

Head Movement Invariant Eye Tracking System

R.M.T.S. Ratnayake, S.J. Sooriyaarachchi and C.D. Gamage

Abstract: The human visual framework, consisting of the eyes, optic nerve, and brain, gives us the sense of sight to capture and interact with our surroundings. The eye, which provides stimulus to generate sight, is a complex organ that collects light and effectively converts it to an electro-chemical impulse inside neurons.

Eye trackers are an important input modality for human-computer interfaces with vision-based applications. Eye-tracking systems have been proven to be valuable assistive devices for people with motor disabilities. Despite significant improvements in eye-tracking technology over time, current eye trackers in the market are complex and require intricate calibration procedures. Hence, there is still room for further research to develop an eye-tracking system with simpler calibration procedures.

This paper presents the development of a head movement invariant real-time eye-tracking system with simpler calibration procedures that can operate non-intrusively. The complexity of setting up the system is also significantly reduced. Experiments conducted with a sample set of users showed an eye gaze accuracy improvement of over 11% compared to the existing setups commonly available in the market.

Keywords: Eye-Tracking System, Pupil Detection, Glint Detection, Hough Transformation, Image Processing

1. Introduction


The human visual framework, consisting of the eyes, optic nerve, and brain, gives us the sense of sight to capture and interact with the surroundings. The eyes, which help as the sensory organ, play a significant role in the visual system by simulating the brain to create the necessary stimulus to generate sight in the brain. The eye is a complex organ that collects light and effectively converts it to an electro-chemical impulse inside neurons. The gaze point is the intersection of the gaze directions in the visual field that is imaged on the retina.

Eye trackers work as a necessary input modality for human-computer interfaces with vision-based applications and have been proven to be valuable assistive devices for people with motor disabilities. For instance, eye-tracking devices can work either as pointing/assistive devices for personal computers or TVs. Anderson et al. [2] have explained how eye tracking can be assistive in day-to-day life for standard control inputs like select, zoom, and scroll. Betke et al. [3] have shown a gaze-based typing method for people with motor disabilities. Compelling use cases of eye tracking systems can be found in studies of attention function, Mode disorder, Training of pilots, and Telemarketing.

Current eye-tracking systems have sophisticated hardware and software to achieve high accuracy when tracking the eye gaze. They include near infra-red (NIR) LEDs, cameras, and dedicated processing units. Despite the current development in eye-tracking technology, the contemporary eye trackers in the market are complex and require complex calibration procedures. For example, the lengthy 16-point calibration process in present eye-tracking systems presents challenges for disabled individuals, requiring them to sustain their attention span for an extended duration during calibration [3]. Hence, this research is aimed to develop a head movement invariant eye-tracking system with simpler calibration procedures.


Eng. R.M.T.S. Ratnayake, AMIE(SL), B.Sc. Eng. Hons (Moratuwa),

Email: tharakasachintha.17@cse.mrt.ac.lk

 <https://orcid.org/0009-0004-6408-7587>


Eng. (Dr.) S.J. Sooriyaarachchi, AMIE(SL), MACM, B.Sc. Eng. (Hons) (Peradeniya), M.Sc. (Moratuwa), Ph.D.

(Moratuwa), Senior Lecturer, Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. Email: sulochanas@cse.mrt.ac.lk

 <https://orcid.org/0000-0003-0905-7196>

Eng. (Prof.) C.D. Gamage, C.Eng., MIE(SL), MACM, B.Sc. Eng. Hons (Moratuwa), M.Eng. (AIT), Ph.D. (Monash), Professor in Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Email: chandag@uom.lk

 <https://orcid.org/0000-0002-8182-812X>



2. Literature Review

The main six branches of research related to eye-tracking systems are image acquisition, eye model, glint detection algorithm, pupil detection algorithm, calibration algorithm, and eye-tracking algorithm. Image acquisition focuses on image capturing using sophisticated as well as off-the shelf cameras. Eye models focus on the representation of the eye, a related movement, and eye extraction from the captured images. The glint detection algorithms focus on the accurate prediction of the location of the glints in the eye image. Pupil detection algorithms focus on the accurate detection of the pupil center. Calibration algorithms focus on hardware calibration and user calibration to accurately calculate relevant constants. Finally, the eye-tracking algorithms focus on fusing glint location data, pupil center location data, and calibration data to estimate the point of gaze.

2.1 Image Acquisition

Video based eye-tracking systems were used in airplanes in the early 40s [1]. Earlier, the eye-tracking technology was only used in academia and industry. With the rapid development of low-cost yet powerful computing hardware, eye-tracking systems became familiar to general users [6]. The present market-leading eye-tracking systems leverage NIR based hardware and processing systems while research focuses on both NIR and off-the-shelf cameras [14].

Corcoran et al. [10] have used a video feed from a low-resolution user-facing VGA camera for gaze estimation and achieved 4 deg accuracy while allowing a small head movement. Ramadan et al. [15] have used a web camera and achieved a 2.5 deg accuracy with no head movement. Huang et al. [24] have achieved an eye gaze estimation of 0.3-0.4 degree, allowing head movement in an IR camera-based setup. Zhange et al. [25] have achieved an eye gaze estimation of 0.3 degrees allowing free head movement using binocular fixation constraint. But the above systems lack variable illumination to generate sufficient glint outside the lab environment.

