IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 October 2024, accepted 11 November 2024, date of publication 14 November 2024, date of current version 22 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3498441

RESEARCH ARTICLE

Revolutionizing Gaze-Based Human–Computer Interaction Using Iris Tracking: A Webcam-Based Low-Cost Approach With Calibration, Regression and Real-Time Re-Calibration

GOVIND RAM CHHIMPA[1], **AJAY KUMAR**[2], **SUNITA GARHWAL**[2], **DHIRAJ**[3], **FAHEEM KHAN**[4], **AND YEON-KUG MOON**[5]

[1]Department of Internet of Things and Intelligent Systems, Manipal University, Jaipur, Rajasthan 303007, India
[2]Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab 147001, India
[3]CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan 333031, India
[4]Department of Computer Engineering, Gachon University, Seongnam-si 13120, South Korea
[5]Department of Artificial Intelligence and Data Science, Sejong University, Seoul 05006, South Korea

Corresponding authors: Faheem Khan (faheem@gachon.ac.kr) and Yeon-Kug Moon (ykmoon@sejong.ac.kr)

This work was supported by the Culture, Sports, and Tourism Research and Development Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports, and Tourism, in 2024, Project Name: Development of AI-Based Content Production Technology to Advance Volumetric Studio and Virtual Production, under Project RS-2024-00351961.

ABSTRACT Eye movements are essential in human-computer interaction (HCI) because they offer insights into individuals' cognitive states and visual attention. Techniques for adequately assessing gaze have increased in the last two decades. Notably, video-based tracking methods have gained considerable interest within the research community due to their nonintrusive nature, enabling precise and convenient gaze estimation without physical contact or invasive measures. This paper introduces a video-based gaze-tracking method that presents an affordable, user-friendly, and dependable human-computer interaction (HCI) system based on iris movement. By utilizing the MediaPipe face mesh model, facial features are extracted from real-time video sequences. A 5-point user-specific calibration and multiple regression techniques are employed to predict the gaze point on the screen accurately. The proposed system effectively handles changes in body position and user posture through real-time re-calibration using z-index tracking. Furthermore, it compensates for minor head movements that may introduce inaccuracies. The proposed system is cost-effective, with a general cost below $25, which may vary based on camera usage. Thirteen participants were involved in the system testing. The system demonstrates a high level of sensitivity to low light conditions, a strong response to changes in distance, and a moderate reaction to glasses, with an average frame processing time of 0.047 seconds. On average, it achieves a visual angle accuracy of 1.12 degrees with head movement and 1.3 degrees without head movement.

INDEX TERMS Human–computer interaction, eye-gaze tracking, iris-tracking, calibration, regression, low-cost, real-time re-calibration.

I. INTRODUCTION

In recent years, the study of gaze tracking has become increasingly popular in various research fields. The technology behind gaze tracking has many different applications, making it a topic of interest for researchers in multiple disciplines. One area where gaze tracking has proven useful is in the field of human-computer interaction (HCI). By utilizing the human gaze, individuals with severe physical disabilities can communicate with others more

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Strada.

effectively [1]. Having a gaze-based interaction system in place is essential as it can be incredibly beneficial to individuals with special needs, particularly those who are disabled. By utilizing a gaze-based system, they can live a better quality of life and improve their ability to interact with their surroundings [2]. Several methods have been proposed to achieve the objective of an eye gaze tracking-based system. The use of gaze tracking to gather information about how people use and interact with computers has exploded in recent years. However, scleral contact lenses/search coil [3] and electrodes [4] were necessary for early approaches and were invasive and uncomfortable for users. These approaches were also limited in practicality, making it challenging to track gaze in real-world scenes. Furthermore, head-mounted equipment such as headgear [5] has been used to track gaze; while this approach is less intrusive than earlier methods, it is still considered inconvenient from a practical standpoint. This is because the user must wear specialized equipment, which may interrupt their everyday activities and cause them to feel self-conscious. Compared to the above approaches, video-based gaze tracking techniques offer a practical, effective, cost-effective, and non-intrusive solution better suited for everyday use. This technology allows individuals to easily and accurately track their gaze without requiring invasive equipment or procedures [6].

Numerous video-based gaze-tracking communication systems have been invented in recent years. However, the high cost of their hardware and software remains a significant barrier to widespread adoption. Some examples of eye control systems include Tobii Rex, ITU GazeTracker [7], and The Eye Tribe [8]. Unfortunately, these systems suffer from several issues, such as high cost, wearable hardware, complex setup procedures, the necessity for the user to maintain a specific distance from the screen, and issues related to correctness and robustness. For instance, the Tobii Dynavox eye gaze trackers can cost between 5,000 to 10,000 US dollars, depending on their model and configuration [9], [10]. The ITU GazeTracker is particularly susceptible to changes in the distance between the head and the camera, which can result in tracking failure. Similarly, rotating the head can significantly impact the accuracy of The Eye Tribe system. Consequently, the existing eye control systems cannot fully address the trade-off between speed, accuracy, and robustness [8].

Most of the research on eye-tracking concentrates on implementing eye-gaze communication systems, classifiers, computer vision techniques, and facial landmark analysis to enhance system performance, utilizing a low-cost system. The primary challenge with low-cost systems lies in their simplicity and affordability, necessitating the use of hardware that is less computationally complex than its expensive counterparts. Additionally, achieving accurate gaze direction relies on effective head pose estimation. Although several techniques based on eye features have been proposed, researchers are still searching for features that allow users to maintain a comfortable head position while using the system, as subjects typically prefer. Furthermore, there are two notable concerns. First, many existing techniques do not account for natural head poses. Second, reproducing unusual settings for eye-gazing experiments proves difficult when aiming for widespread use and applicability.

Video-based gaze tracking has gained significant attention recently due to its potential to improve the lives of people with disabilities. This paper presents an economical, intuitive, and reliable communication system based on iris movement. The system tracks faces in real-time video using the MediaPipe model to extract the facial region. The system calibrates the user's eyes to specific screen points using a webcam and predicts gaze by applying multiple regression to the calibrated data. The presented work also involved real-time re-calibration to automatically adapt the changes in body position or the user's posture. The following are the paper's most significant contributions:

1) An innovative, cost-effective HCI system is introduced, featuring iris tracking, an easy setup, personalized 5-point calibration, and regression techniques.
2) **Noise Reduction:** The first 0.5 seconds of the user gaze on each point calibration is disregarded to reduce noise. Recording the data when the iris is still not focused can lead to incorrect data recordings.
3) **Multiple Regression:** Rather than relying on a single algorithm for X and Y coordinates, we calibrate X and Y axis separately. This ensures that in case of any inconsistency in 1 coordinate, at least the other coordinate gets appropriately predicted.
4) **Head Movement:** At the time of calibration, some other coordinates on the face have been recorded to measure any head movement from the original position. Adequate compensation (up to 15 %) is made for every head movement beyond that recorded in the calibration phase. This technique also ensures that if the user moves outside the coordinates recorded at the calibration time, a guess can be made of their position to some extent.
5) **Z-index Tracking (real-time re-calibration):** Often, the relative distance change between the camera and the user can cause problems with proper iris tracking. Therefore, a real-time re-calibration has been proposed to estimate the difference in the user's position on the Z-axis.
6) The head movement counteractions and Z-index re-calibration give users more freedom to move and still be able to control the screen using their iris.

The formation of the remaining paper is as follows: Section II details a summary of the related work, highlighting the existing research in the field. The descriptions of the proposed work are described in section III. The evaluation results derived from our experiments are presented in Section IV. Section V presents the conclusion and future directions.

II. RELATED WORK

Iris-tracking involves precisely measuring a person's gaze point on a visual scene and monitoring the movements of their iris. Over time, several types of iris-tracking technology have been developed, including Electro-oculography (EOG), infrared-oculography (IROG), scleral contact lens/search coil, and video-oculography (VOG) [11]. Additionally, the VOG method is classified into three categories: model, appearance, and feature-based approaches [12].

The literature review offers a brief overview of various VOG techniques. Feature-based techniques focus on extracting key characteristics from the eye region, such as the pupil, iris center, and corners, to analyze eye movements. These features are essential in accurately determining gaze direction. Conversely, appearance-based methods for gaze estimation do not require explicit feature extraction. Instead of concentrating on individual features, these methods estimate the gaze direction by utilizing full-eye images. Model-based methods are divided into 2D and 3D approaches, both of which use the geometry of the eye to compute the gaze vector for accurate eye gaze estimation [13], [14].

