BTECH MAIN PROJECT REPORT

EYE MOVEMENT BASED HUMAN COMPUTER INTERACTION

ATHIRA M D

FAHEEM HABEEB

KIRTHY FRANCIS

YAMINI ANIL

A PROJECT REPORT ON

EYE MOVEMENT BASED HUMAN COMPUTER INTERACTION

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

By

Athira M D

Faheem Habeeb

Kirthy Francis

Yamini Anil

Under the Guidance of

Mrs. Mary John

Assistant Professor

Dept. of Information Technology

RSET

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

KOCHI –682039

APRIL 2017

DEPARTMENT OF INFORMATION TECHNOLOGY
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
KOCHI –682039

CERTIFICATE

Certified that project work entitled "**Eye Movement Based Human Computer Interaction**" is a bonafide work carried out by **Athira M D(14013284), Faheem Habeeb(14013292), Kirthy Francis(14013306), Yamini Anil(14013335)** in partial fulfillment for the award of Bachelor of Technology in Information Technologyfrom Mahatma Gandhi University, Kottayam, Kerala during the academic year 2014-2018

Prof.Kuttyamma A J
Professor &
Head of Department
Dept. of Information Technology
RSET

Mrs. Mary John
Project Guide
Assistant Professor
Dept.of Information Technology
RSET

Mrs. Neeba E A
Project Coordinator
Assistant Professor
Dept.of Information Technology
RSET

ACKNOWLEDGEMENTS

We would like to express our sincere thanks to **Dr. A. Unnikrishnan,** the Principal of Rajagiri School of Engineering & Technology, for providing the facilities required for doing the project. We are extremely thankful to Professor and Head of the Department, **Prof. Kuttyamma A J**, for the administrative help.

We extend our sincere thanks to the coordinator in-charge, **Mrs. Neeba E A**, Assistant Professor, Department of Information Technology, our guide **Mrs. Mary John**, Assistant Professor, Department of Information Technology for their constant support and guidance to complete the work of this project.

Above all, we thank God for his immense grace and continuous and constant showering of blessings throughout the course of our project, without which all our efforts would have been in vain.

Athira M D
Faheem Habeeb
Kirthy Francis
Yamini Anil

ABSTRACT

With the advancement in technology the number of computing devices that a person uses is increasing and there is a need of faster and non-intrusive methods of communicating with these devices. In this regard, eye movement is a promising input medium for human computer interaction. Human Computer Interaction is a trend-in technology. In this paper we discuss various eye tracking techniques that can be used to find the line of gaze of the user and perform the movement of slides in a PowerPoint presentation. The main aim of this paper is to propose new applications utilizing eye gaze that are suitable for standard user.

The input given to the system is a video which is captured by using a webcam. The still images are captured from the video that undergoes image processing. Using these still images, face and eye detection are performed using Viola-Jones Algorithm. The Circular Hough Transform Algorithm is used for pupil localization. Once the pupil is localized, the pupil centre is estimated and distance values from edge points to centre point are computed. Gaze of the user can be estimated by comparing these distance values. Once the gaze is estimated, the gaze direction is found using Template Matching. In Template Matching, each input frame is compared with templates. If the input frame matches with the template we display the corresponding template and movement of slides occurs in the desired direction. For this we import the Java class Robot which will automate the key press and key release events associated with each movement. The software implementation part can be done in MATLAB.

Table of Contents

List of Figures

LIST OF TABLES

List of Abbreviations

HCI	Human Computer Interaction
CHT	Circular Hough Transform
PCR	Pupil Corneal Reflection

CHAPTER 1

Introduction

Nowadays computers have become a part of everyone's life, as the convenience with the use of computers is increasing day by day. However use of computer is restricted to only those people who can easily handle computer peripherals like keyboard, mouse etc. but due to limb disability the handicap people are away from it. Therefore, it is necessary to develop a system that provides an alternative way for handicaps to interact with computer. The gaze of the user's eye is one of the best options for human computer interaction (HCI). Various non-invasive eye tracking techniques are developed for HCI, either only with head movement or eye movement or the combination of both. With the development of digital image processing, the eye gaze tracking techniques pursued a wide bandwidth for HCI.

Various techniques have been proposed to estimate the eye gaze. The eyeball is assumed as a spherical model to estimate the iris radius. The iris center locations of incoming video frames are detected by matching them with a database. This system is implemented by the concatenation of many algorithms that includes Viola-Jones algorithm for face and eye detection, Circular Hough

Transform (CHT) to make an accurate segmentation of pupil/iris from the sclera, Pupil Corneal Reflection (PCR) method to perform pupil detection and pupil center detection and template matching method for estimating gaze direction of user. Once gaze direction is estimated, Movement of the slides along with the movement of our eye gaze. These techniques can be implemented in modern day Power Point presentations which would make movement of slides much simpler.

1.1 PROBLEM DEFINITION

With the advancement in technology the number of computing devices that a person uses is increasing and there is a need of faster and non-intrusive methods of communicating with these devices.. Interaction with computer is not limited to keyboards and printers anymore. Eye gaze is also a recent input technique which has the potential to be widely used in future. Using eye movement for controlling the computer improves the experience of working with the computer as it is faster and gives the illusion that the computer is complying with the users' thought. It can be used either exclusively or in combination with other input technologies such as eye movement can be used along with a button so that it confirms the users' intentions for performing critical tasks and reduce the chances of error. Eye movement is a promising input medium for human computer interaction. As a direct control medium, the eye movements are obtained and used in real time as an input to the user–computer dialogue. They might be the sole input, typically for disabled users or hands-busy applications.

1.2 PROJECT OBJECTIVE

The main objective of this system is to propose a faster and non-intrusive method that enriches reading experience of the user and provides effortless movement of slides. The system tracks the movement of the eye and uses it as a medium for human computer interaction. Several tracking algorithms can be implemented to track the pupil of the eye or the iris centre. Template matching is carried out successively. Lastly, movement of slides is performed with the help of an algorithm. With the help of these algorithms, improved eye gaze estimation can be easily achieved.

1.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

1.3.1 Issues in face detection

- Irrelevant poses
- Orientation
- Beards, moustache , glasses can occlude the face features
- Poor lighting conditions
- Huge variations in the face sizes or multiple faces

1.3.2 Issues in pupil detection

- Since the parameter space of the CHT is three dimensional, it may require lots ofstorage and computation.
- Choosing an appropriate grid size is difficult.

- The CHT is not very robust to noise.
- Blinking of eyes.

1.4 Development Method

The proposed system uses horizontal gaze gestures of the eye in order to achieve the eye controlled movement of the slides. The system tracks the movement of the eye and use it as a medium for human computer interaction.The webcam captures the video of the user. Then the image processing software in the computer will capture the still images from the video. The face detection and eye detection are carried out using Viola-Jones Algorithm. To estimate the gaze of user, Pupil Corneal Reflection method is used, where pupil detection is carried out using Circular Hough Transform. The gaze of the user can be estimated with respect to pupil centre and Template Matching is performed to estimate the gaze direction. Once gaze direction is estimated , the movement of slides takes place in the desired direction based on the algorithm.

1.5 Assumptions and Dependencies

Several assumptions are made before running the project. The project will give accurate results only when the necessary constraints satisfy. The lighting conditions, persons with spectacles, multiple users etc are important among them. All this three constraints are dicussed in detail below.

Assumption 1: Brightly lit room

The project is run in a brightly lit room such that the user’s face and facial features are well illuminated and distinguishable. This can ensure proper detection and further processing of the face and eyes of the user.

Assumption 2: User does not use spectacles.

Spectacles on the user can occlude the facial features and hinder the detection of user’s eyes. This can affect the further processing needed in the project. Hence the project may not function properly.

Assumption 3: Multiple Users

There should not be more than one user at the same distance from the webcam such that both faces are detected for processing, as this can lead to discrepancies during processing and may not provide the expected output.

CHAPTER 2

Literature Review

Uma Sambrekar[1], proposes an efficient solution for limb disabled people to interact with computer using their eyes. Whenever the user looks at the key to be pressed, the position of pupil center varies. At that time, the eye images are captured by video camera and transmitted to the personal computer through USB cable. Viola-Jones Algorithm is used for face detection and eye detection. The Pupil Corneal Reflection method is used for gaze estimation which includes pupil detection and glint detection. The Hough Transform Algorithm is used for pupil localization. In order to detect the glint (a small and intense dot inside the pupil image) the Blob Analysis method is used. The template matching method is implemented to detect blink, that proceeds to a mouse click in the window. Once the glint and pupil center positions are determined, gaze of the user can be estimated by calibrating current glint location with respect to pupil center.The software implementation part is done in MATLAB. The system gives appreciable results and speed with an accuracy of 90%.Here less hardware complications involved and hence it is cost efficient. Currently the system is implemented for 4-keys but in future ,with some improvement in algorithm the number of keys on the keypad can be incremented or a separate virtual keyboard can be created.

Arpita Ray Sarkar[2], proposes an efficient solution for developing a rehabilitation system for stroke patients that will be used for assistance in daily living of the patient through the use of eye gazes. The first step in this system is to perform eye gaze estimation. This is done using video image based analysis. Using the Haar-like cascade classifier, face detection and subsequent tracking of eye are performed. With the help of AdaBoost algorithm,eye regions are detected at each stage of cascade classifier. An improved version of Hough transform in two dimensional parametric space has been used for effective eye centre localization. With this hybrid approach to perform eye centre localization for eye gaze estimation has become easier using low-cost webcams in different lighting conditions with and without spectacles. Implementation is done using Microsoft Visual Studio with OpenCV libraries. The present approach has been tested in the normal rooms of houses and hospitals where rehabilitation systems are supported using two low cost (low and high resolution)webcams. The current system has the advantage of lower computational load and faster processing of images.It has got simple and easy to implement features. However, there are several issues, such as low illumination, image quality, direct light on the lens of the camera, cleanliness and glaring of glasses etc. behind the hindrances.

Prof. Sasinas [3] proposes an efficient method for tracking eye and detecting blink to produce an environment by which physically disabled persons can access the computer very easily. Eye is the main sensing organ in human body, so it contributes much for user interface. Eye tracking has been widely used in Human Computer Interactions. Real time face image capturing along with eye gaze tracking can provide a means of user input to the computer. The aim is to minimize the hardware and setup cost by using accurate image processing techniques. The system will track the eye of a person by camera and process them to control the mouse pointer on the screen. The user can control the mouse by moving and blinking the eye. The system use isophote eye centre detection method to find the iris. It gives an accuracy of 99.20%. The curves that connect the points of equal intensities are called isophote. Binary equivalent image is used to perform blink detection, in which number of black pixels is counted to determine the blinking of eye.The coordinate position of eye detected using eye detection algorithm, is compared with the value of eye detection result of a template. The difference of eye centre is calculated and mapped to the screen. A scalar is multiplied with this value to map the mouse motion in to the screen. If the eye is closed then mouse click is carried out. Face detection is also done inorder to eliminate unwanted regions.This scheme has succeeded in detecting eyes with high accuracy. So this method which combines isophote eye centre detection and blink detection gives a precise result The future works also include expanding this work to kernel space of the system and to implement eye tracking combined with speech recognition to overcome text based processing.

Muhammad Awais[4], proposes an automated eye blink detection and tracking system using template matching. Eye blink detection is one of the most reliable sources of communication in modern human computer interaction (HCI) systems Inorder to minimize the false detection due to changing background in the video frame, face detection is applied before extraction of the eye template.

Golden ratio concept is introduced for Eye Pair Detection and Template Creation. The two quantities are in the golden ratio if their ratiois the same as the ratio of their sumto the larger of the two quantitiesEye tracking is performed by template matching. The normalized correlation coefficient is computed for eye tracking. Eye blink detection is performed based upon the correlation coefficient as the score changes significantly whenever a blink occurs. There occurs adecrease in correlation score in case of open eye template and increase in correlation score in case of close eye template.The proposed system provides an overall precision of 92.8% and overall accuracy of 99.6% with 0.1% false positive rate in different experimental conditions.The system enables people with disabilities, who can only blink their eyes, to perform different day to day activities. The face detection and golden ratio computation makes eye detection process very robust.The proposed system will focus more on eye blink duration analysis as duration analysis will be used for driver drowsiness detection.