The number of infra-red (IR) LEDs and cameras used in the lighting setup depends on the eye model and the eye-tracking algorithm used in the eye-tracking system. But the accuracy increases with the number of cameras and LEDs. Guestrin et al. [7] show that the increased number of cameras will achieve better accuracy

in the 3D model-based approach. This is because more cameras and NIR LEDs collect more data and features about the system.

Hennessey et al. [20] show that increasing the number of NIR LEDs in the cross ratio-based model will increase the accuracy of the eye-tracking system. This is because each NIR LED will act as a calibration point for the eye-tracking system. Increasing the calibration points will directly increase the accuracy of the eye-tracking system.

2.2 Eye Models

Eye-tracking algorithms which use NIR are based on corneal reflection. In contrast, visible light-based algorithms use content information like shape and features for gaze estimation. NIR-based algorithms can be further classified into 2D regression models, 3D models, and cross-ratio-based models. Visible light-based algorithms can be further classified into appearance-based models and shape-based models.

In appearance-based models, features are extracted from the eye images. Then a model is trained with the extracted features. A statistical model is utilized to describe shape and texture differences in Bacivarov [8]. Tang et al. [11] have proposed an appearance-based model based on the Support Vector Machines (SVM), where gaze direction is estimated from feature points. Koutras et al. [26] have presented an Active Appearance Model (AAM), which uses low-resolution videos to predict eye gaze. In the proposed model, AAM identifies the eye region using feature points. In recent research, Deep Learning (DL) models have been used for gaze estimation. George et al. [30] have used a Convolutional Neural Network (CNN) based model to identify seven gaze directions. Konrad et al. [31] have proposed an eye-tracking model, robust to head movement, using a CNN model.

Shape-based methods use a template that is deformable to identify the eye. In the process, similarity between the eye image and the template is calculated using a cross-correlation. The template consists of two parabolas for eye contours and a circle for the iris.

In 2D regression-based algorithms the vectors between the pupil centre and corneal glints are mapped to matching gaze locations using a polynomial transformation function. In the transformation function, the coefficients are estimated using calibration procedures. Cherifetal. [16] used a transformation of higher order to map gaze vectors to the screen by minimizing the mean square error. It also

measured how the head movement would affect system accuracy using a 5x5 point calibration scheme. Zhu et al. [18] proposed a new algorithm to reduce the accuracy barrier in 2D models using a subpixel tracking method. The proposed algorithm has the capability to detect the inner eye corner and the centre of the iris at subpixel accuracy. Cerrolaza et al. [21] developed a taxonomical classification on system accuracy vs. mapping equation using 400,000 models, and presented the methods to optimize calibration scheme halving the calibration time allowing head movement.

2.3 Pupil and Glint Detection Algorithms

Post processing is needed to detect pupil centre and glint centre locations after the images are captured. The accuracy of the eye tracking system heavily depends on the accurate measurement of the pupil centre and glint centres. However, accurate pupil detection could be difficult due to closed eyelids, bright artifacts, shadows, etc. The pupil detection is done using edge detection schemes.

Pupil and glint detection algorithms must be simple and efficient for real-time performance. Excessive processing and complex algorithms could slightly increase accuracy but require higher processing power and computational time. Hence the trade-off between simplicity and accuracy must be considered when designing pupil and glint detection algorithms.

Pupil Detection

Hennessey et al. [20] have used an image differentiation approach to help in recognizing the pupil shape. In the proposed method, subtracting the dark pupil from the bright pupil makes contour detection easier.

The software-based pupil detection algorithms include eclipse fitting, Hough transform, etc. Pasarica et al. [27] have proposed two pupil detection algorithms based on the circular Hough transform and Starburst algorithm. In [29], the eye-tracking system uses hardware support to identify the center of the pupil using the bright and dark eye effect.

Glint Detection

The glint, also known as the Purkinje image, is a point of corneal reflection of light falling from an outside source onto the pupil. Using conventional edge detectors with various thresholds, the glint region is determined. The eye-gaze tracking system is heavily reliant on the accurate detection of the iris, pupil, and glint [4]. Sharma et al. [13] have evaluated the Sobel, Canny, and Laplacian of Gaussian (LoG)

edge detection algorithms using different parametric variations. The results suggested that different domains of images require different parameter values for optimal results. Hennessey et al. [20] have used simple thresholding techniques to identify glint locations. Possible errors have been rejected using the expected displacement between glint centers.

2.4 Calibration Algorithms

In the 3D model, user specific parameters like the center of corneal curvature, the pupil center, pupil radius are calculated in the calibration procedure [7]. In the cross-ratio-based model, errors that occurred due to planarity assumptions are minimized using the calibration scheme. The 2D regression model calibration is used to determine the coefficient of the polynomial function.

In the calibration procedure, users are asked to look at multiple points on the screen. The number of points in the calibration procedure depends on the eye model used. Also, the number of calibration points can be small as 1 and large as 25. Guestrin et al. [7] have pointed out that the convenience of the calibration technique comes at the expense of system complexity. Guestrin et al. [22] proposed a single point calibration for infants.