Feature-based eye-tracking methods focus on extracting and tracking unique eye feature landmarks, like the pupil center, iris boundaries, or corneal reflections, to determine gaze direction and measure eye movement dynamics. Aunsri and Rattarom [2] introduced an affordable HCI system that utilizes a web camera to track multiple eye features, including the pupil, glint, canthus, and the angles between them. It employs a neural network, eliminating the need for head pose adjustments and calibration. Torricelli et al. [15] proposed a method that leverages iris and corner detection techniques to extract geometric eye features. The extracted feature coordinates were subsequently transformed into screen coordinates using a general regression neural network (GRNN). It obtained an accuracy of 2.40 degrees. Valenti et al. [16] Presented a system that integrates head pose with extracted eye features to improve gaze estimation. The method reduces gaze estimation errors by accounting for these two factors, achieving an accuracy range of 2 to 5 degrees. Karatay et al. [17] developed a real-time computer interface based on eye movement detection designed to assist individuals with disabilities. This system uses the Medipipie model to accurately identify eye features. It achieved a typing speed of 23.33 characters per minute with a high accuracy rate of 95.12%, demonstrating its effectiveness as an accessible communication tool for people with physical impairments.

The model-based eye-tracking approach estimates gaze direction using mathematical models of the eye's anatomy and movement dynamics, improving eye movement detection in diverse applications. The paper [18] proposed a calibration-free, reliable, and 3D head pose estimation-based system that tracks the pupil center to estimate the user's gaze direction. The authors employed a machine learning algorithm to detect the user's iris direction. It achieved a mean accuracy of 3.81 degrees. The study [19] has introduced significant advancements in eye tracking for reading. It has presented a self-calibrating model that aligns eye-feature patterns with reading behaviors. A coarse-to-fine filtering technique has also been developed to retain fixations while removing non-reading data. Lastly, a novel pattern-matching method has been proposed that enhances fixation analysis efficiency by converting nonlinear matching on 2D planes into a linear framework on 1D lines. To improve accuracy for applications such as 3D displays and human-computer interaction, the work [20] proposed a binocular eye-tracking device that incorporates stereo stimuli, enabling highly accurate 3D PoG estimation without extra hardware.

The integration of calibration techniques in video-based systems has been proposed in several studies as a viable method to improve gaze estimation accuracy. Cheung and Peng [6] introduced a calibration-based, hardware-free method for gaze tracking using a standard webcam. The approach estimates 3D head pose through a model trained on facial features, while gaze direction is calculated by combining the iris vector with the head pose vector. The method achieved an average accuracy of 2.27 degrees with head movement and 1.28 degrees without head movement, though lighting conditions can influence its performance. Sun et al. [21] developed a cost-effective, non-invasive, easy-to-implement gaze estimation system that uses a depth camera and a calibration technique, allowing for accurate gaze tracking even with unrestricted head movement. This model-based approach achieved an average gaze estimation error of 1.4 to 2.7 degrees. Chen et al. [22] introduced a non-intrusive eye-tracking system that utilizes an implicit calibration method. During calibration, the system calculates the probability distributions of the user's eye and gaze, allowing it to adjust to changes in the user's eye parameters automatically. The system achieved an average error of less than 3 degrees. Sun et al. [23] developed an advanced real-time gaze estimation system that integrates online calibration. Unlike traditional methods that depend on static calibration points, this dynamic system continuously updates eye parameters with each new data point. The system can adapt to new users, significantly enhancing speed and user experience. It achieves a high level of accuracy, with an average error of approximately 2 degrees. Similarly, Arar et al. [24] proposed a regression and cross-ratio-based system using calibration. It obtained an average accuracy of 1 degree but was not robust to illumination changes and head/body movements. Zhang et al. [25] proposed a real-time, camera-based gaze-tracking system incorporating eye and head gaze interaction modes. It provides an alternative to hardware-based eye trackers and head-mounted devices. Using a unified calibration process for eye and head gaze, the system accurately maps gaze directions onto screen coordinates, achieving visual angle accuracy of 1.76 degrees for eye gaze and 2.65 degrees for head gaze, comparable to commercial eye trackers. Users reported similar performance to keyboard input using eye gaze and enhanced immersion with head gaze, demonstrating the system's potential for more immersive and user-friendly computer interaction.

Recent studies have explored the potential of cost-effective facial landmark-based human-computer interaction (HCI) systems. The study [26] Presents an affordable augmentative and alternative communication (AAC) tool for people with speech impairments, such as those suffering from ALS or stroke. Using a low-cost USB camera, the system employs a unique eye sign language, Netravaani, along with the Sarani algorithm to convert eye movements into text and spoken words. It eliminates the need for interpreters and minimizes frequent recalibrations while incorporating predictive text to enhance communication efficiency. The system demonstrated high accuracy in detecting letters, words, and numbers, showing its potential to assist individuals with speech impairments greatly. Bisen et al. [27] developed a system that facilitates natural and intuitive HCI by accurately detecting and tracking facial landmarks in real-time. The system extracts, identifies, and processes facial features using a combination of the histogram of oriented gradients, support vector machine [28], and regression techniques. It achieved a classification accuracy of 98.33%, though its performance was reduced in environments with inconsistent lighting. Solska et al. introduced an affordable software solution that utilizes eye-tracking technology to assist individuals with disabilities in operating computers. The software tracks the user's eye movements and translates them into input commands, enabling interaction with various computer programs and features. The system demonstrated an average accuracy of 4.1 degrees, with a precision of 0.042 degrees. TAS and Yavuz [29] developed a robust algorithm for real-time tracking and recognition of eye, eyebrow, and head movements. The authors use computer vision approaches to detect and track specific features such as the position of the iris, eyebrow arch, and head orientation. Machine learning methods were employed to train the algorithms, allowing the recognition of movement patterns and their conversion into computer commands. The system achieved an average typing speed of 7 characters per minute. Ramos et al. [30] designed and implemented a cost-effective human-computer interaction (HCI) system that allows computer control through facial landmark detection. In addition to tracking facial features, the system integrates voice commands as an alternative input method. The authors highlight the incorporation of speech recognition technology, enabling users to operate the computer through voice commands. The study [31] introduced a novel HCI system, leveraging facial landmarks with a unique calibration approach. This technique entailed the computation of each participant's individualized screen multipliers and blinking ratios. By computing these calibration parameters on a per-participant basis, the system aimed to enhance the accuracy and personalized nature of the interaction experience.

Bozomitu et al. [32] developed a system that evaluated eight advanced pupil recognition algorithms. The study found that all algorithms detected pupils with an accuracy above 84% at 50 pixels, with the circular Hough transform (CHT) method achieving the highest accuracy rate of 91.39%. The system was tested with 30 participants, who achieved an average typing speed of 20.74 characters per minute with an error rate of 3.55%. The paper [33] presented the Blink-To-Live system, which leverages eye-tracking and facial landmark technologies to interpret intentional and controlled eye blinks as a communication method. The system uses a highly efficient algorithm that accurately detects and analyzes blink patterns in real-time, with an average detection time of 0.21 seconds. Similarly, Paing et al. [34] developed a system that signifies a noteworthy progression in assistive technology to improve the lives of individuals with disabilities. The system introduces a novel approach that empowers users to control their wheelchairs and engage with their surroundings through eye movements. However, it is essential to note that integrating this system with the GazePoint GP3 eye tracker may increase the overall cost.