JianBingXiahou[5], proposes an integrated human eye movement recognition and tracking system. The threshold based eye recognition method identifies eye elements, such as iris and pupils of human eyes. Otsu method and Optimal method are applied in the process of image segmentation. Otsu's method is used to automatically perform clusteringbased image thresholding or the reduction of a graylevel image to a binary image. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread is minimal so that their inter-class variance is maximal. Haar-like method is applied to the pre-processed image to remove the redundancy regions which disturbs the results of the recognition. In addition, the eye movement tracking method is present by analyses motion feature of eyes including translation and velocity. The classification of eye movement is done using Naive Bayes classifier. Naive Bayes classifiers are highly scalable, requiring many parameters linear in the number of variables in a learning problem. Naive Bayes is a collection of classification algorithms based on Bayes Theorem. The method explained in this paper can be applied in different devices, such as Microsoft Kinect or Intel RealSense to extend performance in eye gesture recognition applications. The system is less intrusive, reliable and cost efficient.

Daniel Gêgo [6] proposes a telerobotic platform that uses a user interface based on eye-gaze tracking. The system is evaluated using a task-oriented evaluation and the results permit to conclude that the proposed interface is a feasible option as a mean of HRI in teleoperation applications.This paper presents a telerobotic platform that uses a user interface based on eye-gaze tracking that enables a user to control the navigation of a teleoperated mobile robot using only his/her eyes as inputs to the system. Details of the operation of the eye-gaze tracking system and the results of a task-oriented evaluation of the developed system are also included. The system has more flexibility during monitoring and control tasks and can also provide the ability to implement telepresence.

LilingYu[7], proposes a new type low pixel eye feature point location method. This method can accurately extract the eye-gaze features, namely iris centre point and canthus points when the image pickup requirements are low. The eye-gaze tracking method based on particle swarm optimization (PSO) BP neural network is raised, to capture pictures of eyes under the same environment, and a regression model where the connection weights and threshold values are optimized by PSO algorithm is built via BP network. This method requires only a common camera and normal illumination intensity rather than high-standard hardware, which greatly cuts the restrictive requirements for the system hardware. The experiment results show that PSO-BP model is of higher robustness and accuracy than BP model, and is of higher recognition rate and can effectively enhances the eye-gaze tracking accuracy.This method is free of the inherent defects of BP network and thus enhances the system practicability. The system focuses on the eye-gaze tracking method only under the condition of free head movement. This improved method has fast convergence.

CHAPTER 3

Proposal for Dissertation

The systems aims at developing an efficient solution for hand disabled people to interact with computer using their eyes. Here we present a novel approach for Human Computer Interaction (HCI) where, we move the slides in a PowerPoint presentation by using eye gazes. Our method is to use a webcam to capture the video of the user from which the still images are extracted and processed for face, eye and pupil detection. We used **MATLAB** to call web camera which is set to take images continuously from the eye focusing pupil. With the help of different image processing techniques, the eye recognition and tracking is achieved. The processing techniques involve Viola Jones Algorithm, Circular Hough Transform and Template Matching. The slide movement can be achieved with help of a java class called Robot. We need a key press and key release for every movement. By assigning the key to Robot we can automate this key press and key release.When the user looks in a particular direction the corresponding key press and key release will occur and the slides move in the desired direction. If the user is looking straight then no movement takes place.

3.1 IMPLICATIONS FOR RESEARCH

Many modules have been developed to help the physical world interact with the digital world. This method mainly focuses on the use of a web camera to develop a virtual human computer interaction in a cost effective manner. It also presents hands free interface between computer and human especially for physically disabled persons. It will enrich the reading experience of the user. The project can be further modified to include detection of eyes under spectacles as many users may use spectacles for better vision, to increase the detection of eyes under dim light as all the users may not be sitting under good lighting conditions. Eliminating occlusions such as secondary reflection, glare, etc during image processing can lead to faster and more accurate detection of eye movement. Using the blink of a user to open a file can give greater functionality to the project. Moving the cursor according to the movement of eyes, this can be further modified for various functions according to the needs of the user.

CHAPTER 4

System Overview and Requirements Specifications

4.1 SYSTEM OVERVIEW

The input given to the system is a video which is captured by using a webcam. The still images are captured from the video that undergoes image processing. Using these still images, face and eye detection are performed using Viola-Jones Algorithm. The Circular Hough Transform Algorithm is used for pupil localization. Once the pupil is localized, the pupil centre is estimated and distance values from edge points to centre point are computed. Gaze of the user can be estimated by comparing these distance values. Once the gaze is estimated, the gaze direction is found using Template Matching. In Template Matching we compare each input frame with left, right, straight and no face templates in our MATLAB folder. If the input frame matches with the template we display the corresponding template and movement of slides occurs in the desired direction. For this we import the Java class Robot which will automate the key press and key release events associated with each movement. The software implementation part can be done in MATLAB.

4.2 FUNCTIONAL REQUIREMENTS

4.2.1. Requirements Definition Table

The requirements attribute table is used to track the status of all types of requirements throughout the life of a project. The types of requirements tracked may be features, user requirements (system use cases), or functional requirements. As each type of requirement is documented, it is helpful to create unique tracking numbers for each type so that the status of the requirement can be obtained easily. The same tracking number may be used across all documentation in order to trace the development of requirements over time.

S.No.	Requirements	Requirement Type	Priority
1.	System must capture video images.	STRQ1	HIGH
2.	System should contain a webcam.	STRQ2	HIGH
3.	System must conduct image processing.	STRQ3	HIGH
4.	Face detection must be performed using Viola Jones algorithm.	STRQ4	HIGH
5.	Eye detection must be conducted using Viola Jones algorithm.	STRQ5	HIGH
6.	Pupil localization must be performed.	STRQ6	HIGH
7.	Pupil center is found out.	STRQ7	HIGH

8.	Gaze estimation must be performed.	STRQ8	HIGH
9.	Template matching should be performed.	STRQ9	HIGH
10.	Movement of slides take place in the desired direction.	STRQ10	HIGH
11.	Enriched reading can be performed.	FEAT1	HIGH

Table 4.1.Requirements Definition Table

4.2.2.Use Case Diagrams

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. They provide the simplified and graphical representation of what the system must actually do.

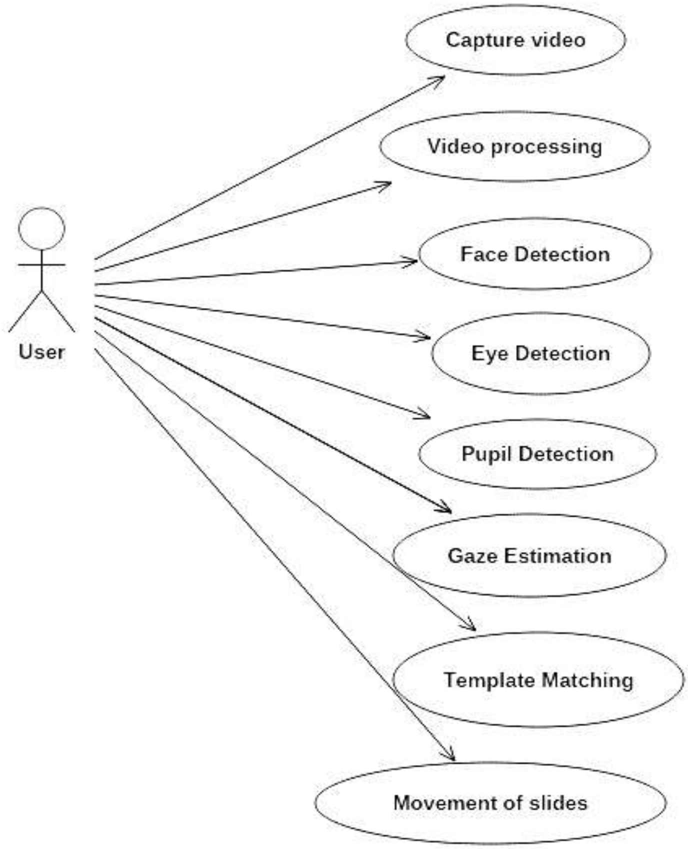


Fig 4.1. Use Case Diagram

4.2.3. Activity Diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control. Arrows run from the start towards the end and represent the order in which activities happen. The activity diagram shows the flow of activities in each functions in the project and the use cases shows the features or user requirements of each functions.

UC1: Capture Video

Flow of Events:

- The input given to the system is a video of the user which is captured by using a webcam.

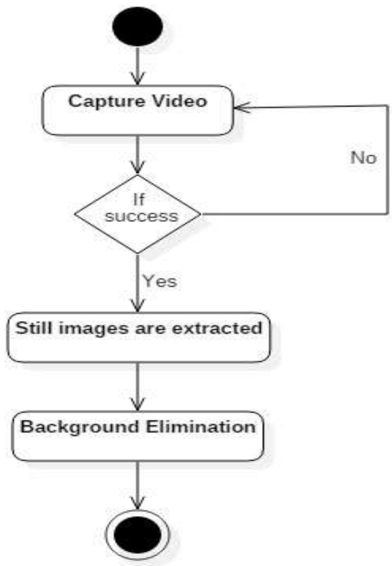


Fig 4.2. Capture Video Activity Diagram

UC2: Video Processing

Flow of Events:

- Still images are processed from the captured video by background elimination using image processing software. The image processing software will then access each frame and performs some processing operations to estimate the gaze of the user.

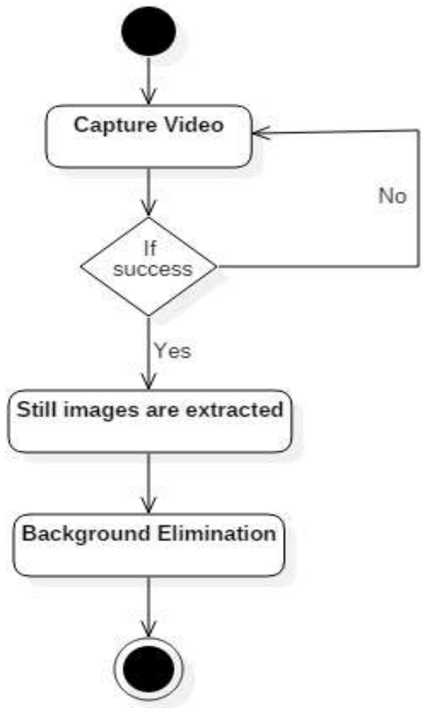


Fig 4.3. Video Processing Activity Diagram

UC3: Face Detection

Flow of Events:

- The face of the user is detected using Viola-Jones algorithm.
- The input to the Viola-Jones Algorithm is a grayscale image.
- The selected features are absent or present in incoming input frame is checked in each strong classifier. If the first feature is present in frame, it is passed to the next classifier and if it is absent, the frame is discarded at the same stage as non face part. The frame with minimum 10 features is considered as face.

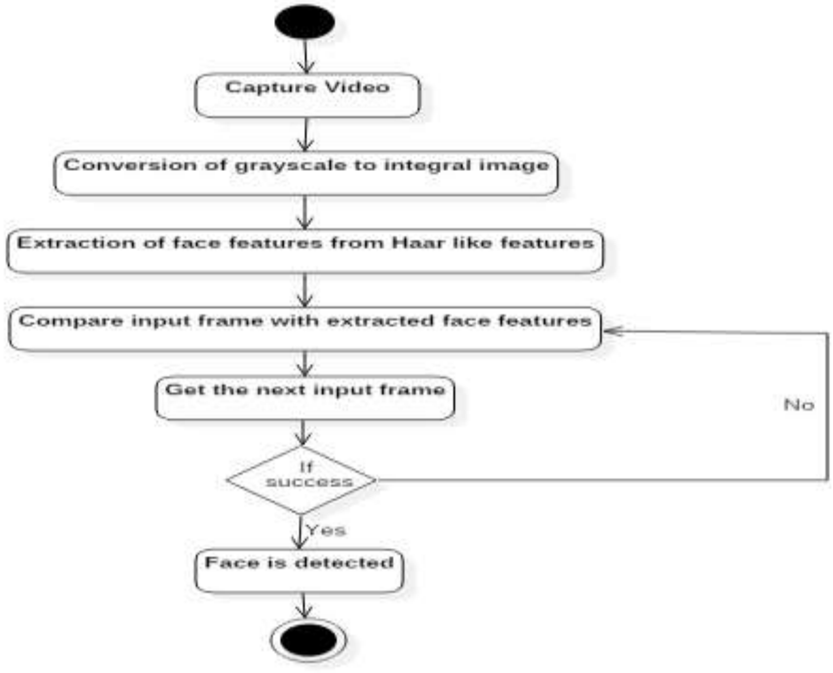


Fig 4.4. Face Detection Activity Diagram

UC4: Eye Detection

Flow of Events:

- The eyes of the user are detected using Viola-Jones algorithm.
- The eyes are detected from the cropped face region.
- The selected features are absent or present in incoming input frame is checked in each strong classifier. If the first feature is present in frame, it is passed to the next classifier and if it is absent, the frame is discarded at the same stage as non eyes part. The frame with minimum 10 features is considered as eyes.