Therefore, a calibration technique that depends on a single point for fixation may be less robust than a calibration procedure that uses multiple points.

2.5 Eye-Tracking Algorithm

The eye-tracking steps include image acquisition, pupil detection, glint detection, error correction, and gaze mapping. Error correction is done using the parameters found in the calibration procedure. Gaze mapping is done using eye-tracking algorithms.

3D model-based approach first determines the optical axis from pupil location and glint locations. Guestrin et al. [7] used the optical axis to determine the visual axis using calibration data.

The cross-ratio-based method maps the polygon created by the corneal reflection to the screen. The assumption made here that the polygon is planar, is a wrong assumption because of the curvature in the corneal surface. The error that occurred due to the planarity assumption is fixed using the intermediate normalizing phase. First, the polygon is normalized based on the calibration parameters. Then the normalized space is

mapped to the screen plane [20] as a scaling operation.

2.5.1 3D Model Based Method

3D model-based methods use geometrical (spherical or ellipsoidal) eye models to estimate the corneal center, optical axis, and visual axis. Since the visual axis cannot be calculated directly, the optical axis is estimated using single or multiple cameras. The 3D model-based multiple-lights multiple-camera approach has shown high accuracy and resistance to head movements but requires a complex calibration of cameras to determine the geometric properties of the monitor and cameras [7]. Guestrin et al. [22] and Shih and Liu [5] have proposed a 3D model-based eye-tracking method with single-point calibration. With high accuracy, free head movements and single-point calibration, 3D-based models are suitable for developing a head movement invariant eye-tracking system. Guestrin et al. [7] have shown that the 3D model-based eye-tracking system can be made using single-camera single-light source, single-camera multiple-light sources, and multiple-camera multiple-light sources, but each limiting some capabilities. The system of equations in [7] demonstrates the accuracy limitations of the single-camera single-light source configuration when the head is moving. Hence, when using a single-camera single-light source, the user must keep the head still. Therefore, this method cannot create a head movement invariant eye-tracking system. Guestrin et al. [7] have shown the need to feed the user-specific constants like the radius of the pupil to the algorithm and the configuration allows significant head movement when using multiple-light sources and single-camera setup.

The most potent form of the 3D model-based eye-tracking system is multiple-camera multiple-light-source configuration [7]. This configuration does not need constant feeding, and all required will be calculated when user calibration is carried out. But the disadvantage is that the configuration has complex setup complexity and requires a lot of hardware compared to other methods.

2.5.2 Cross-Ratio Based Method

Cross-ratio based method uses immutable properties of projective geometry by projecting a rectangular NIR pattern on the user's eye. The pupil location and screen size are used to estimate the gaze direction [28]. The cross-ratio methods do not need hardware calibration or an eye model, but allows free head motions.

Karr et al. [14] have shown a four-point simple calibration procedure associated with the cross-ratio method. Hence, the cross-ratio model is also suitable for developing a head movement invariant eye-tracking system. Figure 1 shows the ray tracing diagram of the cross ratio-based model proposed by Huang et al. [24].

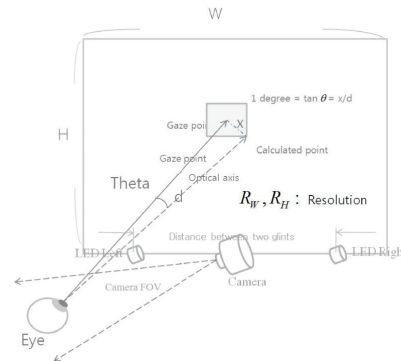


Figure 1 – Ray Tracing Diagram of Cross-Ratio Based Model Proposed by Huang et al. [24]

2.6 Eye-tracking Model Selection

Developing hardware for both systems will be time consuming and costly. Hence, to identify best method from two approaches, a simulation-based evaluation was carried out.

It was chosen not to simulate the image analysis step explicitly in the simulation framework to simplify the simulation. That is, instead of computing the image seen by each camera using 3D rendering algorithms and extracting the position of relevant features from the image, the positions where these features will lie in the camera image, given the spatial positions of the eye, camera, and lights are computed. The effect of finite camera resolution and inaccuracies in the image analysis are simulated. Simulating finite camera resolution involved employing a pinhole camera model and subsequently excluding features outside the screen's designated region.

Test cases were created to cover two head positions with four LEDs with varying calibration points for the simulation. The reason for using four LEDs was that both cross-ratio based methods and 3D model-based methods gave the best results in theoretical approximations in minimum hardware presence. In practical scenario, the cross-ratio method requires four LEDs or four cameras to generate the polygon mapping [24].

The results obtained from simulation are summarized in Table 1. The table shows the number of calibration points, maximum error, mean error and standard deviation between

actual and predicted gaze point in millimeter scale of each eye tracking algorithm.

Table 1 – Simulation Results

# of calibration points	Cross ratio-based method			3D model-based method		
	Max err (mm)	μ err (mm)	σ^2 err (mm)	Max err (mm)	μ err (mm)	σ^2 err (mm)
4	447	249	124	450	230	111
5	718	196	168	340	150	112
9	6.17	6.27	4.19	45	16	12.5
13	5.2	3.42	3.42	9	5.6	2.73

The accuracy increases in both the cross-ratio and the 3D model-based methods when the number of calibration points increases. Considering the values, it can be assumed that both methods have the same accuracy capability.

