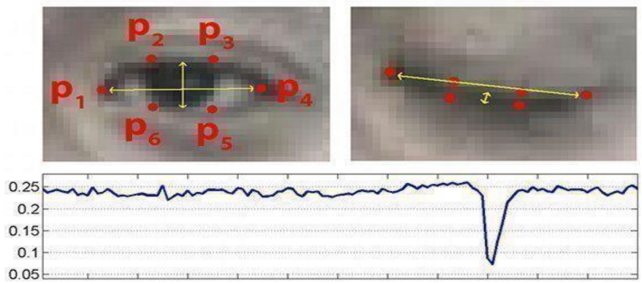


Fig 2: Eye-Aspect Ratio

MAR = MAR=||p2-p8||+||p3-p7||+||p4-p6||/||p1-p5||

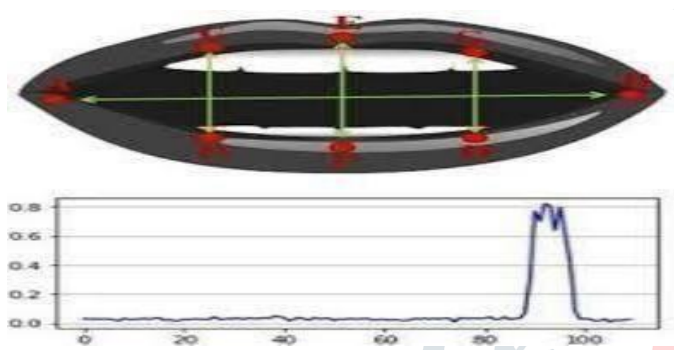


Fig 3: Mouth-Aspect Ratio

IV. SYSTEM DESIGN

System design is the process of transitioning from user-oriented documentation to instructions for programmers or database personnel. It serves as a blueprint for creating a new system and consists of multiple steps aimed at providing a clear understanding and procedural details for implementing the system outlined in the feasibility study. Designing involves both logical and physical stages of development. In the logical design phase, the current physical system is reviewed, input and output specifications are prepared, implementation plans are detailed, and a logical design walkthrough is conducted. Database tables are designed by analysing the functions involved in the system, and the format of fields is determined. Each field in the database tables must define its role in the system, and unnecessary fields should be avoided to prevent unnecessary storage consumption. Additionally, in input and output screen design, the focus is on creating user-friendly interfaces with precise and compact menus.

System architecture

System architecture encompasses the overarching structure of a software system and the methodology involved in creating and organizing such structures. It involves defining the high-level organization of software elements, the relationships between them, and the properties of these elements and relationships. Drawing an analogy to architectural blueprints for buildings, system architecture serves as a comprehensive plan or roadmap for the software system and the entire development project. It outlines the necessary tasks and considerations for design teams to undertake during the development process.

Designing system architecture entails making critical decisions regarding the fundamental structural aspects of the software system. These decisions are significant because they can be difficult and costly to alter once the system is implemented. Software architects must carefully evaluate various design options and select the most appropriate structural choices based on the project requirements, constraints, and objectives.

A software architecture is not just a static representation of the system's structure; it also defines its behaviour and functionality. It provides a conceptual model that encompasses multiple perspectives or views of the system, including its structural, behavioural, and operational aspects. These views help stakeholders, including developers, designers, and project managers, to understand different aspects of the system and collaborate effectively throughout the development lifecycle.

Overall, system architecture plays a crucial role in ensuring the success of software projects by providing a clear and coherent framework for designing, implementing, and managing complex software systems. It serves as a foundation for the development process, guiding teams in making informed decisions and facilitating communication and collaboration among stakeholders.