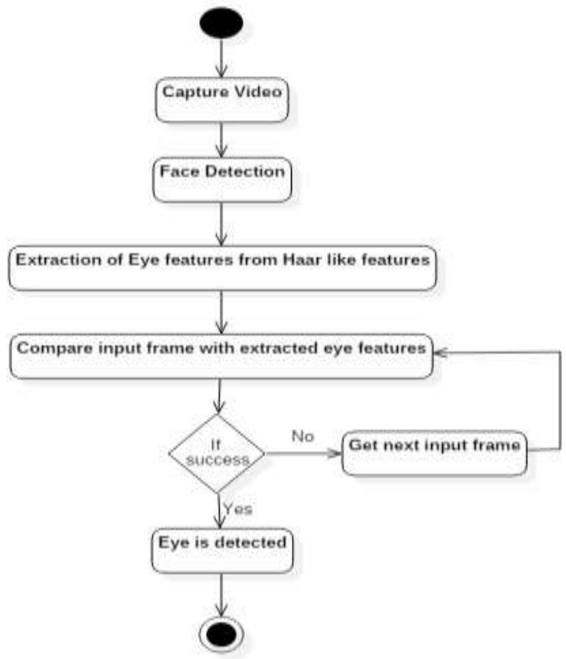


Fig 4.5. Eye Detection Activity Diagram

UC5: Pupil Detection

Flow of Events:

- The gaze of the user is estimated using Pupil Corneal Reflection method.
- Hough Transform (HT) is used to extract the pupil from the sclera.
- The input to the Hough Transform is a smoothed image and canny edge detected image.
- The center of pupil is found out to estimate the gaze of the user.

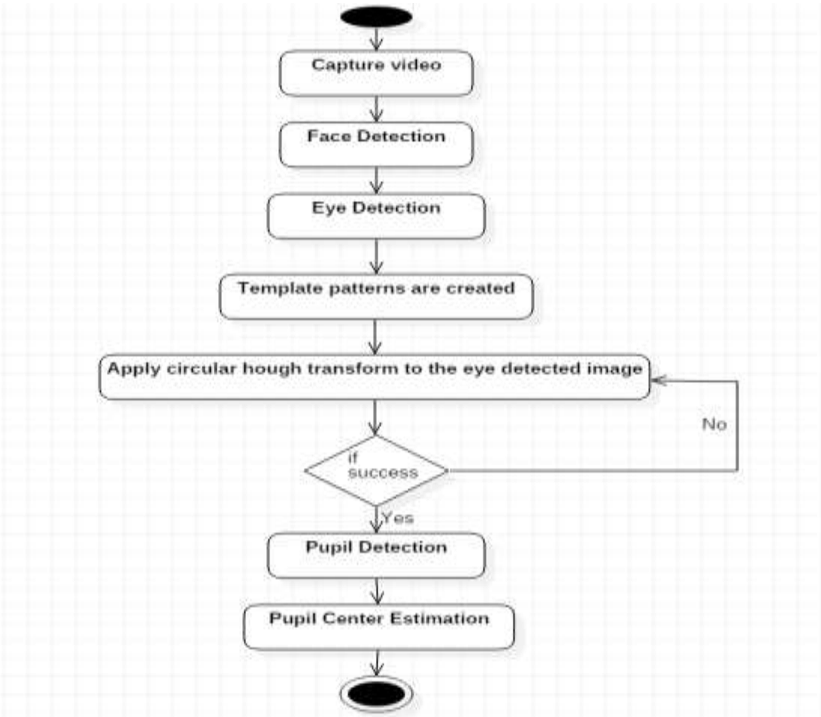


Fig 4.6. Pupil Detection Activity Diagram

UC6: Gaze Estimation

Flow of Events:

- Once the pupil is localized and center is found, we will compute the distance from right and left edge points to the center point.
- Based on this distance value, we estimate the gaze direction.
- For estimating the gaze direction we perform Template Matching where the input frame is compared with the Right, Left, Straight and No face templates and display the matching templates.

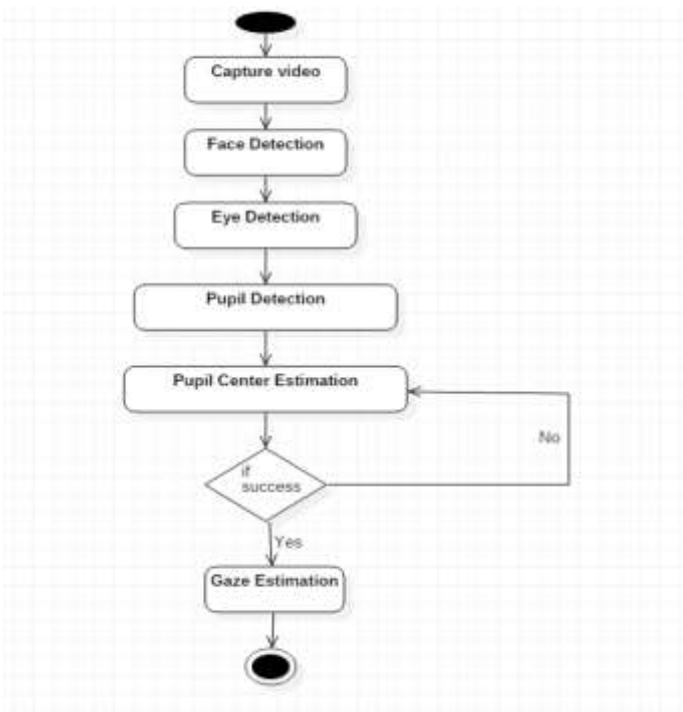


Fig 4.7. Gaze Estimation Activity Diagram

UC7: Template Matching

Flow of Events:

- Template Matching is performed to estimate the gaze direction of the user.
- The input frame is compared with the Right, Left, Straight and No face templates and display the matching templates.

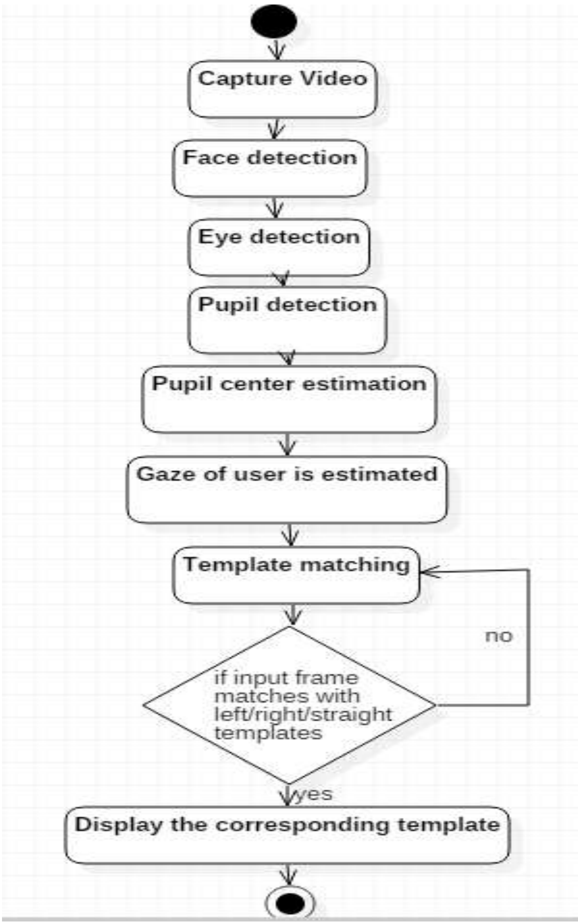


Fig 4.8. Template Matching Activity Diagram

UC8: Movement of Slides

Flow of Events:

- In order to perform the movement of slides we import the Java class Robot.
- For moving slides, there needs a key press and key release.
- By assigning key to Robot class. We will automate this key press and key release functions and the movement of slides takes place in the desired direction.

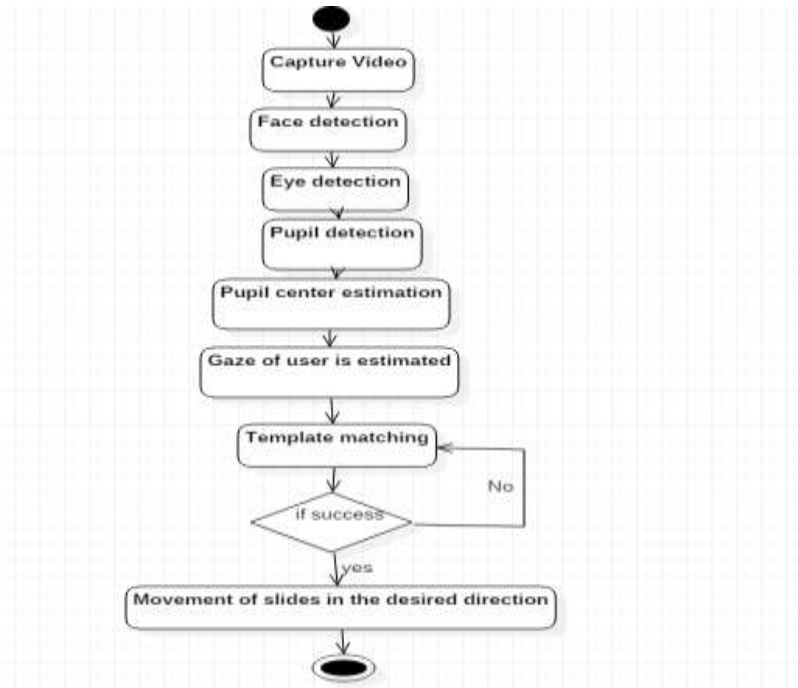


Fig 4.9. Slide Movement Activity Diagram

4.2.3. Sequence Diagrams

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines , different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

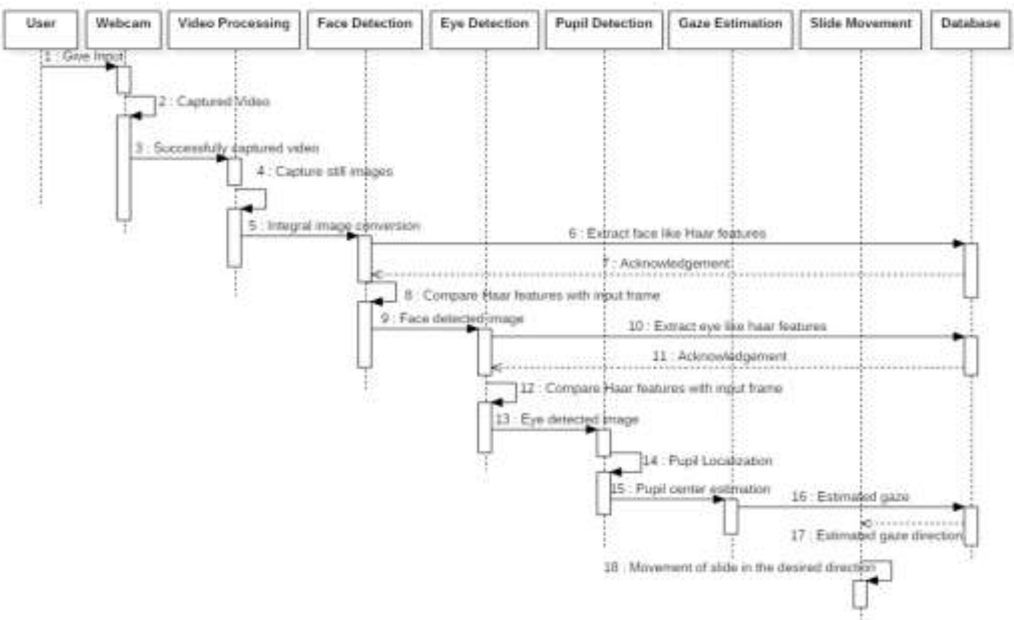


Fig 4.10. Sequence Diagram

4.3 GUI DESIGN

4.3.1 User Interfaces

The UI consists of the video input.

4.3.2 Hardware Interfaces

The application takes input from a camera and processes the video input.