In the 3D model-based technique, it is needed to measure all the geometric distances and feed them to the algorithm. Upon changing camera position or light position, it is needed to measure all the values again. Even though both systems have the same calibration procedures and accuracy levels, the cross ratio-based method has a simple setup complexity. Therefore, to build an eye-tracking system the cross ratio-based model was chosen.

3. Methodology

During the literature review it was found that the accuracy of appearance-based, shape-based and 2D regression-based models are not sufficient for commercial eye-tracking systems. Choosing between 3D model-based system or a cross-ratio-based system for eye-tracking model was determined through the previous simulation.

Finally, the built experimental design was assessed for its intrusiveness, real-time processing capability, accuracy, complexity, and calibration procedure as the key parameters.

3.1 Low Complexity Tracking System Design

The system is expected to capture images in various lighting conditions and the system should illuminate the user's eye with NIR LEDs to generate glints in the eyes. NIR LEDs should be able to switch on and off in high frequency (30Hz) to generate illuminated and non-illuminated eye images. The eye should be captured with an IR camera and captured images should be passed on to the processing module. Because of these requirements the eye

tracking system has a LED light system, an IR camera system and a processing system as subsystems in the hardware system. Each of the subsystems has been explained in the following section.

3.1.1 NIR LED Light Module

From the literature review it was found that in the cross ratio-based model, NIR LEDs can be used as markers in the IR cameras setup, since the users will not see the emission of light in the IR spectrum. To achieve variable illumination, an experiment was carried out to determine the number of required NIR LEDs in different lighting schemes.

3.1.2 IR Camera Module

During the literature review it was found that the resolution of the IR camera directly impacts the accuracy of the eye tracking system. Higher resolution cameras will capture high resolution images with very accurate glint features thus allowing higher accuracy. But the drawback of the higher resolution camera is that the processing module will require high processing power for the real-time processing of the image. Hence it is required to determine the optimum resolution, bit depth and frame rate for the camera.

3.1.3 Processing Module

During the literature review it was found that the processing module should have the capability to preprocess images at the edge and switch on/off the LEDs. The processing module should be able to apply simple preprocessing filters like gaussian filter and thresholding filter with low latency. Switching of LEDs should be done at a frequency governed by the glint detection algorithm.

3.2 High Performance Software-based Eye-Tracking

During the literature review it was identified that the cross ratio-based method is the most suitable eye model to build the eye-tracking system. The software module should work along with the hardware module to capture images and store images in a processing friendly data structure, process the images in real-time, and output the point of gaze. When designing the software module, four different aspects were considered.

But the center of the pupil can also be located using software only means, explained in Section 2.3. In the presented system, it was decided to use a software only pupil detection scheme because it allows for eye-tracking

systems that are not tightly coupled to the hardware.

The literature review identified the processing pipeline should have four stages. These pipeline stages include capturing images, detecting the pupil center, detecting the glint locations, and estimating gaze point.

Hennessy et al. [20] propose that the algorithms used for the eye-tracking system should be simple to achieve real-time performance. The time saved from one stage can be used in another stage. For example, the time saved at the image capture module can be used for a complex pupil detection algorithm. Hence making modules as efficient as possible is a must. Finally, the implementation should provide a simple interface such that it will be easy to plug a new algorithm.

3.2.1 Video Capture Module

This video capture module is expected to run on the processing hardware. The module should implement:

- A mechanism to capture images in real-time,
- Decode the images if the camera uses a compression scheme, and
- Store images in a processing friendly format.

A multi-threaded environment is required to capture images in real-time. The captured images should then be pushed to a queue so that the next stage module can acquire the images from the queue without any blocking in processing.

3.2.2 Glint Tracking Module

This module is expected to obtain the images from the video capture. During the literature review it was found that the glint tracking module can use the algorithm proposed by Hennessey et al. [20] that has used simple thresholding techniques to identify glint locations.

3.2.3 Pupil Tracking Module

This module is expected to obtain the images from the video capture. The literature review found that Pasarica et al. [27] have proposed two pupil detection algorithms based on the circular Hough transform (CHT) and Startdust Algorithm. CHT is a derivative of the Hough Transform, which works with less search space computed in preprocessing steps. Starburst uses a probabilistic model to reduce the search space.

3.2.4 Eye Tracking Module

This module is expected to communicate with both the pupil tracking and glint tracking modules. The module should implement:

- 1) The calibration mechanism
- 2) The prediction mechanism

The literature review found that users are asked to look at multiple points on the screen in the calibration procedure. The collected data should be used to derive the calibration function. The literature review found that the cross-ratio-based method maps the polygon created by the corneal reflection to the screen. The required algorithm should be implemented in the module to estimate the point of gaze.

4. System Design

The hardware for the eye-tracking system was developed in a modular and extensible manner. The essential hardware components were derived from the existing research and then extended to add new capabilities. The prototype was built on a breadboard. Later the circuit was developed to a dedicated PCB board. The overall architecture of the design is shown in Figure 2.