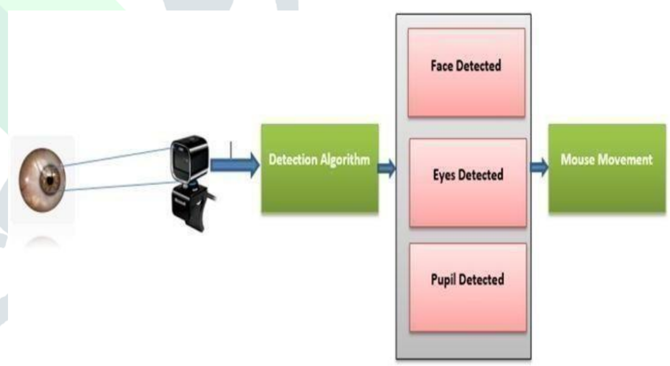


Fig 6: System Architecture

Technical Architecture

Technical architecture, as the system architecture layer of a computer, encompasses the framework that establishes and governs the interactions, arrangement, and regulations of the product architecture. It is responsible for ensuring that the system effectively executes tasks as defined in the programming. In essence, technical architecture serves as the foundational structure that defines how various components within the system communicate and operate to achieve the desired functionalities. It involves specifying the hardware, software, protocols, and interfaces necessary to support the system's operations. This layer of architecture plays a crucial role in determining the system's performance, reliability, scalability, and security by establishing the underlying infrastructure and mechanisms required for seamless operation. Overall, technical architecture forms the backbone of the system, providing the necessary framework to support and optimize its functionality while adhering to predefined rules and specifications.

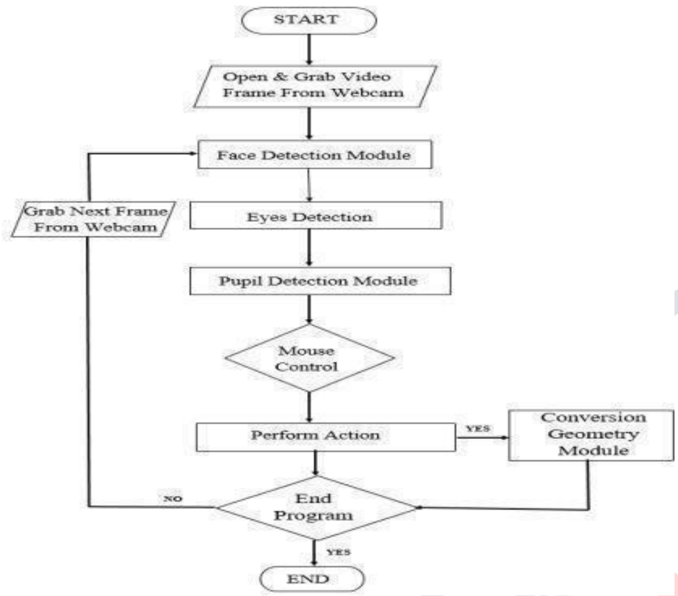


Fig 6: Technical Architecture

V. RESULTS AND DISCUSSION

In The advancement of technology in the digital era continually presents opportunities for computing development, particularly in addressing the needs of individuals with physical limitations such as quadriplegics and amputees who face challenges in using traditional input devices like computer mice. An innovative Human-Computer Interaction (HCI) system has been proposed to cater to these needs by utilizing an eye-based interface that translates eye movements, including blinking, gazing, and squinting, into mouse cursor actions. This system, which requires only a simple webcam and software such as Python, OpenCV, and NumPy, aims to provide hands-free computing capabilities without the need for external hardware or sensors.

The proposed eye-tracking mouse offers a viable solution for individuals with limited or no use of their hands, enabling them to control the cursor using their eyes. Currently, such eye tracking technology is not widely available, but efforts are being made to develop a comprehensive eye-tracking mouse that replicates most mouse functions. By estimating the user's gaze direction, the system aims to accurately move the cursor in alignment with the user's eye movements, thereby facilitating seamless interaction with the computer interface.

In addition to addressing the need for hands-free computing, the system design process plays a crucial role in ensuring the successful implementation of the proposed HCI system. System design involves transitioning from user-oriented documentation to detailed instructions for programmers and database personnel. This process encompasses several steps, including logical and physical design stages, input and output specification, implementation planning, and user-friendly interface design. The design of database tables is also critical, with careful consideration given to the functions involved in the system and the format of fields to optimize storage efficiency.