4.3.3 Software Interfaces

We used **MATLAB** to call web camera. The application uses HAAR cascade algorithms and processes the information. By comparing the distance values between edge points and centre points of pupil the gaze is estimated. The java Robot class is used for performing slide movement.

4.4 NON FUNCTIONAL REQUIREMENTS

4.4.1.Performance Requirements

- Adequate lighting conditions.
- The user must not wear spectacles while recording the video.
- A webcam of high resolution is required.

4.4.2. Software Quality Attributes

- MATLAB is required to process video. The system requires 2 GB disk space for MATLAB.
- The system must have a supported Windows and has a video player like VLC media player.
- System must have a 2-4GB RAM.

4.5 HARDWARE AND SOFTWARE REQUIREMENTS

4.5.1 Hardware Requirements

- PC or laptop with minimum 512mb RAM and 2GB HDD.
- A webcam to produce a video with camera of minimum 720 pixels .
- A processor better or similar to Intel Core i5-4200u).

4.5.2 Software Requirements

- MATLAB: to process video.
- System with a supported Windows.
- A video player like VLC media player.
- System requires 2 GB disk space for MATLAB and 2-4GB RAM.

CHAPTER 5

System Architecture

5.1 FLOW DIAGRAM

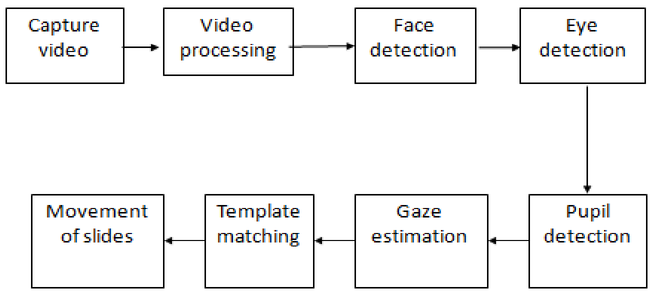


Fig.5.1 Flow Diagram

5.2 REAL TIME VIDEO INPUT/VIDEO PROCESSING

The input given to the system is video, which is acquired by the Logitech HD 720p webcam. The camera is connected to the computer and the camera is mounted on the top of the computer screen. Still images are processed from the captured video by background elimination using image processing software. The image processing software will then access each frame and performs some processing operations to estimate the gaze of the user.

5.3 FACE DETECTION/EYE DETECTION

The first step is face detection and then eye detection. To detect the face and eye, the algorithm used is Viola-Jones algorithm. In Viola-Jones algorithm, Haar like features are used to detect the face. The three key points involved in this algorithm are: first is forming an integral image for fast feature evaluation. Second, we create a detector for face and eye pair using Viola-Jones. The cascade object detector uses the Viola Jones algorithm to detect people's faces, noses, eyes, mouth or upper body, using the detectors we select the most suitable Haar like features to detect faces. And third is forming an integral image, then discard the background region quickly and consider only the face like regions. The input to the Viola-Jones Algorithm is a grayscale image. Therefore the image is converted into an integral image (black and white image). An integral image at any point can be formed as the summation of pixels above and to the left of that point. There are approximately 1,60,000 Haar-like features available but only a few sets of features are useful for detecting the face. The same algorithm can be used to detect the eyes but the features selected for this are different.

Fig 5.2 Captured Eye image

Fig 5.3 Gray image

Whether the selected features are absent or present in incoming input frame is checked in each strong classifier. If the first feature is present in frame, it is passed to the next classifier and if it is absent, the frame is discarded at the same stage as non face part. The frame with minimum 10 features is considered as face.

The Algorithm is:

Input: Grey scale image

Step 1: Start

Step 2: Convert grey scale to integral image.

Step 3 : Repeat

Step 3.1 : Check input frame with first selected Haar-like feature.

Step 3.2 : If feature is present in the input frame, then

Step 3.2.1 : Check input frame with next Haar-like feature.

Step 3.3 : Else

Step 3.3.1 : Discard the frame.

Step 3.4 : Until minimum 10 features are considered.

Step 4 : Stop

Fig.5.4 Face Detection and Eye Detection

5.4 PUPIL LOCALIZATION FOR GAZE ESTIMATION

To estimate the gaze of user Pupil Corneal Reflection method is used. The method consists of pupil detection. When the fraction of light enters into the eye, it gets reflected from the retinal surface and forms pupil image. To determine the gaze of user, it is necessary to segment the pupil. One of the segmentation algorithm known as Hough Transform (HT) is used to extract the pupil from the sclera.

The Algorithm used is as follows:

Input: Eye Detected Image

Step 1: Start

Step 2: Adjust the contrast of the eye detected image.

Step 3 : Compute the centre of the given eye image.

Step 4 : Compute imfindcircles( ) :

Step 4.1 : If the input frame contains circles, then

Step 4.1.1 : Returns centers, radii and metric of the given frame

Step 4.2 : Else

Step 4.2.1 : Discard the frame and accepts next frame.

Step 5 : Stop

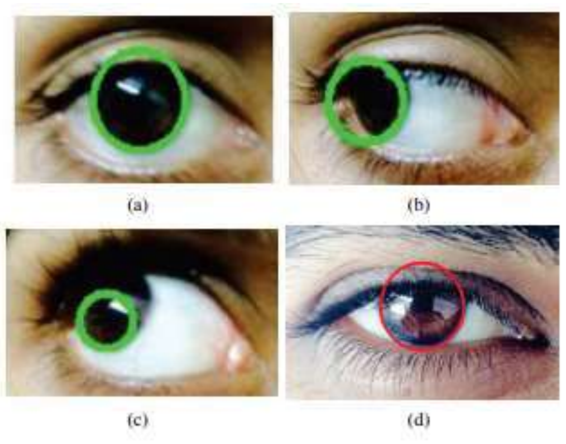


Fig. 5.5 Pupil Detection using Hough Transform

5.4.1 Hough Transform

The Hough Transform (HT) is a technique that locates shapes in images. It is used to extract lines, circles and ellipses (or conic sections). The aim is to find the center coordinate of pupil and its radius to detect the circular shape of pupil for gaze estimation.

The HT can be defined by considering the equation for a circle given by:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

This equation defines locus of points (x,y) centered on an origin (x0, y0) and with radius r. This equation can again be visualised in two dual ways: as a locus of points (x,y)in an image or as a locus of points (x0, y0) centered at (x,y) with radius r. Each edge point defines a set of circles in the accumulator space. All these circles are defined by all possible values of the radius and are centered on the co-ordinates of the edge point.

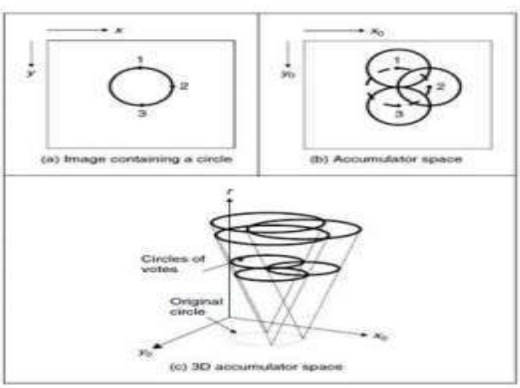


Fig. 5.6 Hough Transform Mapping

These circles are defined for a given radius value. Actually, each edge point defines circles for the other values of the radius. This implies that the accumulator space is three dimensional (for the three parameters of interest) and that edge points map to a cone of votes in the accumulator space. The procedure of evidence gathering is the same as that for the HT for lines, but votes are generated in cones, Equation of a circle can be defined in parametric form as

$$x = a + r * cos\theta$$

$$y = a + r * sin\theta$$

In Hough transform mapping, usually an empty 3-D accumulator array is formed to save the weights of edge points. If the edge point satisfies a specific dimension circle equation then weight of the centre point in the Hough array is incremented. Number of edge points on a circle with specific dimensions (centre (x,y) and radius "r") are counted by the value of the corresponding point in accumulator array and an output is saved in a matrix containing [x co-ordinate, y coordinate, radius, count (no. of edge pixels on that circle)].Now the circle with maximum count is considered, as the prominent circle. This circle position and radius is assigned to the pupil center and radius.Since the parameter space of the CHT is three dimensional, it may require lots of storage and computation. While numerous feature extraction techniques are openly available for circle detection, one of the most robust and commonly used methods is the Circular Hough Transform (CHT).

5.5 GAZE ESTIMATION

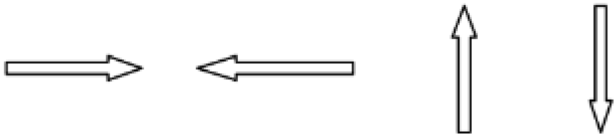


Fig.5.7 Relevant Gaze Gestures

Once the pupil is detected, the pupil centre is estimated. The MATLAB function imfindcircles() detects the circles within the eye detected image, which is pupil. The imfindcircles() return the centre and radius of the pupil. The distance from left and right edge to centre point is computed. We compare this distance values in order to estimate the actual gaze of user can be estimated. Once the gaze of user is estimated we perform the template matching.

5.6 TEMPLATE MATCHING

The template matching is done in order to estimate the gaze direction of the user. In the MATLAB folder , four templates are created for left gaze, right gaze, straight gaze and no face. We check each input frame with the templates. If the frame matches with any of these templates, then the user is looking in that particular direction and we display the corresponding template. Once the matching template is obtained, we perform the movement of slides in the desired direction.

5.7 EYE CONTROLLED MOVEMENT OF SLIDES

In order to perform the movement of slides we import the Java class Robot. The Robot class in the Java AWT package is used to generate native system input events for the purposes of test automation and other applications where control of the mouse and keyboard is needed. For moving slides, there needs a key press and key release events. By assigning key to Robot class, we will automate this key press and key release functions. If the gaze direction is left or right, then the slides move in that desired direction. Otherwise if the gaze direction is straight, no movement takes plac . The Algorithm used is:

Input : Gaze estimated image.

Step 1 : Start

Step 2 : Load the power point document to be opened into the MATLAB.

Step 3 : Import the java class Robot.

Step 4 : Assign key to Robot

Step 5 : Get the input frame and check for its gaze direction.

Step 6 : If direction is towards left, then

Step 6.1 : Movement from current slide to previous slide.

Step 7 : If direction is towards right, then

Step 7.1 : Movement from current slide to next slide.

Step 8 : Otherwise

Step 8.1: No movement.

Step 9 : Stop

CHAPTER 6

System Analysis

6.1. USER MODULE

MATLAB is a multi-paradigm numerical computing environment. MATLAB is not only a programming language, but a programming environment as well. We can perform operations from the command line, as a sophisticated calculator. Or we can create programs and functions that perform repetitive tasks, just as any other computer language. In our system, we use MATLAB to develop a virtual human computer interaction especially for physically disabled persons. In MATLAB we can import classes into a function to simplify access to class members. Here we import the java class Robot. Robot Class can simulate keyboard and mouse event. The keyPress() method in Robot class will press down arrow key of keyboard. For example, robot.keyPress(KeyEvent.VK_DOWN) .The keyRelease() method in Robot class will release down arrow key of Keyboard. For example, robot.keyRelease(KeyEvent.VK_DOWN) . The primary purpose of Robot is to facilitate automated testing of Java platform implementations.The Robot class in the Java AWT package is used to generate native system input events for the applications where control of the mouse and keyboard is needed.

The user initially opens the MATLAB software. Then he/she can selectsthe required Power Point presentation . The user then runs the program in MATLAB. A video of the user is taken through webcam. The VideoDevice System object allows single-frame image acquisition and code generation from MATLAB. We can use the following command for this, obj = imaq.VideoDevice(adaptorname), which creates a VideoDevice System object obj, using the first device of the specified adaptorname, where adaptorname is a character vector that specifies the name of the adaptor used to communicate with the device. The function frame = step(obj) acquire a single frame from the the VideoDevice System object, obj. The slides move in the desired direction of the user with respect to his gaze direction. If the gaze direction of the user is to the left direction, the previous slide is displayed and if the gaze direction of the user is to the right direction, the next slide is displayed. If the gaze direction of the user is straight, no movement of slides takes place and the current slide is displayed. The estimated gaze of the user is plotted on a window along with face detected and eye detected images. The function subplot( ) will plot all this images in a single window. We can also use *subimage( )* in conjunction with subplot to create figures with multiple images, even if the images have different colormaps. The function pause(n) pauses execution for n seconds before continuing.