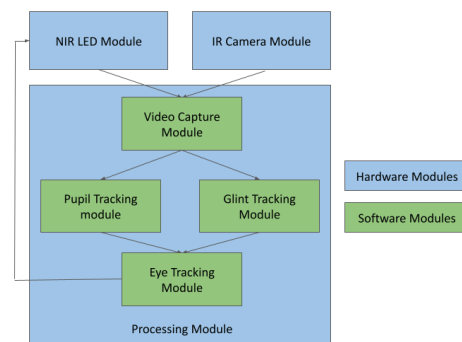


Figure 2 - Overall System Design of the Implementation

4.1 Camera Resolution

Yoo et al. [17] suggested that the polygon size created by glints in the eyes should be maximized when using a camera to capture the image.

In this research, the standard screen size 1920pix * 1080pix was used where pix denotes pixels. The values were derived using 25pix which is a reasonable minimal resolution for an eye-tracking system with medium-level accuracy.

Width of the polygon -> $(1920/25)$ pix
 = 76.8 or 77units
 Height of the polygon -> $(1080/25)$
 pix
 = 43.2 or 44units

At a minimum, the glint can be represented in a single pixel in the image. Hence the 77×44 unit can be mapped to $77\text{pix} \times 44\text{pix}$ in a camera. Assuming the upper bound, a polygon of size of $80 \times 50\text{pix}$ can be used, which should deliver a quite reasonable resolution. Li et al [9] suggest the average pupil width of Asian population to be $4.73 \pm 0.85\text{mm}$ and $4.30 \pm 1.02\text{mm}$ in Caucasian population. For the calculations, the lower bound 4.30mm is considered (excluding variations in pupil size attributable to age).

Vasanthakumar et al. [12] suggest that the Outer canthal distance (OCD) of Asian population as $95.55 \pm 6.39\text{cm}$ for men and $92.44 \pm 5.71\text{cm}$ for women and Palpebral fissure width as $31.08 \pm 1.79\text{cm}$ for men and $29.90 \pm 2.18\text{cm}$ for women.

The cross-ratio method proved that the head movement invariance is preserved if all the glints are inside the pupil. Hence $80\text{pix} \times 50\text{pix}$ polygon should be able to put inside the 4.30mm pupil.

Considering a setup without magnifier lenses, the following can be calculated.

Capturing only one eye, Palpebral fissure width is 32mm , suggesting $80 \times (32/4.30)$ pixels in the camera width for minimum accuracy. This refers to a 600pix camera. Capturing both eyes, Outer Canthal distance is 95mm , suggesting $80 \times (95/4.30)$ pixels in the camera width for minimum accuracy. This refers to a 1767pix camera. Hence a $1920\text{pix} \times 1080\text{pix}$ camera was chosen in the research.

4.2 Frames per Second

According to the Nyquist sampling theorem it is needed to capture at least every 25-300ms interval to reconstruct the fixations which have an average duration of 50-600ms. In this research a 30fps camera was used even if the calculations yield to a 3-40fps camera. The schematic and the PCB were designed using open source KiCad software.

A Raspberry pi header module was created, which can be plugged into the Raspberry pi3,3B, 4 boards which can supply current to the LEDs and communicate with the PC for switching on and off LEDs.

4.3 Software Modules

4.3.1 Video Capture Module

The video capture module has three different functional requirements. The module gives an easy-to-implement interface to introduce a new camera to the system.

In implementation, a videoGet module was implemented as a separate thread for the non-blocking operation of the main user interface (UI) on a QThread. The single-threaded performance reached a maximum of 13FPS capturing a 1080P video stream while the multi-thread implementation reached a maximum of 41FPS capturing in the same camera. For the real-time operation of the 30FPS requirement, this implementation is proven to be sufficient. The videoGet interface was implemented such that it has the necessary functionality to continuously capture images, get a reduced scale preview of the image and get the actual image. The images are stored in row-major order to facilitate vector operations like vector addition and vector multiplication for image processing operations in the following modules.

4.3.2 Pupil Capture Module

The pupil module has three different functionalities. The module detects the pupil from the image extracted by the videoGet module and extracts the pupil center from the detected pupil. If necessary, the module calibrates the pupil center data. An interface is provided to implement different pupil capture algorithms in the implementation. Also, two different pupil detection algorithms were implemented based on contour detection and Hough transform (inspired by Pasarica et al. [27]). For simplicity, only the Hough transform based pupil detection algorithm is explained.

First, the median filter is applied to the image to remove any salt and pepper noise that could interfere with the Hough transform. Then the circular Hough transform is applied to identify potential pupil candidates in the image. When using the circular Hough transform, the minimum and maximum radii are calculated based on the adaptive thresholding mechanism specified in the calibration function.

The domain knowledge of the pupil is used to extract the exact pupil from the candidate pupils. When the eye is captured from an IR camera, the pupil has a darker region relative to the background. Hence, when calculating the potential pupil candidates' mean, the actual pupil should have the minimum intensity value.

In the calibration process, an adaptive method is used to find the minimum and maximum radius of the pupil. The initial radius values are set to a nominal value corresponding to a man

sitting in front of the desktop 60 cm away. The values are changed if the algorithm cannot find the pupil in the image. Finding minimum and maximum is necessary to reduce the computational time since the Circular Hough Transform is a computationally extensive time-consuming task. The minimum and maximum radii were used to reduce the domain space in the Hough transformation.