Furthermore, the architecture of the system plays a fundamental role in shaping its structure, behavior, and functionality. System architecture refers to the high-level structures of a software system, encompassing software elements, their relationships, and properties. It serves as a blueprint for the development project, guiding design teams in making fundamental structural choices that are costly to change once implemented. Software architecture choices include specific structural options that define the conceptual model of the system, including its structure, behaviour, and various views.

Moreover, technical architecture, as the system architecture layer of a computer, defines and controls the exchanges, organization, and rules of the product architecture to ensure the system performs tasks effectively in programming. It establishes the foundational framework necessary for seamless operation, including hardware, software, protocols, and interfaces, to support the system's functionalities.

Fig 8:

VI. CONCLUSION

In conclusion, the development of the proposed Human Computer Interaction (HCI) system that utilizes eye-based interaction presents a significant advancement in addressing the needs of individuals with physical limitations, particularly those who are unable to use traditional input devices such as computer mice. By leveraging intuitive eye movements, this system offers a promising solution for enabling hands-free computing, thereby enhancing accessibility and usability for users with disabilities such as quadriplegia or amputations. The integration of eye-tracking technology into the HCI system demonstrates a pioneering approach to interface design, allowing users to control the cursor and interact with digital interfaces using natural eye movements.

Furthermore, the success of implementing this innovative technology relies heavily on the systematic design process and the architectural framework of the system. The system design process involves meticulous planning and consideration of user requirements, interface specifications, and technical constraints. By following a structured design approach, development teams can ensure that the HCI system meets user needs effectively while maintaining compatibility with existing software and hardware environments. Additionally, the architectural framework of the system serves as a blueprint for organizing and optimizing its structure, behaviour, and functionality. This framework guides development teams in making informed decisions regarding software elements, relationships, and properties, ensuring seamless integration and operation of the HCI system.

Overall, the proposed HCI system represents a significant step forward in advancing assistive technology for individuals with physical disabilities. Through the integration of eye-based interaction, the system offers a versatile and intuitive means of accessing and interacting with digital devices and applications. Moreover, by emphasizing the importance of systematic design and architectural considerations, development teams can ensure the successful implementation of this innovative technology, ultimately enhancing accessibility and usability for users with physical limitations.

VII. REFERENCES

[1] In Encyclopedia of Human-Computer Interaction, Alex Poole and Linden J. Ball, "Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects" (30 December 2005) CiteUlike: 3431568, 2006, pp. 211-219, is the key.

[2] "Non-contact Eye Gaze Tracking System by Mapping of Corneal Reflections," Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR02), 2002, pp. 94-99, D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung.

[3] "System for assisted mobility using eye movements based on electrooculography," Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena Lpez, IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, vol. 10, no. 4, pp. 209–217, DECEMBER 2002.

[4] "A review on electrooculography" by H. Singh and J. Singh was published in the International Journal of Advanced Engineering Technology in 2012 (vol. III, no. IV).

[5] "A laser-based eye-tracking system," by K. Irie, B. A. Wilson, and R. D. Jones Actions Vol. 34, no. 4, pp. 561-572, Research Methods, Instruments, & Computers, 2002.

[6] In Pattern Recognition, 1992, Vol. I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on, 1992, pp. 407-410, P. Ballard and George C. Stockman, "Computer operation via face orientation."

[7] "Computing 3-D head orientation from a monocular image sequence," by T. Horprasert, Y. Yacoob, and L.S. Davis, in Second International Conference on Automatic Face and Gesture Recognition, 1996, pp. 242-247.

[8] K. Arai and M. Yamaura, "Working in a Real-Time Basis without Calibration: Computer Input with Human Eyes-Only Using Two Purkinje Images," CSC Journals, vol. 1, no. 3, pp. 71–82, 2010.

[9] D. Back, "Neural Network Gaze Tracking using Web Camera," 2005 Master's Thesis, Linkping University.

[10] R. Gonzalez and R. Woods, Pearson Education, 2009, Digital Image Processing, Third Ed.

[11] P. Viola and M. Jones, "Using a Boosted Cascade of Simple Features for Rapid Object Detection," in COMPUTER VISION AND PATTERN RECOGNITION, 2001.