CHAPTER 7

Implementation Details

7.1 REAL TIME VIDEO INPUT/VIDEO PROCESSING

The input given to the system is video. First of all we will disconnect and delete all image acquisition objects using the function “imaqreset”. It is the image acquisition command that returns MATLAB to the known state of viewing no image acquisition objects and no loaded image acquisition adaptors. The VideoDevice system object allows single frame image acquisition and code generation from MATLAB. The function obj=imaq.VideoDevice(adaptorname, deviceid, format) creates a VideoDevice system object obj using the specified adaptor name. Here format is a text string that specifies a particular video format supported by the device or a device configuration file. The function acquires one frame at a time from VideoDevice. We can use the “imaqhwinFo” function to determine the adaptors available on the system. The “ReturnedColorSpace” property specifies the color space we want the toolbox to use when it returns image data to the MATLAB workspace.

7.2 FACE DETECTION/EYE DETECTION

Face recognition is the process of identifying one or more people in images or videos by analyzing and comparing patterns. Algorithms for face recognition typically extract facial features and compare them to a database to find the best match.Here we create a detector for face and eye using Viola-Jones.

detector=vision.CascadeObjectDetector

creates a system object, detector that detects objects using the Viola-Jones algorithm. The function “imread()” reads a grayscale or color image from the file specified by the String filename. The function “ rgb2gray()” convert RGB image or color map to grayscale. The function:

BBOX= step(DETECTOR, I)

returns BBOX a Mx4 matrix defining M bounding boxes containing the detected objects. Each row of the output matrix BBOX contains a four element vector[x, y, width, height] that specifies in pixels, the upper left corner and size of a bounding box. The new image, I must be a grayscale or RGB image. The function “flip()” will vertically or horizontally reverses the MxN input matrix. We can use the function “rank(bbox)” to find the biggest face/eye. The function “imcrop()” will extract the required portion from the specified image. The function “subplot()” will plot all the images in a single window. The function “subimage()” can be used in conjunction with subplot to create figures with multiple images, even if the image has different colormaps. For face detection and eye detection we use the same algorithm but the detector we create will be different for face and eye.

7.3 PUPIL LOCALIZATION FOR GAZE ESTIMATION

The function “imfindcircles()” is used to detect the pupil using Circular Hough Transform.

[centers, radii, metric]=imfindcircles(A, radiusRange)

This function finds circles with radii in the range specified by radiusRange. The metric is a column vector. The function “floor()” rounds the elements of A to the nearest integers less than or equal to A. The function “viscircles()” can be used to plot the detected pupil.

7.4 GAZE ESTIMATION

For gaze estimation, we compute the distance from left edge and right edge of the pupil to the center point. Then, we will compute the difference between these values in order to estimate the gaze of the user. In our MATLAB folder, we create four templates for left, right, straight and no_face. By comparing these distance values, we will display the matching templates using the function “subimage()”. Once the matching template is obtained, we perform the movement of slides in the desired direction.

7.5 EYE CONTROLLED MOVEMENT OF SLIDES

The Robot class in the Java AWT package is used to generate native system input events for the purposes of test automation and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations. In simple terms, the class provides control over the mouse and keyboard devices. For moving slides, there needs to be keyPress and keyRelease events. By assigning key to Robot class, we will automate this keyPress and keyRelease functions and the movement of slides takes place according to the estimated gaze. If the gaze direction is to the left, then the slide moves to the previous slide. If the gaze direction is to the right, then it moves to the next slide. Otherwise if the gaze direction is straight, no movement takes place and the current slide is shown. The important benefits of Robot class is that it will simulate the Keyboard and Mouse events. Robot Class can easily be integrated with current automation framework. Robot framwork has few disadvantages too. Keyword/mouse event will only works on current instance of Window.Most of the methods in Robot class is screen resolution dependent so there might be a chance that code working on one machine might not work on other.

CHAPTER 8

Testing Strategies

8.1 TESTING METHODOLOGIES

8.1.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.

We performed unit testing on each of our modules

1. Face Detection: Detection of face from still images.
2. Eye detection: Detection of eye from the face region .
3. Pupil Detection: Detection of pupil from the eye region.
4. Gaze Estimation: Pupil centre is estimated. The gaze of user is estimated by comparing distance values from edge points to the pupil centre.
5. Template Matching: Compare input frames with eye templates stored in the

MATLAB folder.

6. Eye Controlled Movement of Slides: Movement of slides occurs in the estimated gaze direction with the help of an imported java class Robot.

8.1.2 Integration Testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are created to check whether each of the components within the assemblages interact correctly. We integrated each of the modules one by one and tested the partially integrated system.

8.1.3 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. All modules were integrated at the end and the testing was performed on the entire system. The eye controlled movement of slides occurs if the necessary condition satisfies.

8.2 TEST CASES

#		TS1	
Title		To perform “Face Detection”.	
Description		Create detector for face like Haar features. Capture still images from video and compare the frame with this detector. If a match is found, then the face is detected.	
#	Summary	Conditions	Expected Output
TC1	Verify whether the input frame contains face.	If the input frame matches with face detector.	Face detected image is displayed.

#		TS2	
Title		To perform “Eye Detection”.	
Description		Create detector for eye like Haar features. Compare the Face detected image with this detector. If a match is found, then the eye is detected.	
#	Summary	Conditions	Expected Output
TC1	Verify whether the face detected image contains eye .	If the input frame matches with eye detector.	Eye detected image is displayed.

#		TS3	
Title		To perform “Pupil Detection”.	
Description		By using Circular Hough Transform algorithm call the function imfindcircles( ). If the eye detected image contains circle in it , then the function returns the centre and radius of that detected circle and it will be the pupil.	
#	Summary	Conditions	Expected Output
TC1	Verify whether the input frame contains circle in it.	If the function imfindcircles( ) return values corresponding to the centre and radius of the circle.	The detected circle is displayed in the eye image.

#		TS4	
Title		To perform "Template Matching".	
Description		Create left, right, straight and no face templates in the MATLAB folder. Check the input frame with each of these templates. If a match is found display the corresponding template.	
#	Summary	Conditions	Expected Output
TC1	Verify whether the input frame matches with "Straight" template.	If the user is looking in straight direction.	Display the "Straight" template.
TC2	Verify whether the input frame matches with "Left" template	If the user is looking in the left direction.	Display the "Left" template.
TC3	Verify whether the input frame matches with "Right" template	If the user is looking in the right direction.	Display the "Right" template.
TC4	Verify whether the input frame matches with "No Face" template	No face detected in the input frame.	Display the "No Face" template.

#	TS5
Title	To perform “Eye Controlled Movement Of Slides” by estimating user’s gaze direction.
Description	Initially user’s gaze direction is estimated. If the user is looking in the left direction, then the slide moves backward, i.e. to the previous slide. If the user is looking in the right direction, then the slide moves forward, i.e. to the next slide. If the user is looking straight then no movement takes place.

#	Summary	Conditions	Expected Output
TC1	Verify whether the user’s gaze is towards left.	Input frame matches with “Left” template	Slide moves backward
TC2	Verify whether the user’s gaze is towards right	Input frame matches with “Right” template	Slide moves forward
TC3	Verify whether the user’s gaze is straight	Input frame matches with “Straight” template	No slide movement

CHAPTER 9

Risks and Challenges

Gaze is an attractive modality for public displays, hence the recent years saw an increase in deployments of gaze enabled public displays. Although gaze has been thoroughly investigated for desktop scenarios, gaze-enabled public displays present new challenges that are unique to this setup. In contrast to desktop settings, public displays (1) cannot afford requiring eye tracker calibration, (2) expect users to interact from different positions, and (3) expect multiple users to interact simultaneously. In this work we discuss these challenges, and explore the design space of gaze-enabled public displays. We conclude by discussing how the current state of research stands wrt. the identified challenges, and highlight directions for future work.

Some of the most common challenges faced during gaze estimation process are as follows:

Risk 1: Calibration. The usability problems associated withcalibration have received considerable attention in the past years, resulting in a number of calibration-free gaze-enabled systems. Some works estimated gaze with relatively low accuracy using RGB and depth cameras, these methods relied heavily on head-tracking and face detection. Other works, such as Pursuits and the pupilcanthi-ratio, focused on developing calibration free gaze-interaction techniques rather than estimating a precise gaze point. Another direction of work in this area is to make calibration easier and blend it into public display applications. Pfeuffer et al. introduced pursuit calibration, where users calibrate by following a moving object on the screen. Khamis et el. developed Read2Calibrate, which calibrates the eye tracker as users read text on the display such as welcome messages and usage instructions.

Risk 2: User Positioning. Since commercial eye trackers impose strict user positioning requirements, researchers investigated ways to guide users to the sweet spot at which remote eye trackers would detect their eyes. In their evaluation of GazeHorizon, Zhang et al. guided passers by using an on-screen mirrored video feed as well as distance information.

Risk 3: Supporting multiple Users. Commercial IRPCR remote eye trackers track only one user at a time. Pfeuffer et al. built a collaborative information display that uses two remote eye trackers. Users were required to stand infront of the eye trackers to begin interaction. An alternative is to use video-based techniques that can track multiple users. However a drawback is that tracking quality in video-based approaches is heavily influenced by many factors such as varying light conditions and reflections of eye glasses.