The Hough transformation steps are represented in a flowchart in Figure 3.

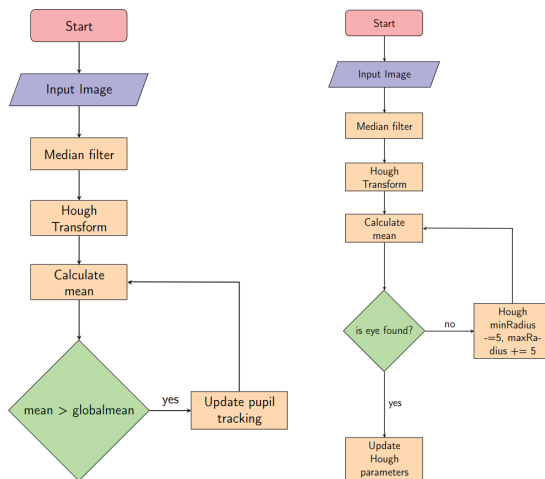


Figure 3 - Pupil Capturing Process

4.3.3 Glint Capture Module

The glint capture module has three different functionalities. The module detects the glints and extracts the glint centers from the image extracted by the videoGet module. If necessary, the module calibrates the glint center data. The pupil center data extracted from the previous Pupil Capture module were used in this algorithm. Because of this dependency, the eye tracking algorithm must run the Pupil Capture module and Glint Capture module in sequential order.

When looking at a point in the screen, the pupil center should be inside a polygon created by the glint locations. This domain knowledge was used to reduce the search space for glint identification. First, a region of interest (ROI) is selected based on the pupil center detected in the previous module. Then the contour detection algorithm is applied to identify the glints with high-intensity values. Considering the glint as a blob, the eye tracking system uses image moment to find the exact center of the glint locations. This process needs to identify at least four glint locations by the constraints in

the cross-ratio-based method. Hence only the glint data with four or more glint locations are used for the eye-tracking module.

The glint capture steps are represented in a flowchart in Figure 4.

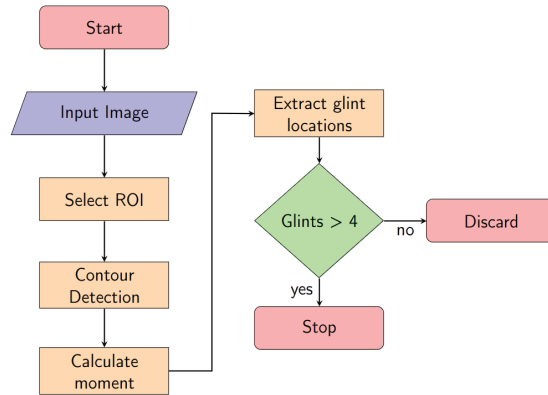


Figure 4 - Glint Capturing Process

4.3.4 Eye Tracking Module

The eye-tracking module has three different functionalities. The module calculates the homography matrix in the calibration phase, normalizes the configuration space and estimates the gaze using previously found pupil and glint locations.

The data collected in the calibration procedure is used to calculate the homography matrix as in the eye-tracking algorithm in the literature review. In the prediction phase, the eye-tracking module uses the pupil center data found from the pupil capture module and glint center data found from the glint capture module. First, a polygon is generated from the glint location data which is the configuration space for eye-tracking space. The gaze point is estimated after the polygon is normalized to match the actual monitor task space.

5. Evaluation

The cross ratio-based eye tracking system was evaluated in a user experiment with a group of 6 people. When selecting subjects, they were selected such that 3 male and 3 female participants with eye colors dark brown, light brown, and black pupil colors. All subjects are aged 19 to 60 years old. The monitor, lights, and screen were fixed 60 cm from the subject. Each subject went through a 9-point calibration scheme. Software built on Qt framework was used for subject calibration and acquisition of data. Each calibration point was shown 6s in the screen to capture 180 frames (at 30 frames per second) and 100 data points were selected

using the least square error. These points were used for the calibration procedure.

Figure 5 shows the actual experimental setup used. The subjects were asked to position the head such that the monitor's top line is in line with the subject's eye. This position is considered as (0,0, 600) eye location. In subsequent data acquisitions, the subjects were asked to tilt their head to get left, right (8cm) vertically (5cm) forward and backward (5cm) tilting head positions before the field of view is lost.



Figure 5 –Experimental Setup and Corneal Reflection when looking at Corners

A total of 60 frames (2s) were captured of each fifteen randomly generated locations on the screen. The image capturing started when the subject pressed the space button. This was done to avoid image capturing while eye movement is transitioning from the previous location to the new location.

To estimate the point of gaze, the captured images were stored using the video capture module in memory data-structure. Then the pupil capture module and glint capture module extracted the features from the eye images. Finally, the eye-tracking module calculated the point of gaze using a homography matrix and displayed the point of gaze in the display.