Numerous studies have shown that appearance-based techniques, such as neural networks and convolutional neural networks (CNNs), are extensively used in fields like gaze estimation and human-computer interaction. For gaze tracking, Tosen et al. [35] introduced a system called 'InvisibleEye,' which utilizes four miniature RGB cameras. The system achieved an accuracy of 1.79 degrees in estimating a single user's gaze direction. However, synchronizing the output from four cameras presents more complexity compared to using a single webcam. Similarly, The work [36]presented a study focused on developing an eye-gaze input interface using the combination of multiple cameras and CNN. The interface utilizes images from numerous cameras to accurately track and interpret the user's eye gaze. Notably, the system attained commendable results, with an average of 146.02 and 148.94 pixel errors, horizontal and vertical, respectively. Ansari et al. [37] developed a system to estimate gaze direction using images captured from low-resolution webcams. The authors created a dataset by collecting facial and eye images from low-quality webcams. To enhance the precision of gaze estimation, they trained a machine learning model that accounts for unique facial and eye characteristics and achieved an accuracy of 1.98 degrees. Donuk et al. [38] developed a real-time system utilizing convolutional neural networks (CNNs) to detect a user's points of interest on a webpage. The system's model has been trained on a dataset of 23,040 eye images captured under various lighting conditions. During testing, the model demonstrated an average error of 32 pixels in training and 53 pixels in testing. Shi et al. [39]presented a CNN-based approach for the HCI system by utilizing facial features. Combining various aspects such as facial features, expression features, and sight tracking, the authors explore three-dimensional user usage information and associate it with specific facial activities

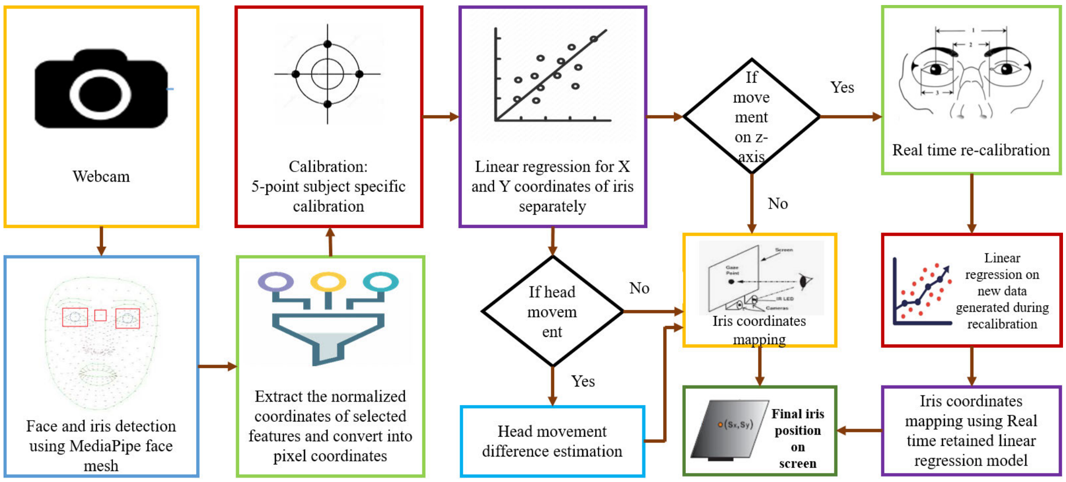


FIGURE 1. Workflow of the proposed method.

to enable effective interaction. Notably, their approach demonstrated outstanding performance in tracking human sight, achieving an impressive accuracy rate of 0.91.

The current study highlights several limitations associated with existing gaze estimation methods. Systems that utilize feature-based techniques are particularly sensitive to changes in lighting conditions and variations in eye shape or size, which can compromise the accuracy and reliability of gaze estimation. Similarly, model-based eye tracking can be computationally demanding and necessitates precise calibration, with a significant drawback being its limited generalizability across diverse users. Moreover, such systems often require depth cameras, contributing to higher costs and reduced performance when applied to various populations. In contrast, appearance-based eye tracking depends on large and diverse datasets for training, which can be difficult to acquire, while also being adversely affected by variations in eye appearance, head position, and the user's distance from the screen, all of which can further diminish accuracy.

To address these challenges, this work presents an innovative and cost-effective human-computer interaction (HCI) system that leverages iris tracking, featuring a straightforward setup, personalized five-point calibration, and regression techniques. To enhance accuracy, the system ignores the first 0.5 seconds of the user's gaze during calibration, minimizing noise from incorrect data when the iris is not yet focused. It employs multiple regression methods, calibrating the X and Y coordinates separately to ensure precise predictions even if one coordinate is inconsistent. Additionally, the system records extra facial coordinates to monitor head movements during calibration, allowing for compensation of up to 15% for any deviations. Furthermore, Z-index tracking is incorporated for real-time re-calibration, addressing potential issues that arise from changes in the user's distance from the camera. Together, these features empower users with greater freedom of movement while ensuring reliable gaze-tracking capabilities.

III. PROPOSED METHOD

Within a facial image, the iris center is one of the most significant elements associated with the gaze. As the eyes scan different locations on a screen, the eyeball dynamically shifts within its socket, resulting in a corresponding change in the position of the iris center within the eyeball. This positional change is a reliable indicator of the direction of the individual's gaze. However, it is crucial to acknowledge that head movements can introduce inaccuracies in the gaze direction, leading to errors. To address this challenge, estimating the head pose accurately is imperative, enabling compensation for head movements and ensuring more accurate gaze determination. The proposed method consists of seven main phases: 1) scaling and feature extraction phase; 2) Extracting the normalized coordinates of selected features and converting them into pixel coordinates; 3) 5-point subject-specific calibration; 4) linear regression; 5) head movement estimation; 6) real-time re-calibration; and 7) final iris position mapping on the screen. An overview of the proposed method is depicted in Figure 1.

A. SCALING AND FEATURES EXTRACTION

The proposed real-time iris-tracking-based HCI system consists of one webcam for video frame capture, and the captured frame's and operating screen's dimensions are calculated. The dimensions are represented as F_w, F_h, S_w, S_h. Where F_w, F_h, S_w, and S_h represent the frame width, frame height, screen width, and screen height, respectively.

The proposed system starts with the localization of the face in the video frame. To estimate the gaze, we extract the iris center and glabella (midpoint between eyebrows) features in the face region. To localize and track the selected features, we utilize a MediaPipe FaceMesh model [40], [41]. MediaPipe comprises two distinct modules, namely "MediaPipe face mesh" and "MediaPipe iris," which facilitate the development of interactive real-time systems using computer vision and machine learning methodologies. MediaPipe Iris is a computer vision model that provides the normalized coordinates of the iris position within a video frame. Unlike conventional pixel value outputs, MediaPipe Iris presents these coordinates on a scale from 0.0 to 1.0. The top left corner of the frame is denoted by the coordinates (0.0, 0.0), while the bottom right corner is represented by (1.0, 1.0). Unlike standard coordinate systems, normalized coordinates are not tied to physical measurements. However, they can easily be converted back to pixel coordinates by taking into account the dimensions of the input video frame [42]

$$\pi_i = \psi\,(x_n) * f_w$$
$$\pi_j = \psi\,(y_n) * f_h \tag{1}$$

where $\psi\,(x_n, y_n)$ are the normalized coordinates of the extracted features. These coordinates provide a standardized representation of the location and positioning of the features within the captured video frame. On the other hand, f_w and f_h refer to the frame's width and height, respectively.

In Equation 1, we define two variables: $i = (g_x, LI_x, RI_x)$ and $j = (g_y, LI_y, RI_y)$. Here, (g_x, g_y) represents the X and Y coordinates of the glabella, (RI_x, RI_y) denotes the X and Y coordinates of the right iris center, and (LI_x, LI_y) refers to the X and Y coordinates of the left iris center. Equation 2 is employed to calculate the midpoint of both irises.

$$M_x = \frac{RI_x + LI_x}{2}$$
$$M_y = \frac{RI_y + LI_y}{2} \tag{2}$$

where (M_x, M_y) is the central position coordinates of both the iris.

B. 5-POINT SUBJECT-SPECIFIC CALIBRATION

The subject-specific calibration process is an essential aspect of eye-tracking technology, as it enables the system to acquire knowledge about the subject's unique facial features and characteristics. The calibration process involves presenting a predetermined set of points to the subject, during which the corresponding feature vectors are recorded. Subsequently, the mapping function is employed to establish the correlation between the recorded feature vector and the screen coordinates to determine the gaze positions accurately [12]. The proposed work employs a 5-point user-specific calibration approach to facilitate iris tracking and cursor movement. Linear regression is used for mapping between coordinates. The entire calibration process is designed to be completed within a duration of 15 seconds, with each calibration point allocated a time frame of 3 seconds. Figure 2 depicts the calibration steps. We started by drawing five points (radius: 15 pixels, color: green) on a calibration screen. Users have to look at each point for 3 seconds. On the 4-corner point (Top left, Top right, bottom left, and bottom right), we recorded only the iris coordinates, but some extra features (iris diameter, glabella) have also been recorded on the middle point along with iris coordinates. The horizontal

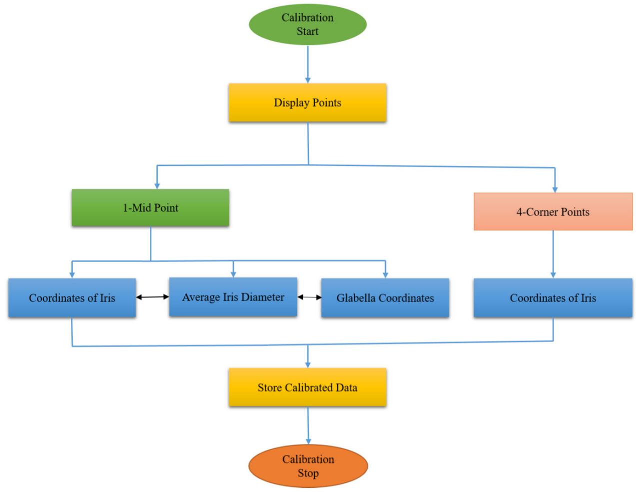


FIGURE 2. Visualization of the calibration approach.