CHAPTER 10

Results and Discussion

10.1 RESULTS

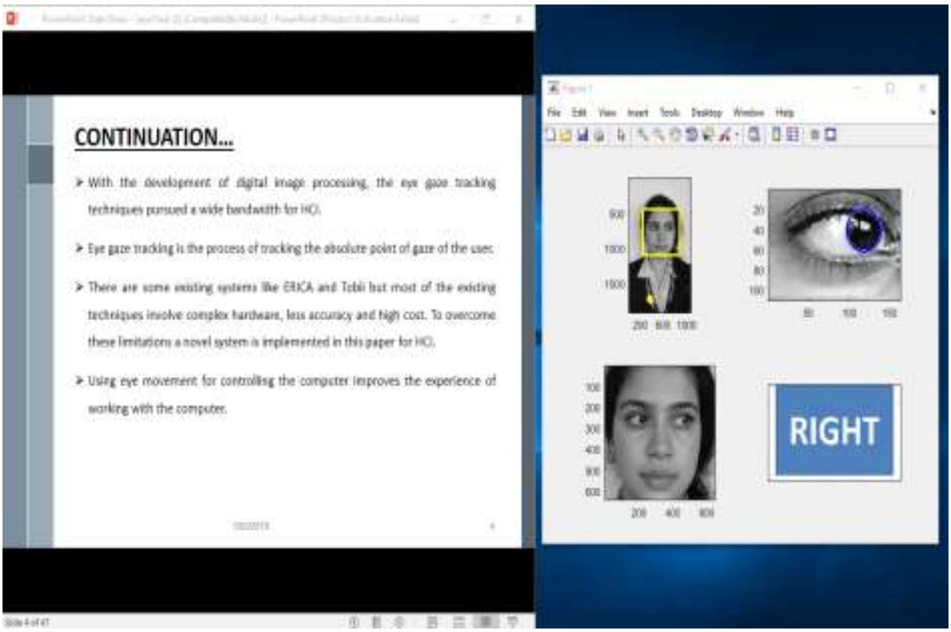


Fig. 10.1 Right Gaze Estimation

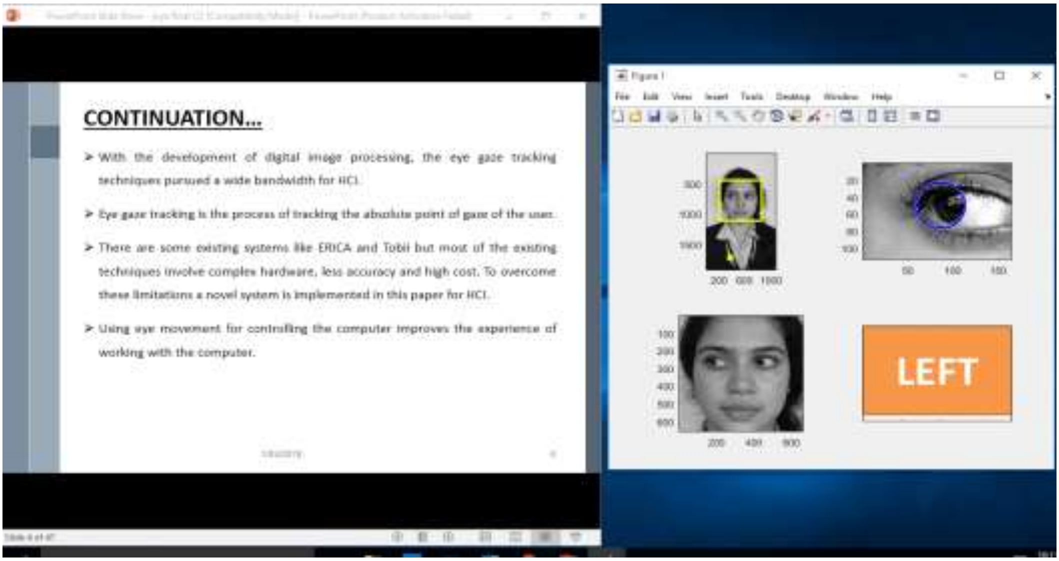


Fig. 10.2 Left Gaze Estimation

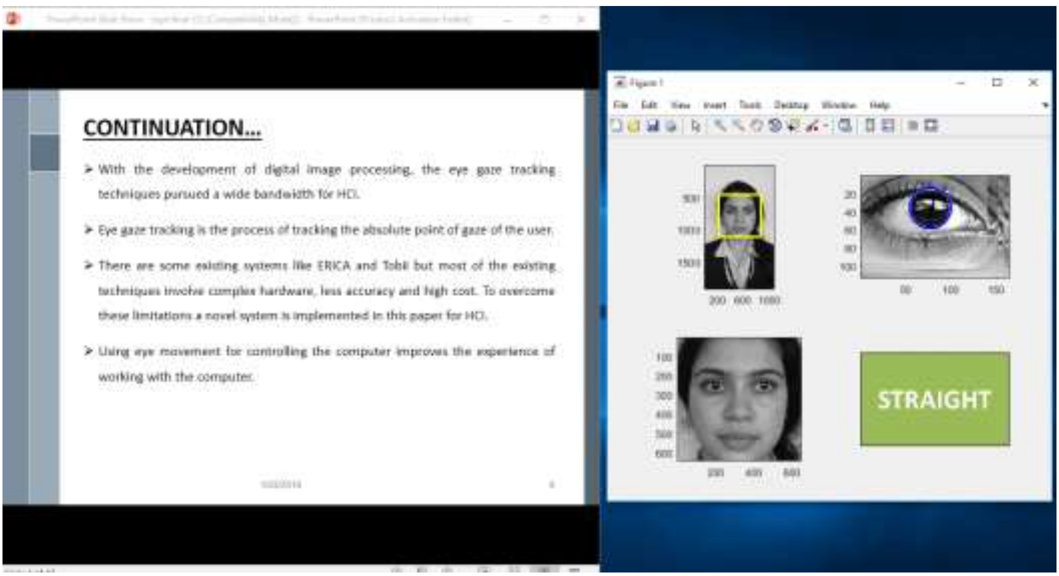


Fig. 10.3 Straight Gaze Estimation

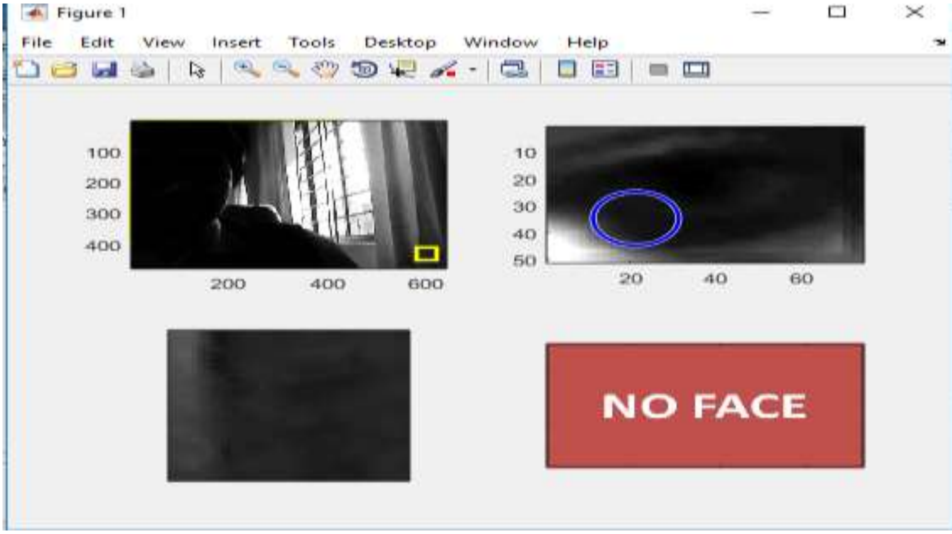


Fig. 10.4 No Face Gaze Estimation

10.2 ACCURACY

Accuracy is measured based on several factors. It varies depending on different working conditions. Some of the factors based on which accuracy is measured are-

i. Lighting conditions

ii. User with and without spectacles

10.2.1 Lighting conditions

Lighting is one of the important factors that determines the accuracy of the program. In this case, it is tested under two conditions-

- Good lighting conditions
- Normal lighting conditions

10.2.1.1 Good lighting conditions

The accuracy of the program when kept under good lighting conditions is computed as follows:

Accuracy = (Number of expected outcomes/ Total number of outcomes) x 100

= (13/15) x 100

= 86%

10.2.1.2 Normal lighting conditions

The accuracy of the program when kept under normal lighting conditions is computed as follows:

Accuracy = (Number of expected outcomes/ Total number of outcomes) x 100

= (11/15) x 100

= 73.33%

10.2.2. User with and without spectacles

Use of spectacles is another factor that affects the accuracy of the project . It is tested under two conditions:

- User with spectacles
- User without spectacles

10.2.2.1 User with spectacles

The accuracy of the program when the user is wearing spectacles is computed as follows:

Accuracy = (Number of expected outcomes/ Total number of outcomes) x 100

= (10/15) x 100

= 66.66%

10.2.2.2 User without spectacles

The accuracy of the program when the user is not wearing spectacles is computed as follows:

Accuracy = (Number of expected outcomes/ Total number of outcomes) x 100

= (14/15) x 100

= 92%

CHAPTER 11

Conclusion

Eye movement as an input medium has many advantages. It is faster than any other current input media and requires no training or particular coordination for normal users. Moreover, it is convenient for disabled users Because of these advantages, many new applications have been designed in the past are still being developed in spite of the limitations of eye trackers. Several techniques exist in practice, for tracking the direction of eye gaze. All these techniques are either intrusive or less accurate, hardware complications involved and hence are costly. This paper presents a cost effective system for physically disabled people to interact with computers using their eyes. In the algorithm implemented, the methods used for eye tracking including face detection, eye detection and pupil localization produces promising results and achieves an appreciable speed. We think these applications can be incorporated along with modern applications like cursor control using eye gazes, open files using eye blink etc in order to achieve much more benefits. By making some specific customizations in each software, we may control the entire software by tracking gaze.

CHAPTER 12

Future Enhancements

Future innovations can include

- **Detection of eyes under spectacles**.

  The project can be further modified to include detection of eyes under spectacles as many users may use spectacles for better vision and the project could then be able to cater to such users.

- **Increasing detection under dim light**

  To increase the detection of eyes under dim light as all the users may not be sitting under good lighting conditions. This can give greater flexibility to users as the comfort of users is main priority.

- **Eliminate occlusions such as secondary reflection, glare, etc**

  Eliminating occlusions such as secondary reflection, glare, etc during image processing can lead to faster and more accurate detection of eye movement.

- **Opening of a file upon blink of an eye**

  Using the blink of a user to open a file can give greater functionality to the project.

References

[1] Uma Sambrekar, DipaliRamdasi, "Human Computer Interaction For Disabled Using Eye Motion Tracking", 2015 International Conference on Information Processing (ICIP)

[2]Arpita Ray Sarkar, G. Sanyal, SomajyotiMajumder, "A Hybrid Approach for Eye-centre Localization for Estimation of Eye-Gazes using Low-cost Web Cam",Spring Conference on Computer Graphics (SCCG 2006); Bratislava, Slovakia; Apr 2006; ACM Press

[3] Asst. Prof. Sasinas Alias Haritha Z A , Raveena P.V , Arun K.S , Nithya P.N , Vishnu Balan M , Sabareesh Krishnan, " Eye Tracking System Using Isophote Eye Center Detection With Blink Perception", International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016

[4] Muhammad Awais, NasreenBadruddin, MichealDrieberg, "Automated Eye Blink Detection and Tracking Using Template Matching", 2013 IEEE Student Conference on Research and Development (SCOReD), 16 -17 December 2013, Putrajaya, Malaysia

[5] JianBingXiahou, Hao He, Ke Wei, Yingying She, “Integrated Approach of Dynamic Human Eye Movement Recognition and Tracking in Real-time”, 2016 International Conference on Virtual Reality and Visualization

[6] Daniel Gêgo, Carlos Carreto, Luis Figueiredo, "Teleoperation of a mobile robot based on eye-gaze tracking", 2011 Annual International Conference of the IEEE.IEEE, 2011. p. 4227-4230.

[7] Liling Yu, JiangchunXu* , and Shengwang Huang, “Eye-Gaze Tracking System Based on Particle Swarm Optimization and BP Neural Network”, 2016 12th World Congress on Intelligent Control and Automation (WCICA) June 12-15, 2016, Guilin, China

[8] Takeshi Takegami, Toshiyuki Gotoh, “A Hough Based Eye Direction Detection Algorithm without On-site Calibration,” Proc. VIIth Digital Image Computing: Techniques and Applications, Page(s): 459-468, Year: 2003.

[9] D. H. Ballard, “Generalizing the hough transform to detect arbitrary shapes”, Page(s): 714-725, Year: 1987.

[10] Mark S. Nixon, Alberto S. Aguado, “Feature Extraction and Image Processing”, Newnes, Oxford Auckland, Boston, Johannesburg,Melbourne, New Dehli. Page(s): 173-201, Year: 2002

Appendix A-MOVEMENT OF SLIDES

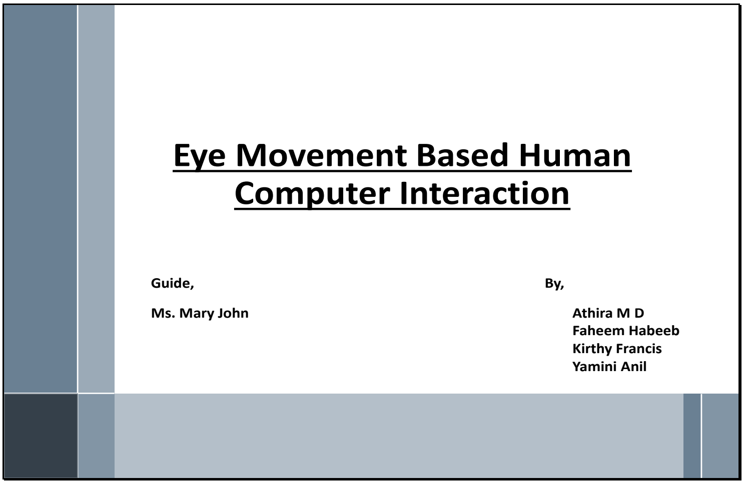
```
clear all;
clf('reset');
clc;
clear;
imaqreset;

import java.awt.Robot;
import java.awt.event.*;
key = Robot ;

camera = imaq.VideoDevice('winvideo',2,'YUY2_640x480');
camera.ReturnedColorSpace = 'rgb';
camera.ReturnedDataType = 'uint8';
right=imread('RIGHT.jpg');
left=imread('LEFT.jpg');
noface=imread('no_face.jpg');
straight=imread('STRAIGHT.jpg');
detector = vision.CascadeObjectDetector();
detector1 = vision.CascadeObjectDetector('EyePairSmall');
z = 1;
while z
    la_imagen = step (camera);
     if size(la_imagen,3)==3
        la_imagen=rgb2gray(la_imagen);
```