The evaluation was conducted utilizing the data gathered from the previously described configuration. The accuracy evaluation involved computing pixel differences between randomly generated points on the screen and their corresponding predictions in our algorithm. Subsequently, these pixel differences were translated into angle differences, taking into account the head position of the participant.

The setup and the algorithm demonstrated a weighted 11% improvement in accuracy compared to the commonly used homography setup (Table 2) in different head positions [3]. The weighting was based on a Gaussian distribution, considering that users exhibit

fewer head poses in the closest and farthest positions, while the normal pose is more prevalent.

Table 2 – Accuracy of Proposed Algorithm

HeadPose	Homography[3]	ProposedAlgorithm	
	Meanerror(degrees)	Meanerror(degrees)	Stddev(degrees)
(0,0,500)	0.95	0.64	0.07
(0,0,600)	0.96	0.71	0.08
(0,0,700)	0.80	0.78	0.10
(0,0,800)	1.14	0.89	0.11

Figure 6 shows the actual points and the estimated points in the experiment.

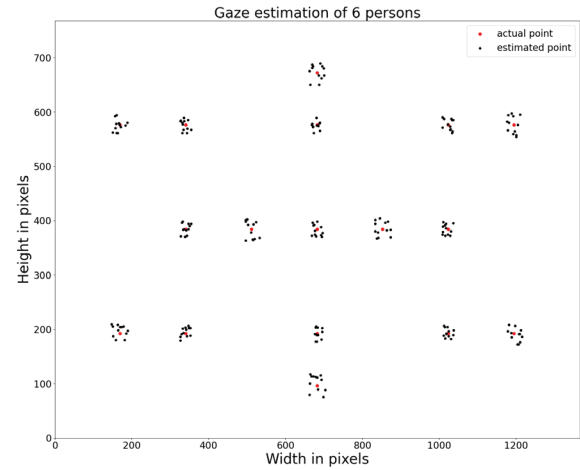


Figure 6 – Actual and Estimated Points

Table 3 shows the accuracy comparison with prominent existing setups. Our implementation outperforms the other cross based methods but lags behind due to hardware limitations [17]. The 3D model-based implementation in [6] is much superior but requires additional calibration points compared to ours.

Table 3 – Accuracy of Eye-Tracking System

Ref	# of lights	# of cameras	Head movement	Accuracy [%]	Categories
[6]	2	2	Moderate	1	3D
[19]	2	4	Large	0.6	3D
[23]	1	4	Large	3.5	cross
[17]	5	2	Moderate	0.98	cross
[25]	8	1	Slight	0.3	cross
Our	4	1	Moderate	0.88	cross

The improvements of processing time due to the proposed pupil detection algorithm was measured using ai 5 intel 2.4 GHz 8 GB RAM computer. The saved time from the pupil detection algorithm can be used for a complex eye-tracking algorithm which should yield better results.

6. Conclusion

In this research, a non-intrusive, real time head movement invariant eye-tracking system with high accuracy, simpler setup complexity, and a simplified calibration procedure is presented. The eye tracking system is evaluated with various head movements. The results show that the proposed system has parallel accuracy with the state-of-the-art eye tracking system. The eye-tracking system is designed in a modular manner that can be further developed independently.

Pre-processing can be done in the hardware module further reducing the computation time. This edge processing in the hardware module itself will allow it to capture high- quality images and transfer images to the computer without a bandwidth issue. In the software module pupil capture module, glint capture module and eye-tracking module can be used to develop novel algorithms in the future. The eye-tracking system can be used in other research areas such as attention tracking and human computer interaction.