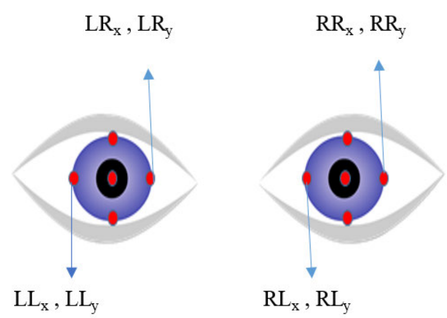


FIGURE 3. Visualization of iris landmark coordinate.

coordinates of the irises are used to calculate the iris diameter, as they are less likely to be obscured by the eyelids compared to the vertical coordinates. The horizontal iris landmark is determined using Equation 3, and the Euclidean distance between these horizontal coordinates is calculated using Equations 4 and 5. Finally, the average iris diameter is computed using Equation 6.

$$\eta_i = \psi\,(x_n) * f_w$$
$$\eta_j = \psi\,(y_n) * f_h \tag{3}$$

where $\psi\,(x_n, y_n)$ are the normalized coordinates of both iris and f_w and f_h are the width and height of the frame's.

Equation 3 introduces the variables: $i = (LL_x, LR_x, RR_x, RL_x)$ and $j = (LL_y, LR_y, RR_y, RL_y)$. The variables in the equation denote specific coordinates of landmarks within the context of iris recognition. For example, (LL_x, LL_y) refers to the X and Y coordinates of the leftmost landmark of the left iris, while other variables similarly represent distinct landmarks. The visual representation of these variables is shown in Figure 3.

$$LI_D = \sqrt{(LR_x - LL_x)^2 + \left(LR_y - LLy\right)^2} \tag{4}$$

$$RI_D = \sqrt{(RR\text{x-R}Lx)^2 + (RR\text{y-R}Ly)^2} \tag{5}$$

Algorithm 1 Pseudo-Code for Calibration

Require: video frame, calibration screen, time(t), M_x, M_y, A_D, g_x, g_y
Ensure: Return V

Start
if $t > 0$ & $t < 3$ **then** ▷ Top left point
  TL_x.append(M_x)
  TL_y.append(M_y)
else if $t > 3$ & $t < 6$ **then** ▷ Top right point
  TR_x.append(M_x)
  TR_y.append(M_y)
else if $t > 6$ & $t < 9$ **then** ▷ Bottom left point
  BL_x.append(M_x)
  BL_y.append(M_y)
else if $t > 9$ & $t < 12$ **then** ▷ Bottom right point
  BR_x.append(M_x)
  BR_y.append(M_y)
else if $t > 12$ & $t < 15$ **then** ▷ Mid point
  MP_x.append(M_x)
  MP_y.append(M_y)
  AE_w.append(A_D)
  G_x.append(g_x)
  G_y.append(g_x)
end if
Return $V = (TL_x, TL_y, TR_x, TR_y, BL_x, BL_y, BR_x, BR_y, MP_x, MP_y, AE_w, G_x, G_y)$

$$A_D = \frac{LI_D + RI_D}{2} \tag{6}$$

Algorithm 1 outlines the step-by-step calibration procedure, which takes 15 seconds. It begins by capturing and recording the coordinates of the iris center at the top left (TL_x, TL_y), top right (TR_x, TR_y), bottom left (BL_x, BL_y), and bottom right (BR_x, BR_y). Similarly, we recorded the iris diameter (AE_w) and glabella (G_x, G_y) along with the iris center (MP_x, MP_y) at the middle point. Following the calibration procedure, we computed the mean value of the glabella (G'_x, G'_y), iris center (X'_m, Y'_m), and iris diameter (A'_D) that was recorded at the midpoint, utilizing Equations 7, 8, and 9. Figure 4 presents the calibrated gaze data recorded when users focus on five specific points on the screen. Figure 4 includes data captured both with and without head movement. The individual point chunks represent the iris coordinates observed by the camera as the user looks at each calibration point. Each chunk illustrates the concentration of gaze data, providing insight into how consistently the system captures the gaze for each calibration point. The different colors in the figure correspond to the clusters of recorded iris coordinates for each calibration point, helping to distinguish between the data collected at different screen locations visually.

$$\begin{aligned} G'_x &= \text{sum}\,(G_x)\,/\,\text{length}\,(G_x) \\ G'_y &= \text{sum}\left(G_y\right)/\,\text{length}\left(G_y\right) \\ X'_m &= \text{sum}\,(MP_x)\,/\,\text{length}\,(MP_x) \end{aligned} \tag{7}$$

$$Y'_m = \text{sum}\left(MP_y\right)/\,\text{length}\left(MP_y\right) \tag{8}$$

$$A'_D = \text{sum}\,(AE_w)\,/\,\text{length}\,(AE_w) \tag{9}$$

C. PREDICTING THE GAZE THROUGH REGRESSION ANALYSIS

Regression-based methodologies are computational procedures that establish a correlation or mapping between two-dimensional feature vectors and the corresponding gaze positions on a display screen. During the calibration, we collect a set of calibrated data of the iris concerning calibration points on the screen. Rather than relying on a single regression for X and Y coordinates, we train regression for X and Y axis separately. This ensures that in case of any inconsistency in 1 coordinate, at least the other coordinate gets appropriately predicted. We combine X coordinates ($X_I = TL_x, TR_x, BL_x, BR_x, MP_x$) for all recorded coordinates of the iris center on five points in a single array. Along with that, we also create an equal array for the X coordinates of the desired value (X_v, calibration point position). The same thing is repeated for the Y coordinates ($Y_I = TL_y, TR_y, BL_y, BR_y, MP_y$) and their desired value (Y_v).

$$X_v = \alpha\,(X_I) + \beta \tag{10}$$

$$Y_v = \alpha\,(Y_I) + \beta \tag{11}$$

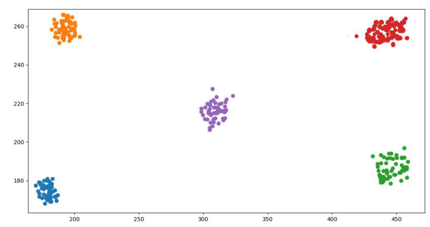
ModelX and ModelY are represented by Equations 10 and 11 respectively.

We can not use regression for glabella and eye diameter since we are recording their coordinates for counterattacking head movement and Z-axis movement. At the time of calibration, we record the coordinates of the glabella point, which is used to measure any head movement from the original position. Adequate compensation (up to 15 %) is made for every head movement beyond that recorded in the calibration phase. This also ensures that if the user moves outside the coordinates recorded at the time of calibration, we can still guess their position to some extent. The head movement difference (Hm_x, Hm_y) is measured by Equation 12.