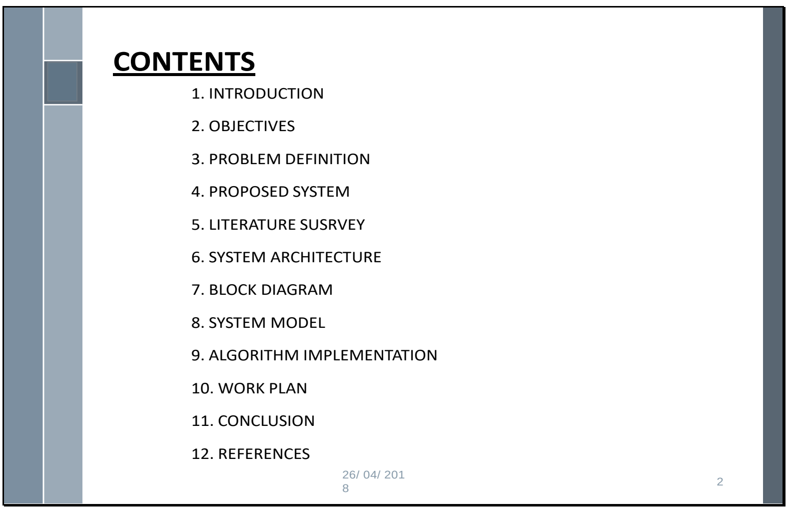
```
end
la_imagen = la_imagen;
img = flip(la_imagen, 3);
bbox = step(detector, img);
 if ~ isempty(bbox)
    biggest_box=1;
    for i=1:rank(bbox)
        if bbox(i,3)>bbox(biggest_box,3)
            biggest_box=i;
        end
    end
    faceImage = imcrop(img,bbox(biggest_box,:));
    bboxeyes = step(detector1, faceImage);
    subplot(2,2,1),subimage(img); hold on;
    for i=1:size(bbox,1)
        rectangle('position', bbox(i, :), 'lineWidth', 2, 'edgeColor', 'y');
    end
    subplot(2,2,3),subimage(faceImage);
    if ~ isempty(bboxeyes)
        biggest_box_eyes=1;
        for i=1:rank(bboxeyes)
            if bboxeyes(i,3)>bboxeyes(biggest_box_eyes,3)
                biggest_box_eyes=i;
            end
        end

        bboxeyeshalf=[bboxeyes(biggest_box_eyes,1),bboxeyes(biggest_box_eyes,2),
bboxeyes(biggest_box_eyes,3)/3,bboxeyes(biggest_box_eyes,4)];
        eyesImage = imcrop(faceImage,bboxeyeshalf(1,:));
```

```
        eyesImage = imadjust(eyesImage);
        r = bboxeyeshalf(1,4)/4;
         [centers, radii, metric] = imfindcircles(eyesImage, [floor(r-r/4)
         floor(r+r/2)],'ObjectPolarity','dark', 'Sensitivity', 0.93);
        [M,I] = sort(radii, 'descend');
        eyesPositions = centers;
        subplot(2,2,2),subimage(eyesImage); hold on;
        viscircles(centers, radii,'EdgeColor','b');
        if ~isempty(centers)
          pupil_x=centers(1);
          disL=abs(0-pupil_x);
          disR=abs(bboxeyes(1,3)/3-pupil_x);
          subplot(2,2,4);
          val = disL - disR
          if val < -10
              subimage(right);
              key.keyPress(java.awt.event.KeyEvent.VK_N);
              key.keyRelease(java.awt.event.KeyEvent.VK_N);
          else if val > 10
              subimage(left);
              key.keyPress(java.awt.event.KeyEvent.VK_P);
              key.keyRelease(java.awt.event.KeyEvent.VK_P);
              else
                subimage(straight);
              end
          end
        end
    end
  else
```

```
        subplot(2,2,4);
        subimage(noface);
      end
      set(gca,'XtickLabel',[],'YtickLabel',[]);
    pause (0.5);
    hold off;
end
```

Appendix B-PRESENTATION SLIDES





INTRODUCTION

- Human Computer Interaction is a trend-in technology.
- Nowadays, interaction with computer is not limited to keyboards and printers anymore. Many more advanced input technologies have been proposed and implemented in order to make interaction with computers more easy.
- The gaze of the user's eye is one of the best options for human computer interaction. For the same reason eye gaze tracking has become an active research topic in various applications.
- During this ongoing research, various non-invasive eye tracking techniques developed for HCI, either only with head movement or eye movement or the combination of both.

CONTINUATION...

- With the development of digital image processing, the eye gaze tracking techniques pursued a wide bandwidth for HCI.
- Eye gaze tracking is the process of tracking the absolute point of gaze of the user.
- There are some existing systems like ERICA and Tobii but most of the existing techniques involve complex hardware, less accuracy and high cost. To overcome these limitations a novel system is implemented in this paper for HCI.
- Using eye movement for controlling the computer improves the experience of working with the computer.

OBJECTIVES

- The main objective of this system is to propose a faster and non-intrusive method that enrich reading experience of the user and provides movement of slides in a power point presentation.
- The system track the movement of the eye and use it as a medium for human computer interaction.
- Several tracking algorithms can be implemented to track the pupil of the eye or the iris centre which helps in finding the x, y coordinates of the users' point of gaze on screen .

PROBLEM DEFINITION

- Eye movement is a promising input medium for human computer interaction.
- The main aim of this paper is to utilize eye gaze to enable the movement of slides in a power point presentation thereby improving the reading experience of the user. Here we discuss various eye tracking techniques that can be used to find the line of gaze of the user.
- The algorithms that can be used to implement these eye tracking techniques are also discussed.

PROPOSED SYSTEM

- The proposed system uses horizontal gaze gestures of the eye in order to achieve the eye controlled movement of slides.
- The proposed system includes:
  - Face/Eye detection.
  - Pupil Localization.
  - Gaze Estimation.
  - Template Matching.
  - Eye Controlled Movement of slides .

LITERATURE SURVEY

- **Uma Sambrekar, Dipali Ramdasi "Estimation of Gaze For Human Computer Interaction", International Conference on Industrial Instrumentation and Control (ICIC), Year: 2015.**

- Gaze of the human eye may act as a communication medium for limb disabled people to interact with computers.
- In order to determine gaze direction, the first necessary step is segmentation of iris from entire eye image.
- Here Daugman's algorithm is implemented that uses Integro Differential Operator (IDO) to differentiate iris boundary from sclera part of eye and to find the centroid, to determine where the person is looking.
- This method can be useful in eye tracking to handle the input devices of computer like keyboard and mouse.

LITERATURE SURVEY

- **Sasinas Alias Haritha Z A, Raveena P.V"Eye Tracking System Using Isophote Eye Center Detection With Blink Perception", International conference on Signal Processing, Year: 2016.**

- Here we combine an efficient method for tracking eye and detecting blink to produce an environment by which physically disabled persons can access the computer very easily.
- The system will track the eye of a person by camera and then these images are preprocessed which include image conversions and cropping.
- This preprocessed image is then given to two algorithms:
  1) **Isophote eye center detection** method to find the iris.
  2) **Blink detection algorithm** for detection of eye blink.
- In the post processing stage ,the user can control the mouse pointer on the screen by moving and blinking the eye.

LITERATURE SURVEY

- **Cheng-Lung Jen, Yen-Lin Chen,You-Jie Lin, Chao-Hsien Lee, Augustine Tsai, Meng-Tsan Li "Vision Based Wearable Eye-Gaze Tracking System", IEEE International Conference on Consumer Electronics (ICCE), Year: 2016.**

- Proposes a new wearable eye-gaze tracking system.
- First, the region of interest (ROI) of eye is extracted by skin detection.
- Then Hough circle detection is used to search the candidate of circles in the ROI.It is also applied to determine the status of opening or closing of eye.
- Based on the circle detection, the eye center and radius are detected by using least square based starburst algorithm. It uses pupil to track the iris center.
- Gaze position on the screen is projected by using polynomial interpolation.

LITERATURE SURVEY

- **Kyung-Nam Kim and R. S. Ramakrishna, "Vision-Based Eye-Gaze Tracking for Human Computer Interface" ,IEEE, Year: 2016.**

- A Non-intrusive eye gaze tracking that allows slight head movement is addressed in this paper.
- **Longest Line Scanning** (LLS) and **Occluded Circular Edge Matching**(OCEM) algorithm is implemented for eye tracking.
- The iris center has been chosen for purposes of measuring eye movement.
- Estimation of Gazing Point is done using Estimation Algorithms :

  1)Geometry-Based Estimation.

  2)Adaptive Estimation.
- A small 2D mark is employed as a reference to compensate for the eye movement.

26/ 04/ 2018 11

LITERATURE SURVEY

- **Xindian Long, Ozan K. Tonguz, "A High Speed Eye Tracking System with Robust Pupil Center Estimation Algorithm" , Proceedings of the 29th Annual International Conference of the IEEE, Year: 2015.**

- This paper presents a new high-speed headmounted binocular on-line horizontal and vertical eye position measurement system using image processing technology.
- The symmetric mass center algorithm provides more robust measurement for the eye position.
- To calculate the pupil center , a two-step processing algorithm is used.

  1) First processing the image to find the approximate location of the pupil.

  2) processing the image at the approximate pupil location .
- This will improve the processing rate of the image.

26/ 04/ 2018 12

LITERATURE SURVEY

- **Arpita Ray Sarkar, G. Sanyal, SomajyotiMajumder, "A Hybrid Approach for Eye-centre Localization for Estimation of Eye-Gazes using Low-cost Web Cam",Spring Conference on Computer Graphics ; Bratislava, Year: 2015**

- This paper proposes an efficient solution for developing a rehabilitation system for stroke patients that will be used for assistance in daily living of the patient through the use of eye gazes.
- Using the Haar-like cascade classifier, face detection and subsequent tracking of eye are performed.
- An improved version of Hough transform in two dimensional parametric space has been used for effective eye centre localization.
- The system has the advantage of lower computational load and faster processing of images

LITERATURE SURVEY

- **Muhammad Awais, NasreenBadruddin, MichealDrieberg, "Automated Eye Blink Detection and Tracking Using Template Matching", IEEE Student Conference on Research and Development (SCOReD), Year: 2015**

- This paper proposes an automated eye blink detection and tracking system using template matching.
- Golden ratio concept is introduced for Eye Pair Detection and Template Creation. Eye tracking is performed by template matching.
- The normalized correlation coefficient is computed for eye tracking. Eye blink detection is performed based upon this correlation coefficient .
- The system enable people with disabilities, who can only blink their eyes, to perform different day to day activities

LITERATURE SURVEY

- **Liling Yu, JiangchunXu* , and Shengwang Huang, "Eye-Gaze Tracking System Based on Particle Swarm Optimization and BP Neural Network",12th World Congress on Intelligent Control and Automation (WCICA) ,Year:2016**

- This paper proposes a new type low pixel eye feature point location method.
- This method can accurately extract the eye-gaze features, namely iris centre point and cantus points when the image pickup requirements are low.
- The eye-gaze tracking method based on particle swarm optimization (PSO) BP neural network requires only a common camera and normal illumination intensity rather than high-standard hardware.
- The system focuses on the eye-gaze tracking method only under the condition of free head movement

SYSTEM ARCHITECTURE

- The video of the user is captured, then the image processing software in the computer will capture the still images from the video.
- The face detection and eye detection is carried out using Viola-Jones Algorithm.
- The pupil detection is carried out using Circular Hough Transform.
- In order to estimate the gaze direction of the user, the pupil center is identified and template matching is performed.
- Once gaze direction is estimated, the movement of slides takes place in the desired direction with the help of Java class Robot.