References

1. Mohamed, A., Da Silva, M. P., and Courboulay, V., "A History of Eye Gaze Tracking," 12 2007.
2. Salvucci, D. D., and Anderson, J. R. "Automated Eye-Movement Protocol Analysis," *Human Computer Interact.*, 16(1):39-86, mar 2001.
3. Betke, M., Gips, J., and Fleming, P., "The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People with Severe Disabilities," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 10(1):1-10, 2002.
4. Zhu, Z., and Ji, Q., "Eye and Gaze Tracking for Interactive Graphic Display," *Mach. Vision Appl.*, 15(3):139-148, Jul 2004.
5. Shih, S.-W., and Liu, J., "A Novel Approach to 3-d Gaze Tracking using Stereo Cameras," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):234-245, 2004.
6. Morimoto, C.H., and Mimica, M.R., "Eye Gaze Tracking Techniques for Interactive Applications," *Computer Vision and Image Understanding*, 98(1):4-24, 2005, Special Issue on Eye Detection and Tracking.
7. Guestrin, E., and Eizenman, M., "General Theory of Remote Gaze Estimation using the Pupil Center and Corneal Reflections," *IEEE Trans. on Biomedical Engineering*, Vol. 53, pp. 1124-1133, 2006.
8. Bacivarov, I., Ionita, M., and Corcoran, P., "Statistical Models of Appearance for Eye Tracking and Eye-Blink Detection and Measurement," *IEEE Trans. on Consumer Electronics*, 54(3):1312-1320, 2008.
9. Yan, Li., and Huang, David, "Pupil Size and Iris Thickness Difference Between Asians and Caucasians Measured by Optical Coherence Tomography," *Investigative Ophthalmology & Visual Science* 50 (2009): 5785-5785.
10. Corcoran, P. M., Nanu, F., Petrescu, S., and Bigioi, P., "Real-Time Eye Gaze Tracking for Gaming Design and Consumer Electronics Systems," *IEEE Trans. on Consumer Electronics*, 58(2):347-355, 2012.
11. Wu, Y.-L., Yeh, C.-T., Hung, W.-C., and Tang, C.-Y., "Gaze Direction Estimation using Support Vector Machine with Active Appearance Model," *Multimedia Tools and Applications*, Vol. 70(3), 06 2012.
12. Vasanthakumar, P., Kumar, P. and Rao, M., (2013). Anthropometric Analysis of Palpebral Fissure Dimensions and its Position in South Indian Ethnic Adults. *Oman Medical Journal*, 28(1):26-32.
13. Sharma, A., and Abrol, P., "Glint Detection and Evaluation using Edge Detectors," *International Journal of Scientific and Technical Advancements*, Vol. 1, pp. 319-323, 01 2015.
14. Kar, A., and Corcoran, P., "A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms," *IEEE Access*, Vol., pp. 1-1, 08 2017.
15. Ramadan, S., Abd-Elmageed, W., and Smith, C. E., "Eye Tracking using Active Deformable Models," in *ICVGIP*, 2002.
16. Cherif, Z., Nait-Ali, A., Motsch, J., and Krebs, M., "An Adaptive Calibration of an Infrared Light Device used for Gaze Tracking," in *IMTC/2002. Proc. of the 19th IEEE Instrumentation and Measurement Technology Conf.*, Vol. 2, 2002, pp. 1029-1033.
17. Yoo, D. H., Kim, J. H., Lee, B. R., and Chung, M. J., "Non-Contact Eye Gaze Tracking System by Mapping of Corneal reflections," in *Proc. of 5th IEEE Int. Conf. on Automatic Face Gesture Recognition*, 2002, pp. 101-106.
18. Zhu, J. and Yang, J., "Subpixel Eye Gaze Tracking," in *Proc. of 5th IEEE Int. Conf. on Automatic Face Gesture Recognition*, 2002, pp. 131-136.
19. Beymer, D., and Flickner, M., "Eye Gaze Tracking using an Active Stereo Head," in *2003 IEEE Comp. Soc. Conf. on CVPR*, 2003. Proc., Vol. 2, 2003, pp. II-451.
20. Hennessey, C., Nouredin, B., and Lawrence, P., "A Single Camera Eye-Gaze Tracking

- System with Free Head Motion," ser. *ETRA '06*. pp. 87-94.
21. Cerrolaza, J. J., Villanueva, A., and Cabeza, R., "Taxonomic Study of Polynomial Regressions Applied to the Calibration of Video-Oculographic Systems," in *Proc. of the 2008 Symp. on Eye Tracking Research & Applications*, ser. *ETRA*.
 22. Guestrin, E. D., and Eizenman, M., "Remote Point-of-Gaze Estimation Requiring a Single-Point Calibration for Applications with Infants," ser. *ETRA '08*.
 23. Hansen, D., Agustin, J., and Villanueva, A., "Homography Normalization for Robust Gaze Estimation in Uncalibrated Setups," 2010, pp. 13-20.
 24. Huang, J.-B., Cai, Q., Liu, Z., Ahuja, N., and Zhang, Z., "Towards Accurate and Robust Cross-Ratio Based Gaze Trackers through Learning from Simulation," in *Proc. of the Symp. on Eye Tracking Research and Applications*, ser. *ETRA '14*. pp. 75-82.
 25. Zhang, Z., and Cai, Q., "Improving Cross-Ratio-Based Eye Tracking Techniques by Leveraging the Binocular Fixation Constraint," in *Proc. of the Symp. on Eye Tracking Research and Applications*, ser. *ETRA '14*. pp. 267-270.
 26. Koutras, P., and Maragos, P., "Estimation of Eye Gaze Direction Angles Based on Active Appearance Models," 2015 *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 2424-2428, 2015.
 27. Pasarica, A., Bozomitu, R. G., Cehan, V., Lupu, R. G., and Rotariu, C., "Pupil Detection Algorithms for Eye Tracking Applications," in *2015 IEEE 21st International Symp. for Design and Technology in Electronic Packaging (SIITME)*, 2015, pp. 161-164.
 28. Park, J., Jung, T., and Yim, K., "Implementation of An Eye Gaze Tracking System for the Disabled People," in *2015 IEEE 29th Int. Conf. on Advanced Information Networking and Applications*, 2015, pp. 904-908.
 29. Shin, Y.-G., Choi, K.-A., Kim, S.-T., Yoo, C.-H., and Ko, S.-J., "A Novel 2-d Mapping-Based Remote Eye Gaze Tracking Method using Two irLight Sources," in *2015 IEEE Int. Conf. on Consumer Electronics (ICCE)*, 2015, pp. 190-191.
 30. George, A. and Routray, A., "Real-Time Eye Gaze Direction Classification using Convolutional Neural Network," 06 2016, pp. 1-5.
 31. Konrad, R., Shrestha, S., and Varma, P., "Near-Eye Display Gaze Tracking via Convolutional Neural Networks," 2016.