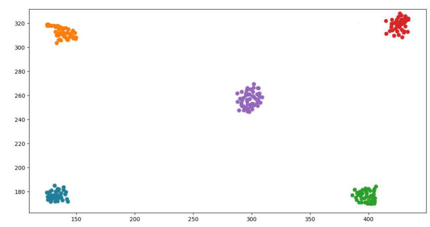
$$\begin{aligned} Hm_x &= (g_x) - \left(G'_x\right) \\ Hm_y &= \left(g_y\right) - \left(G'_y\right) \end{aligned} \tag{12}$$

D. Z-INDEX TRACKING (REAL-TIME RE-CALIBRATION)

To ensure accurate iris tracking, it is crucial to address potential issues arising from the relative distance change between the camera and the user. To tackle this challenge, we employ a method that estimates the disparity in the user's position on the Z-axis. At the time of calibration, the diameter of the iris is measured. Since the actual iris diameter is always constant, whenever the camera observes an increase/decrease in the diameter of the iris (in terms of pixels), we can estimate the iris to have moved close/away from the screen respectively. A scaling factor is calculated by comparing the current average iris diameter (A'_D) versus the actual diameter (A_D). A re-calibration is performed based on the iris diameter



(a) With Head Movement



(b) Without Head Movement

FIGURE 4. **Calibrated iris-gaze data with and without head movement.**

difference, and the linear regression algorithm is retrained in real-time. The difference in iris diameter, denoted as (ID_d), can be obtained using Equation 13, while the scaling factor (S_f) can be determined using Equation 14.

$$ID_d = |A_D - A'_D| \tag{13}$$

$$S_f = (A'_D + ID_D)/A'_D \tag{14}$$

The proposed system incorporates a comprehensive approach to iris recognition by calculating the iris diameter difference (ID_d) and the scaling factor (S_f) for each frame. We compare the ID_d with the actual iris diameter (A_D) and also consider the change in scaling factor from the previous frame to the current frame. When a significant difference is detected between these values, we initiate real-time re-calibration. This re-calibration process allows us to adapt and adjust the iris recognition parameters dynamically, enhancing the precision and reliability of our system. In the re-calibration process, the recorded coordinates (X_I, Y_I) obtained during calibration are scaled relative to the iris center (X'_m, Y'_m). The real-time re-calibration process is conducted using Equation 15, which generates the new x and y coordinates (N_x, N_y).

$$\begin{aligned} N_x &= \left(X'_m + ((1 + (S_f - 1)/2) * (X'_m - x))\right)) \\ N_y &= \left(Y'_m + ((1 + (S_f - 1)/2) * (Y'_m - y))\right)) \end{aligned} \tag{15}$$

The real-time re-calibration procedure is outlined in Algorithm 2, which efficiently computes the XR_i and YR_i values for the x and y coordinates. To initiate the process, we generate an array, denoted as X_t, comprising the XR_i values. Simultaneously, we construct an equal array, X_v, consisting of the desired values for the X coordinates (calibration point position). A linear regression model (Equations 16 and 17) is then trained to establish the relationship between X_t and X_v. Similarly, we repeat this procedure for the Y coordinates by creating the Y_t array (containing YR_i values) and the Y_v array (consisting of the desired values for the Y coordinates). RTModelX and RTModelY are represented by Equations 16 and 17, respectively.

$$X_v = \alpha\,(X_t) + \beta \tag{16}$$

$$Y_v = \alpha\,(Y_t) + \beta \tag{17}$$

Algorithm 2 Pseudo-Code for Real-Time Re-Calibration

Require: Iris diameter difference (ID_d), Scaling Factor (S_f)
Ensure: Return R

Calculation of Iris diameter difference (ID_d)
Scaling Factor (S_f) and stored in (S'_f)
if $ID_d > 0.1$ **then**
  $S'_f - S_f$ ▷ comparison between scaling factor
  if $S'_f - S_f > 0.1$ **then** ▷ real-time re-calibration
    $XN_I = X_I$ ▷ for X coordinate
    for x in XN_I **do**
      $N_x = (X'_m + ((1 + (S_f - 1)/2) * (X'_m - x)))$
      if $N_x > F_w$ **then**
        $N_x = F_w$
      end if
      if $N_x < 0$ **then**
        $N_x = 0$
      end if
      XR_i.append(N_x)
    end for
    $YN_I = Y_I$ ▷ for y coordinate
    for y in YN_I **do**
      $N_y = (Y'_m + ((1 + (S_f - 1)/2) * (Y'_m - y)))$
      if $N_y > F_h$ **then**
        $N_y = F_h$
      end if
      if $N_y < 0$ **then**
        $N_y = 0$
      end if
      YR_i.append(N_n)
    end for
  end if
end if
Return $R = (XR_i, YR_i)$

E. FINAL IRIS-GAZE POSITION ON SCREEN

Following the preliminary stages, including scaling and feature extraction, calibration, head movement estimation, and z-index re-calibration, as well as model training, we can accurately determine the final gaze position (S_x, S_y) on the screen in real-time scenarios. This prediction is achieved through the use of a trained regression model. When the

TABLE 1. System performance results (seconds per frame).

Testing metrics	Average speed (in seconds)
Video acquisition	0.033366700
Facial features detection	0.008636523
Eye, Iris, and glabella plotting on screen	0.008297681
General time per loop	0.047212794
Retraining fit (excluding detection)	0.006794573
Head position counter actions	0.000233010

user remains stable in real-time usage, we employ the trained model (Equation 18) to predict the gaze accurately. However, if the user exhibits movement along the Z-axis, we utilize a retrained regression model (Equation 19) to anticipate the gaze position effectively.

$$S_x = ModelX.predict\ (M_x)$$
$$S_y = ModelY.predict\ (M_y) \tag{18}$$
$$S_x = RTModelX.predict\ (M_x)$$
$$S_y = RTModelY.predict\ (M_y) \tag{19}$$

In the event of significant glabella movement, a corrective measure is applied by subtracting the iris coordinates to realign the iris within the frame. This adjustment allows us to ensure accurate gaze prediction by bringing the iris back into the appropriate position. Subsequently, the trained model (referred to as Equation 18) is employed to estimate the gaze position effectively. By incorporating this approach, we can mitigate the impact of head movement and maintain the precision of our gaze prediction system. The steps involved in implementing the proposed system are briefly outlined in Algorithm 3.

Algorithm 3 Pseudo-Code of Proposed Work for Iris-Tracking Based HCI

Require: video frame
Ensure: Final gaze position on screen (S_x, S_y)
Initialization
  Frame scaling and features extraction using MediaPipe
  Calibration, Linear regression, Head-movement estimation, Real-time re-calibration
Tracking the gaze through all the frames:
while True **do**
    Extract the features from the video frame
    Iris center calculation (M_x, M_y)
    Calibration
    Linear regression ($ModelX$, $ModelY$)
    Head movement estimation (Hm_x, Hm_y)
    Z-index tracking (Real-time re-calibration) (ID_d, S_f) ($RTmodelX$, $RTmodelY$)
    if Movement on z-index **then** ▷ re-calibration
        S_x = RTmodelX.predict(M_x)
        S_y = RTmodelY.predict(M_y)
        Gaze position (S_x, S_y)
    else if Head-movement **then**
        S_x = ModelX.predict(M_x)
        S_y = ModelY.predict(M_y)
        S_x = S_x - Hm_x
        S_y = S_y - Hm_y
        Gaze position (S_x, S_y)
    else if Stable: without head or z-index **then**
        S_x = ModelX.predict(M_x)
        S_y = ModelY.predict(M_y)
        S_x = S_x - Hm_x * 1.15
        S_y = S_y - Hm_y * 1.15
        Gaze position (S_x, S_y)
    end if
end while

IV. EVALUATION RESULTS OF THE PROPOSED METHOD

In this section, we provide the evaluation results of the proposed method. We performed several experiments on real-time generated data to evaluate the proposed system's performance. The performances were calculated and subsequently reported in terms of gaze estimation accuracy errors. This error metric quantifies the mean displacement in degrees of visual angle between the ground truth points and the estimated gaze points.

A. SYSTEM SETUP

The experimental setup utilizes an Acer Predator PH315-51, featuring an Intel Core i5-8300H CPU at 2.30GHz, 8 GB of RAM, a 4 GB graphics card, and a 128 GB SSD. The machine boasted a resolution and display of 1920 × 1080, featuring a 15.6-inch Display. The system uses a built-in laptop camera and an external webcam to capture participants' facial regions. A calm environment is maintained throughout the experiments to minimize potential distractions. The distance between the participants and the monitor was set at approximately 40-50 cm. We included 13 participants in the experiment, five of whom wore glasses. Among them, nine were male, aged between 17 and 60 years, and four were female, aged between 17 and 50 years. None of the participants had prior experience with iris-tracking-based computer cursor control. The experiments were conducted in a controlled environment, where lighting sources included fluorescent lights, room tube lights, and natural sunlight. This ensured diverse illumination conditions to examine the subjects' performance under different lighting scenarios. The experimental procedure encompassed two different sessions: a training session and a practice session. Before testing, each participant underwent a brief training session to ensure accurate calibration of the proposed system. During this phase, participants were guided to focus on specific points on the screen to align their gaze accurately with the system's parameters. Participants were also instructed on how to use the developed technique, ensuring they understood the system's functionality. After completing the training, participants independently interacted with the system, controlling the on-screen cursor using only their iris movements for at least 5 minutes. Each participant repeated

TABLE 2. Robustness of the system and comparison with existing method.