BLOCK DIAGRAM

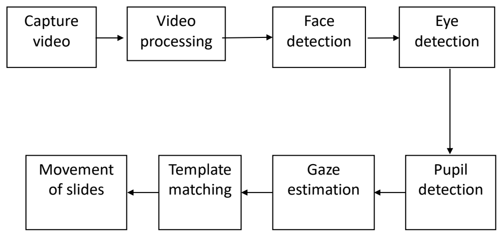


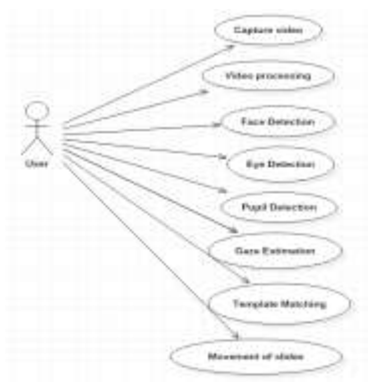
Fig. 1. Flow Diagram

26/ 04/ 201
8

17

SYSTEM MODEL

USECASE DIAGRAM

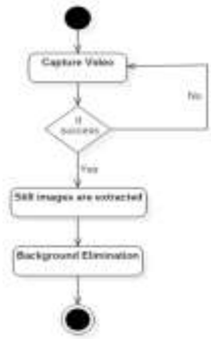


26/ 04/ 201
8

18

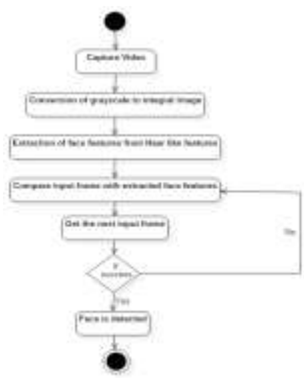
SYSTEM MODEL

ACTIVITY DIAGRAM 1



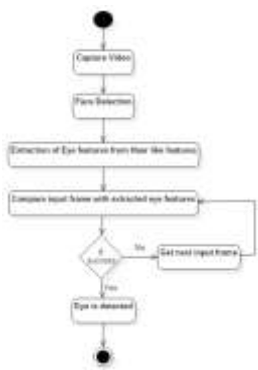
SYSTEM MODEL

ACTIVITY DIAGRAM 2



SYSTEM MODEL

ACTIVITY DIAGRAM 3



26/ 04/ 201
8

SYSTEM MODEL

ACTIVITY DIAGRAM 4

26/ 04/ 201
8

SYSTEM MODEL

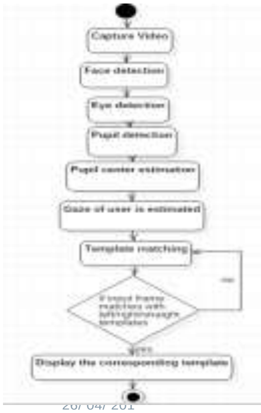
ACTIVITY DIAGRAM 5

26/ 04/ 201
8

23

SYSTEM MODEL

ACTIVITY DIAGRAM 6

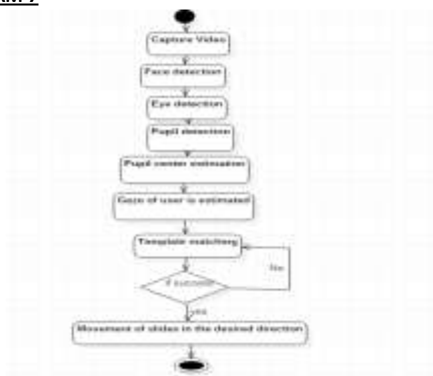


26/ 04/ 201
8

24

SYSTEM MODEL

ACTIVITY DIAGRAM 7

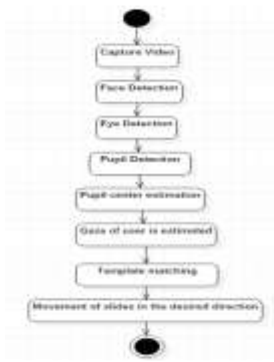


26/ 04/ 201
8

25

SYSTEM MODEL

ACTIVITY DIAGRAM 8

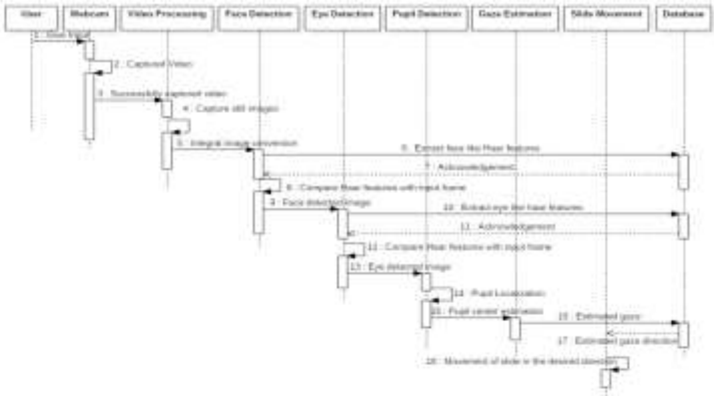


26/ 04/ 201
8

26

SYSTEM MODEL

SEQUENCE DIAGRAM



ALGORITHM IMPLEMENTATION

- **Real Time Video Input / Video Processing**

- The input given to the system is video, which is acquired by the logitech hd 720p webcam.
- The image processing software access each video frame , capture the still images and perform background elimination.

❑ Face Detection / Eye Detection

➢ Face detection and eye detection is carried out using Viola-Jones algorithm.

ALGORITHM:

Input: Grey scale image

Step 1 **:** Start

Step 2 **:** Convert grey scale to integral image.

Step 3 **:** Repeat

  Step 3.1 **:** Check input frame with first selected Haar-like feature.

  Step 3.2 **:** If feature is present in the input frame, then

    Step 3.2.1 **:** Check input frame with next Haar-like feature.

  Step 3.3 **:** Else

    Step 3.3.1 **:** Discard the frame.

  Step 3.4 **:** Until minimum 10 features are considered.

Step 4 **:** Stop

26/ 04/ 201
8

29

❑ Face Detection / Eye Detection

➢ In order to detect suitable Haar-like face and eye features, we create detector for face and eye pair using Viola-Jones. The cascade object detector uses the Viola Jones algorithm to detect people's faces, noses, eyes, mouth or upper body.

➢ An integral image at any point can be formed as the summation of pixels above and to the left of that point in the gray scale image.

➢ The advantage of the converting gray scale image to integral image is that, it reduces the computation time while evaluating the Haar-like features.

Fig. 2. Face Detection and Eye Detection

26/ 04/ 201
8

30

❑ Pupil Localization for Gaze Estimation

- To estimate the gaze of user Pupil Corneal Reflection method is used.
- When the fraction of light enters into the eye, it gets reflected from the retinal surface and forms pupil image which can be detected using Hough Transform.

26/ 04/ 2018

31

PUPIL DETECTION

- The Hough Transform (HT) is a technique that locates shapes in images.
- As the pupil area is circular, here we are using circular Hough transform to locate the pupil in eye image.

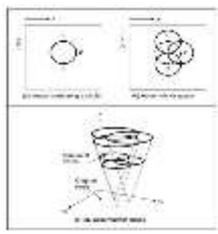


Fig. 3. Hough Transform

❑ CIRCULAR HOUGH TRANSFORM

- HT can be defined by considering the equation for a circle given by:

$$(x - x0)^2 + (y - y0)^2 = r^2$$

- Consider a circle with center (x0,y0) and radius r is assumed to be known.Then consider the points 1,2,3 marked on the circle.
- For each (x,y) of these points, we can define a circle centered at (x,y) with radius r and lies on the surface of an inverted right angled cone.
- The intersection point of these conic surfaces are found which corresponds to the pupil in the eye image.

26/ 04/ 2018

32

❑ **Continuation…**

- In MATLAB, we use the function called *imfindcircles()* for locating the pupil region.
- The function returns the center and radius of the pupil.

26/ 04/ 201
8 33

GAZE ESTIMATION

- Once the pupil is localized and center is found, we will compute the distance from right and left edge points to the center point.
- Based on this distance value, we estimate the gaze direction.
- For estimating the gaze direction we perform Template Matching where the input frame is compared with the Right, Left, Straight and No face templates and display the matching templates.

26/ 04/ 201
8 34

MOVEMENT OF SLIDES

- In order to perform the movement of slides we import the Java class Robot.
- For moving slides, there needs a keyPress and keyRelease.
- By assigning key to Robot class. We will automate this keyPress and keyRelease functions and the movement of slides takes place in the desired direction.

26/ 04/ 2018 35

SLIDE MOVEMENT ALGORITHM

Input: Gaze estimated image.

Step 1 : Start

Step 2 : Load the power point document to be opened into the MATLAB.

Step 3 : Get the input frame and check for its gaze direction.

Step 4 : If direction is towards left, then

  Step 4.1 : Movement from current slide to previous slide.

Step 5: If direction is towards right, then

  Step 5.1 : Movement from current slide to next slide.

Step 6 : Otherwise

  Step 6.1: No movement.

Step 7 : Stop.

26/ 04/ 2018 36

WORK PLAN

OBJECTIVES	ACTIVITIES	WHO	WHEN	OUTCOMES
1.Video Processing	a) Capture video using webcam b) Capture Still Images c) Background Elimination	Yamini Anil	Week 1 Jan 20,2018	Successful video processing.
2.Face Detection	a) Integral image conversion b) Extraction of face like Haar features using Adaboost algorithm. c) Detect face by using Viola –Jones algorithm	Faheem Habeeb	Week 2 Jan 27,2018	Face region extracted from the entire image
3.Eye Detection	a) Extraction of eye like Haar features using Adaboost algorithm. b) Detect eye by using Viola –Jones algorithm	Faheem Habeeb	Week 3 Feb 6,2018	Eye extracted from the face region.
4.Pupil Detection	a) Smoothening of the image b) Pupil Localization using Hough Transform c) Find center coordinates and radius of the pupil.	Kirthy Francis	Week 4 Feb 12,2018	Pupil Localization done successfully.

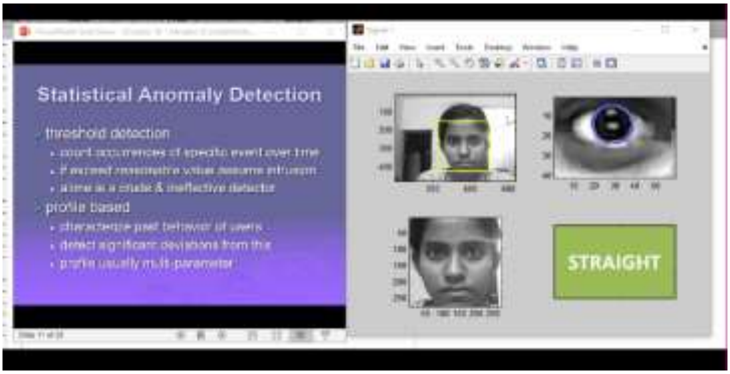
26/ 04/ 201 8 37

WORK PLAN

OBJECTIVES	ACTIVITIES	WHO	WHEN	OUTCOMES
5. Glint Detection	a) Binary image conversion b) Detect blobs c) Choose glint from blobs.	Kirthy Francis	Week 5 Feb 20,2018	Successfully detected glint from the blobs.
6.Gaze Estimation	a) Calibration of glint and pupil location	Athira M D	Week 6 Feb 29,2018	Gaze of the user is estimated successfully.
7.Template matching	a) Extraction of eye templates from the database b) Compare input frame with eye templates c)Estimate gaze direction of user	Athira M D	Week 7 Mar 5,2018	Gaze Direction of user estimated successfully.
8.Movement of slides in ppt	a)Movement of the slides according to the user's gaze estimation. b) Execution of the algorithm.	Yamini Anil	Week 8 Mar 16,2018	Movement of slides in the desired direction.

26/ 04/ 201 8 38

OUTPUT



26/ 04/ 2018 39

CONCLUSION

- Eye movement as an input medium has many advantages. It is faster than any other current input media and requires no training or particular coordination for normal users. Moreover, it is convenient for disabled users.
- These applications can be incorporated in the modern day Power Point presentations which would raise their standards.
- The methods implemented here will produce promising results.
- The system has less hardware complications involved.

26/ 04/ 2018 40

REFERENCES

[1] Jianbin Xiong, Weichao Xu, Wei Liao, Qinruo Wang, Jianqi Liu, andQiong Liang, "Eye Control System Base on Ameliorated Hough Transform Algorithm", Sensors Journal,IEEE, Volume 13, Issue 9, Page(s):3421-3429, Year 2013.

[2] Zhiwei Zhu and Qiang Ji, "Novel Eye Gaze Tracking Under Natural Head Movement", Biomedical Engineering, IEEE Transaction on, Volume 54,Issue 12, Page(s): 2246-2260, Year 2007.

[3]Kyung-Nam Kim and R. S. Ramakrishna, "Vision-Based Eye-Gaze Tracking for Human Computer Interface" ©1999 IEEE.

[4] Xindian Long, Ozan K. Tonguz, and Alex Kiderman, "A High Speed Eye Tracking System with Robust Pupil Center Estimation Algorithm", 29th Annual International Conference of the IEEE EMBS Cit Internationale, Lyon, France, Page(s): 3331-3334, Year: August-2007.

[5] Yuki Oyabu, Hironobu Takano, Kiyomi Nakamura, "Development of the Eye Input Device Using Eye Movement Obtained by Measuring the Center Position of the Pupil", IEEE International Conference on Systems,Man, and Cybernetics, Page(s): 2948-2952, Year: 2012.

[6] Uma Sambrekar, Dipali Ramdasi "Estimation of Gaze For Human Computer Interaction", International Conference on Industrial Instrumentation and Control (ICIC), Page(s): 1236-1239, Year: 2015.

[7] Paul Viola, Michael J. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision, Volume: 57, Issue: 2, Page(s):137-154, Year: 2004.

THANK YOU