Metrics	Proposed system	Wu et al. [8]	ITU gaze tracker [7]
Calibration	Yes	No	Yes
Camera used	Webcam	Microsoft Kinect 2.0	Webcam
Sensitivity to Low light	intermediate-high	Low	Mid
Sensitivity to Glasses	Moderate	Low	Mid
Sensitivity to Distance Change	High	Mid	High

this task three times, allowing us to collect comprehensive data.

B. SYSTEM PERFORMANCE: SPEED AND ROBUSTNESS

The system's speed and robustness evaluation confined different aspects, including lighting conditions, operational distance of participants, and occluding glasses. During the trial, the system's performance is measured by tracking the average time it took to complete a single loop. It involved tasks such as video acquisition, tracking chosen features, and accurately moving the cursor to the point of gaze. Based on the analysis, the system exhibited an impressive speed, with an average completion time of 0.047212794 seconds per frame. The performance results of the system are presented in Table 1, highlighting its achieved capabilities. Furthermore, Table 2 provides the outcomes of the robustness test, demonstrating the system's resilience under various conditions.

C. VISUAL ANGLE ACCURACY IN DEGREE

This section is focused on evaluating the effectiveness of the proposed iris-tracking interface under real-time conditions, aiming to offer valuable insights into its performance. The evaluation of the proposed system encompasses two key aspects: gaze estimating without head movement and gaze estimating with head movement. To determine the accuracy of the visual angle, we employed a methodology involving the placement of ground truth points (top-left (TL), top-right (TR), middle point (MP), bottom-left (BL), and bottom-right (BR)) on the screen. These points were defined by a radius of 30 pixels and depicted in green color, strategically positioned at coordinates (0, 0), (1920, 0), (0, 1080), (1920, 1080), and (960, 540). Participants were instructed to look at each point for a duration of 10 seconds. To obtain the visual angle accuracy, we compute the pixel distance between each recorded gaze point, denoted as $R_1(x_1, y_1)$ to $P_n(x_n, y_n)$, and their respective ground truth points, denoted as $C(x_c, y_c)$, utilizing Equation 20. The monitor's pixel size (μ), which is based on the screen dimensions and pixel resolution, is calculated utilizing Equation 21 [43].

$$P_d = \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2} \tag{20}$$

$$\mu = d_m / s_d \tag{21}$$

where μ is the size of a pixel, d_m is the diagonal size in cm (converted from inches), and s_d is the screen diagonal size in pixels as calculated from Equation 22. w_p & h_p are the screen width and height in pixels, respectively.

$$s_d = \sqrt{(w_p)^2 + (h_p)^2} \tag{22}$$

The visual angle accuracy of the proposed iris-gaze tracking system is calculated by Equation 23.

$$AV_d = atan(dc_m / u_d) \tag{23}$$

where in Equation 23 dc_m is the pixel distance in centimeters, which is calculated by $(\mu * P_d)$, and u_d is the distance between the user's iris and the operating screen.

TABLE 3. Average visual angle accuracy (in degree) with head movement for all points.

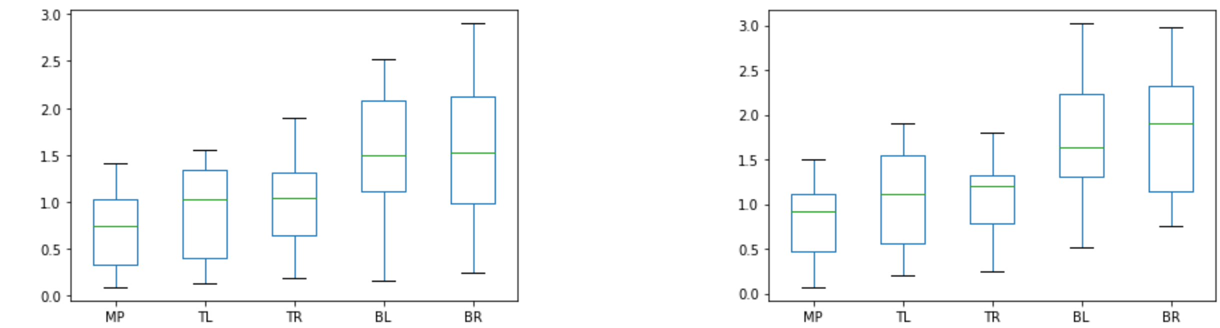
Points	Average degree
MP	0.70
TL	0.94
TR	1.01
BL	1.4
BR	1.58
Average	1.12

TABLE 4. Average visual angle accuracy (in degree) without head movement for all points.

Points	Average degree
MP	0.82
TL	1.09
TR	1.06
BL	1.77
BR	1.82
Average	1.3

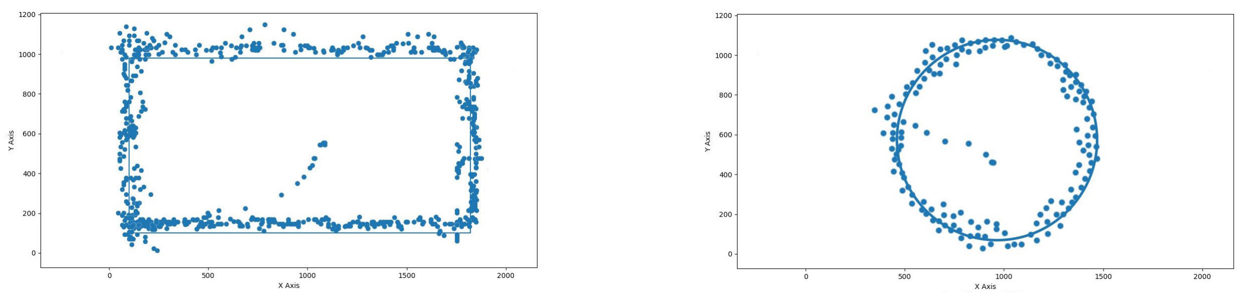
The average results obtained from the proposed system are demonstrated in Tables 3 and 4. Table 3 exhibits the results obtained while incorporating head movements, whereas Table 4 showcases the results achieved without considering head movements. The tables provided offer a comprehensive overview of the performance of the proposed system. The table data indicates that the system performs notably better when minor head movements are incorporated.

The average visual angle accuracy of the proposed HCI system is approximately 1.12 degrees when head movement is taken into account and 1.3 degrees when head movement is not considered. The accuracy of gaze tracking is not uniformly distributed across the screen. The accuracy remains consistent and stable in the middle portion of the screen. However, as we move towards the screen edges, there is a slight increase in accuracy. The system exhibits commendable performance at the midpoint, regardless of the presence



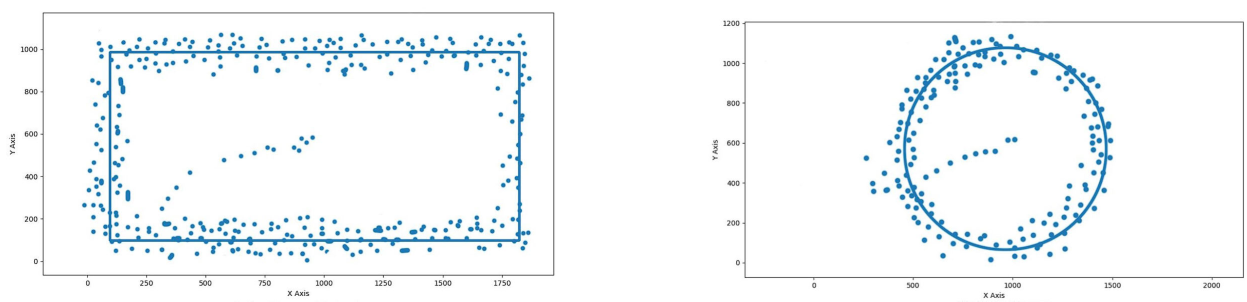
(a) With Head Movement (b) Without Head Movement

FIGURE 5. The accuracy achieved by each participant on all points.



(a) Rectangle (b) Circle

FIGURE 6. Visualization of real-time user interaction (with head movement) with predefined shapes.



(a) Rectangle (b) Circle

FIGURE 7. Visualization of real-time user interaction (without head movement) with predefined shapes.

or absence of head movement. However, slight variations in accuracy may arise in the corner points, depending on the participant's position in relation to the camera. Although these variations are detectable, they are relatively minimal. The accuracy assessment of the system, accounting for head movement, underscores its impressive performance across different positions. Notably, at the midpoint, the system achieves an outstanding accuracy of 0.12 degrees, representing its highest. However, the accuracy at the bottom right point drops to 2.53 degrees, reflecting the system's lowest performance. On the other hand, without head movement, the system's accuracy remains commendable at the midpoint, reaching an impressive 0.07 degrees. Nonetheless, at the bottom left point, resulting in the system's lowest accuracy of 3.02 degrees. Figure 5 presents the complete accuracy achieved by each participant, where the x-axis represents the points, and the y-axis represents the degrees.

Additionally, to assess our proposed system's real-time efficacy, we conducted an experiment involving predefined shapes, namely a rectangle and a circle. Users were instructed to follow these shapes, with the option to include head movements or without head movement. The results of this experiment are visually represented in Figures 6 and 7. These figures provide a comprehensive glimpse into the outcomes obtained during the evaluation process. The gaze points depicted on the shapes presented in Figures 6 and 7

TABLE 5. Results comparison of the proposed method with existing methods.

Methods	Accuracy(degrees) with head movement	Accuracy(degrees) without head movement
Ansari et al. [37]	-	1.98
Torricelli et al. [15]	2.40	-
Valenti et al. [16]	2	5
Cazzato et al. [18]	3.81	-
Cheung et al. [6]	2.27	1.28
Sun et al. [21]	1.4	2.7
Chen et al. [22]	3	-
Sun et al. [23]	2	-
Arar et al. [24]	1	-
Zhang et al. [25]	2.65	1.76
Solska et al. [44]	-	4.1
Tonsen et al. [35]	-	1.79
Proposed method	**1.12**	**1.3**

TABLE 6. Cost comparison with existing methods.

Methods	Hardware	Cost (approximate)
Aunsri et al. [2]	DEPSTECH 1080P HD webcam, SIGNO WC-208 webcam	15-25$
Paing et al. [34]	GazePoint GP3	845$
Proposed	Laptop inbuilt webcam	No additional cost
	Logitech c270 webcam	18-20$

pertain to two distinct users: one performed the task with head movements, while the other did so without any head movement. An observation made from these figures is that, in most cases, the gaze error is noticeably higher towards the bottom of the screen due to the eyelids' partial occlusion of the iris. Upon careful examination of the visual representation of the gaze points in Figures 6 and 7, it becomes apparent that the proposed system showed slightly superior performance when considering minor head movements compared to scenarios where the head movement was not involved. In order to evaluate the efficacy of the proposed method, a comprehensive comparison was conducted against an existing method, taking into account the metrics of both with head movement and without head movement. The outcomes of this comparative analysis are presented in Table 5.

The presented framework has demonstrated superior performance compared to other proposed methods, particularly when utilizing only a few calibration points. The results reveal an impressive average accuracy of approximately 1.12 degrees when accounting for head movement and 1.3 degrees without head movement. With a smooth tracking rate of 30 frames per second webcam, the system provides natural and personalized gaze tracking, rendering it highly applicable in HCI scenarios. Notably, the developed framework boasts a lower cost than existing methods, as highlighted in Table 6, confirming its cost efficiency and practicality.

The results demonstrate the effectiveness of the proposed gaze-tracking system. Still, the study involves a limited number of participants, which may not be a representative sample of the broader population, impacting its generalizability. The current approach has an inherent inability to operate when multiple faces are detected in the frame. While the system compensates for minor head movements, its effectiveness in managing larger or more rapid movements like complete repositioning of the device and user is challenging.

V. CONCLUSION AND FUTURE WORK

In this research work, we have introduced an innovative real-time cursor movement system that relies on iris tracking, making it highly suitable for HCI. A distinctive aspect of our system is its ability to operate only using a standard webcam, eliminating the need for additional hardware. By analyzing real-time video sequences, our method effectively tracks the human face. It utilizes the face mesh model from Google's MediaPipe library to extract facial and iris regions accurately. Significantly, our method does not mandate high-resolution eye data for operation, enabling it to manage adeptly with variations in illumination and accommodate diverse head movements. To establish a personalized calibration for individual users, we employ a 5-point calibration technique that precisely maps the user's iris to specific points on the screen. Additionally, we incorporate regression techniques to train the model using the calibrated data. An essential feature of our approach is the integration of real-time re-calibrations, which allow the system to autonomously adapt to changes in the user's body position or posture, ensuring continuous and accurate cursor movement. Through these advancements, our system provides a reliable and efficient solution for real-time iris-based cursor control in HCI scenarios.

As we imagine the future enhancement of our work, we aim to further enhance our system's accuracy by utilizing more calibration points. Additionally, we plan to explore the potential of training a CNN model for each extracted feature. The proposed work focused on presenting the overall results to highlight the general effectiveness of our gaze-tracking system. The intent was to demonstrate the system's overall effectiveness and adaptability across a range of conditions and provide a clear, comprehensive overview of the system's capabilities in a controlled manner. Conducting detailed subgroup analyses in future research will enhance the understanding of how different aspects, like lighting conditions and the presence of glasses, affect system reliability and usability. It will provide valuable insights and contribute to refining gaze-based interaction systems for diverse user scenarios.

REFERENCES

[1] G. R. Chhimpa, A. Kumar, S. Garhwal, and Dhiraj, "Empowering individuals with disabilities: A real-time, cost-effective, calibration-free assistive system utilizing eye tracking," *J. Real-Time Image Process.*, vol. 21, no. 3, p. 97, Jun. 2024.

[2] N. Aunsri and S. Rattarom, "Novel eye-based features for head pose-free gaze estimation with web camera: New model and low-cost device," *Ain Shams Eng. J.*, vol. 13, no. 5, Sep. 2022, Art. no. 101731.

[3] E. Whitmire, L. Trutoiu, R. Cavin, D. Perek, B. Scally, J. Phillips, and S. Patel, "EyeContact: Scleral coil eye tracking for virtual reality," in *Proc. ACM Int. Symp. Wearable Comput.*, Sep. 2016, pp. 184–191.

[4] Y. Nam, B. Koo, A. Cichocki, and S. Choi, "GOM-face: GKP, EOG, and EMG-based multimodal interface with application to humanoid robot control," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 2, pp. 453–462, Feb. 2014.

[5] K. Takemura, K. Takahashi, J. Takamatsu, and T. Ogasawara, "Estimating 3-D point-of-regard in a real environment using a head-mounted eye-tracking system," *IEEE Trans. Hum.-Mach. Syst.*, vol. 44, no. 4, pp. 531–536, Aug. 2014.

[6] Y.-M. Cheung and Q. Peng, "Eye gaze tracking with a web camera in a desktop environment," *IEEE Trans. Hum.-Mach. Syst.*, vol. 45, no. 4, pp. 419–430, Aug. 2015.

[7] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen, "Evaluation of a low-cost open-source gaze tracker," in *Proc. Symp. Eye-Tracking Res. Appl. (ETRA)*, 2010, pp. 77–80.

[8] T. Wu, P. Wang, Y. Lin, and C. Zhou, "A robust noninvasive eye control approach for disabled people based on Kinect 2.0 sensor," *IEEE Sensors Lett.*, vol. 1, no. 4, pp. 1–4, Aug. 2017.

[9] M. T. Chew and K. Penver, "Low-cost eye gesture communication system for people with motor disabilities," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2019, pp. 1–5.

[10] X. Zhang, H. Kulkarni, and M. R. Morris, "Smartphone-based gaze gesture communication for people with motor disabilities," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2017, pp. 2878–2889.

[11] X. Zhang, X. Liu, S.-M. Yuan, and S.-F. Lin, "Eye tracking based control system for natural human–computer interaction," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–9, Dec. 2017.

[12] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[13] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[14] O. Ferhat and F. Vilariño, "Low cost eye tracking: The current panorama," *Comput. Intell. Neurosci.*, vol. 2016, no. 1, pp. 1–14, 2016.

[15] D. Torricelli, S. Conforto, M. Schmid, and T. D'Alessio, "A neural-based remote eye gaze tracker under natural head motion," *Comput. Methods Programs Biomed.*, vol. 92, no. 1, pp. 66–78, Oct. 2008.

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

[17] R. Karatay, B. Demir, A. A. Ergin, and E. Erkan, "A real-time eye movement-based computer interface for people with disabilities," *Smart Health*, vol. 34, Dec. 2024, Art. no. 100521. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352648324000771

[18] D. Cazzato, F. Dominio, R. Manduchi, and S. M. Castro, "Real-time gaze estimation via pupil center tracking," *J. Paladyn Behav. Robot.*, vol. 9, no. 1, pp. 6–18, Feb. 2018.

[19] Z. Wan, H. Zhang, M. Yang, Q. Wu, and S. Wu, "Self-calibrating gaze estimation via matching spatio-temporal reading patterns and eye-feature patterns," *IEEE Trans. Ind. Informat.*, vol. 20, no. 10, pp. 11697–11707, Oct. 2024.

[20] J. Sun, Z. Wu, H. Wang, P. Jing, and Y. Liu, "A novel integrated eye-tracking system with stereo stimuli for 3-D gaze estimation," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–15, 2023.

[21] L. Sun, Z. Liu, and M.-T. Sun, "Real time gaze estimation with a consumer depth camera," *Inf. Sci.*, vol. 320, pp. 346–360, Nov. 2015.

[22] J. Chen and Q. Ji, "A probabilistic approach to online eye gaze tracking without explicit personal calibration," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1076–1086, Mar. 2015.

[23] L. Sun, M. Song, Z. Liu, and M.-T. Sun, "Real-time gaze estimation with online calibration," *IEEE Multimedia Mag.*, vol. 21, no. 4, pp. 28–37, Oct. 2014.

[24] N. M. Arar, H. Gao, and J.-P. Thiran, "A regression-based user calibration framework for real-time gaze estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2623–2638, Dec. 2017.

[25] H. Zhang, L. Yin, and H. Zhang, "A real-time camera-based gaze-tracking system involving dual interactive modes and its application in gaming," *Multimedia Syst.*, vol. 30, no. 1, pp. 1–14, Feb. 2024.

[26] R. K. Megalingam, S. K. Manoharan, G. Riju, and S. M. Mohandas, "Netravaad: Interactive eye based communication system for people with speech issues," *IEEE Access*, vol. 12, pp. 69838–69852, 2024.

[27] D. Bisen, R. Shukla, N. Rajpoot, P. Maurya, A. K. Uttam, and S. K. Arjaria, "Responsive human–computer interaction model based on recognition of facial landmarks using machine learning algorithms," *Multimedia Tools Appl.*, vol. 81, no. 13, pp. 18011–18031, May 2022.

[28] N. A. Wani, R. Kumar, and J. Bedi, "DeepXplainer: An interpretable deep learning based approach for lung cancer detection using explainable artificial intelligence," *Comput. Methods Programs Biomed.*, vol. 243, Jan. 2024, Art. no. 107879.

[29] M. O. Taş and H. S. Yavuz, "A human–computer interaction system based on eye, eyebrow and head movements," *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 28, no. 5, pp. 632–642, 2022.

[30] P. Ramos, M. Zapata, K. Valencia, V. Vargas, and C. Ramos-Galarza, "Low-cost human–machine interface for computer control with facial landmark detection and voice commands," *Sensors*, vol. 22, no. 23, p. 9279, Nov. 2022.

[31] G. R. Chhimpa, A. Kumar, S. Garhwal, and Dhiraj, "Development of a real-time eye movement-based computer interface for communication with improved accuracy for disabled people under natural head movements," *J. Real-Time Image Process.*, vol. 20, no. 4, p. 81, Aug. 2023.

[32] R. G. Bozomitu, A. Păsărică, D. Tărniceriu, and C. Rotariu, "Development of an eye tracking-based human–computer interface for real-time applications," *Sensors*, vol. 19, no. 16, p. 3630, Aug. 2019.

[33] M. Ezzat, M. Maged, Y. Gamal, M. Adel, M. Alrahmawy, and S. El-Metwally, "Blink-to-live eye-based communication system for users with speech impairments," *Sci. Rep.*, vol. 13, no. 1, p. 7961, May 2023.

[34] M. P. Paing, A. Juhong, and C. Pintavirooj, "Design and development of an assistive system based on eye tracking," *Electronics*, vol. 11, no. 4, p. 535, Feb. 2022.

[35] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "InvisibleEye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–21, Sep. 2017.

[36] S. Mu, S. Shibata, K.-C. Chiu, T. Yamamoto, and T.-K. Liu, "Study on eye-gaze input interface based on deep learning using images obtained by multiple cameras," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108040.

[37] M. F. Ansari, P. Kasprowski, and P. Peer, "Person-specific gaze estimation from low-quality webcam images," *Sensors*, vol. 23, no. 8, p. 4138, Apr. 2023.

[38] K. Donuk, A. Ari, and D. Hanbay, "A CNN based real-time eye tracker for web mining applications," *Multimedia Tools Appl.*, vol. 81, no. 27, pp. 39103–39120, Nov. 2022.

[39] Y. Shi, Z. Zhang, K. Huang, W. Ma, and S. Tu, "Human–computer interaction based on face feature localization," *J. Vis. Commun. Image Represent.*, vol. 70, Jul. 2020, Art. no. 102740.

[40] M. Adnan, M. Sardaraz, M. Tahir, M. N. Dar, M. Alduailij, and M. Alduailij, "A robust framework for real-time iris landmarks detection using deep learning," *Appl. Sci.*, vol. 12, no. 11, p. 5700, Jun. 2022.

[41] Aman and A. Sangal, "Drowsy alarm system based on face landmarks detection using MediaPipe FaceMesh," in *Proc. 1st Int. Conf. Comput. Electron. Wireless Commun. (ICCWC)*. Cham, Switzerland: Springer, 2022, pp. 363–375.

[42] *MS Windows NT Kernel Description*. Accessed: Jun. 10, 2023. [Online]. Available: https://google.github.io/mediapipe/solutions/face_mesh.html

[43] A. Kar and P. Corcoran, "Performance evaluation strategies for eye gaze estimation systems with quantitative metrics and visualizations," *Sensors*, vol. 18, no. 9, p. 3151, Sep. 2018.

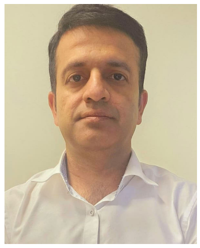
[44] K. Solska and T. Kocejko, "Eye-tracking everywhere–software supporting disabled people in interaction with computers," in *Proc. 15th Int. Conf. Hum. Syst. Interact. (HSI)*, Jul. 2022, pp. 1–5.

GOVIND RAM CHHIMPA is currently an Assistant Professor with the Department of Internet of Things and Intelligent Systems, Manipal University, Jaipur. He is a Research Scholar with the Thapar Institute of Engineering and Technology, Patiala, India, where he has submitted his Ph.D. thesis. His research interests include computer vision, object detection, eye and iris/pupil tracking, face recognition, real-time image and video processing, machine learning, deep learning, and human–computer interaction.

AJAY KUMAR received the Ph.D. degree in theoretical computer science from the Thapar Institute of Engineering and Technology, Patiala, Punjab, India. He is currently a Professor with the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology. His research interests include quantum computing, quantum software testing, and automata theory.

SUNITA GARHWAL received the Ph.D. degree in automata theory from the Thapar Institute of Engineering and Technology, Patiala, Punjab, India. She is currently an Associate Professor with the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology. Her research interests include software engineering, automata theory, and quantum computing.

DHIRAJ is currently a Principal Scientist with the CSIR-Central Electronics Engineering Research Institute (CSIR-CEERI), Pilani, Rajasthan, and the Government of India Research Laboratory. He is also holding an honorary position of Associate Professor of engineering sciences with the Academy of Scientific and Innovative Research (AcSIR), India. He has executed multiple research and development projects as a Project Coordinator/Principal Investigator/Co-Principal Investigator/Team Member-funded by leading industries, such as Samsung, and Krystal Vision, and Government Agencies, such as CSIR and DST. He has published more than 50 research articles in journals, conferences, and book chapters. His research interests include machine learning, deep learning, artificial intelligence, computer vision, and security.



FAHEEM KHAN received the Ph.D. degree from the University of Malakand, Pakistan. He was an Assistant Professor for four years in Pakistan. He is currently an Assistant Professor with the Department of Computer Engineering, Gachon University, South Korea. His research interests include computer networking, wireless networks, MANET, VANET, the IoT, and artificial intelligence. He is the Winner of the Best Researcher Award 2023 from Gachon University, South Korea.

YEON-KUG MOON received the B.S. and M.S. degrees in electronics engineering from Inha University, in 1998 and 2000, respectively, and the Ph.D. degree in bio-microsystem technology engineering from Korea University, in 2014. From 2002 to 2004, he with Samsung Electronics. From 2005 to 2023, he was the Director of the Data Convergence Platform Research Center, Korea Electronics Technology Institute. Since 2024, he has been an Associate Professor with the Department of Artificial Intelligence and Data Science, Sejong University, Seoul, Republic of Korea. His research interests include spatial platforms and emotional AI.