Gaze-tracker: A GUI based framework for experiment design and eye movement monitoring

By
Muhammad Rizwan
Reg No: NUST201362030MSMME62113F

Supervised By
Dr. Syed Omer Gillani

Co-supervised By
Dr. Mohsin Jamil

A Report Submitted to the
Department of Robotics and Artificial Intelligence
In Partial Fulfillment of the Requirements for the Degree of
Masters in
ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

School of Mechanical and Manufacturing Engineering (SMME),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan

August, 2017

Declaration

I certify that this research work titled "*Gaze-tracker: A GUI based framework for experiment design and eye movement monitoring"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Muhammad Rizwan

NUST201362030MSMME62113F

Signature of Student

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Muhammad Rizwan

Signature of Student

Registration Number

NUST201362030MSMME62113F

Signature of Supervisor

Acknowledgements

I am especially thankful to Allah Almighty who have all the powers for considering me worthy of this educational honor.

I am also thankful to my parents and all other family members for their continuous support throughout my life, without them I wouldn't have been anywhere near this.

I would also like to thank my friend Zain Murtaza and all other friends for their constant support and advice throughout my master's program.

I would like to thank my supervisor Dr. Syed Omer Gilani and co-supervisor Dr. Mohsin Jamil along with other committee members for their continuous support and precious advice for evaluating this dissertation and for providing me with the requisite guidance and support, in order to successfully complete this work.

Dedicated to my Mom and Dad

Abstract

In this project we, propose a novel eye-gaze tracking interface that can work with a normal HD web camera. The developed framework can be used to create eye movement datasets, running behavioral experiments with images, and gaze tracking. This facilitates behavioral experiments and research in visual cognition. The developed framework also enables the user to select images, display them for specific period and record corresponding (fixations and saccades). There is also a built in calibration tool for estimating the gaze error. The developed framework uses HAAR cascade classifier to detect face and then localize eye in the region of interest. The process is robust to different light conditions with throughput of 25 to 30 FPS. The framework also enables the user to annotate the experimental data and/or stimulus with subject specific information (such as name, age, gender, profession and interests). The experiment data for each run of experiment is saved in a binary/text file for further processing.

Key Words: *Eye gaze detection, Haar cascade classifier, Face detection, Eye detection, Eye tracking*

Table of Contents

List of Figures

Chapter 1: INTRODUCTION

Eye Tracking refers to the tracking of eye movement by measuring the position of the eye's gaze point or the movement of the eyeball relative to the head. Eye tracker is a kind of device that can track and measure the eye position and eye movement information, and has a wide range of applications in the research of visual system, psychology and cognitive linguistics. On the other hand. Eye tracking is about using appropriate technical aids to record the eye or eye movements of a test person and evaluate them with regard to various issues. Eye tracking is therefore often used in market research analyzes on the effectiveness of advertising campaigns or also on usability studies, e.g. Internet sites are used as an important technique and method of evaluation as shown in Fig.1.

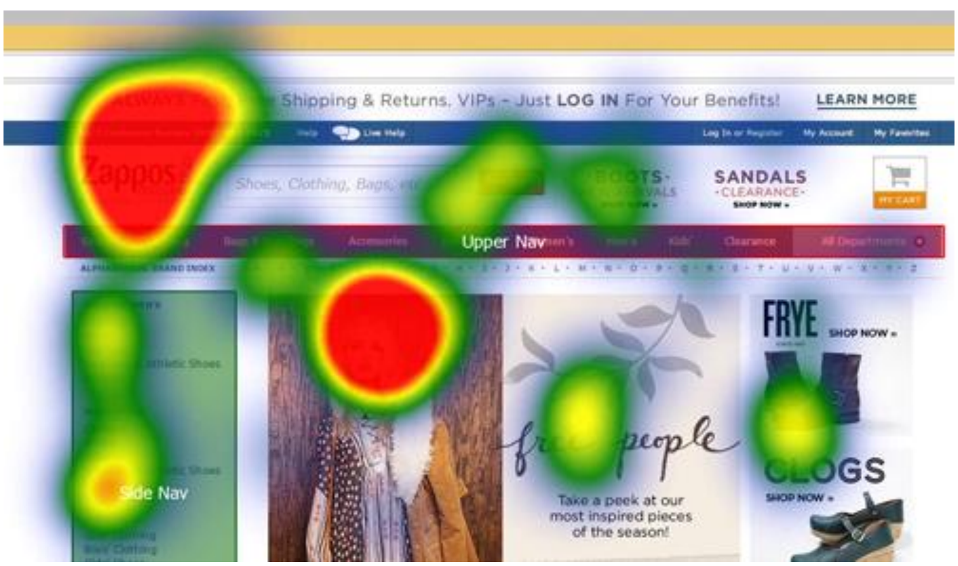

Figure 1. Visual attention on a webpage

Already in the 19th century eye movements were detected by direct observation. One of the first was the Frenchman Émile Javal, who described the eye movements while reading [1]. With the invention of the film camera, it became possible to record the direct observation and to analyze it later. This was done in 1905 by Judd, McAllister and Steel. As a true pioneer of eye registration with high accuracy is the Russian A.L.Yarbus, who in particular showed the influence of the task on the eye movements when viewing images. New methods were developed in the 1970s.

1. **Retinal afterimages,** caused by a series of strong light stimuli, produce so-called after-images on the retina, through whose position one can close the eye movement.
2. **Electrooculograms** measure the electrical voltage between the retina (negative pole) and the cornea (positive pole).
3. **Contact lens method**, which are mirrored and whose reflection is recorded by a camera.
4. **Search coil,** as shown in Fig.2, in which also contact lenses are used, which are provided with coils and are exposed to a magnetic field. The eye movement can then be calculated from the induced tension.

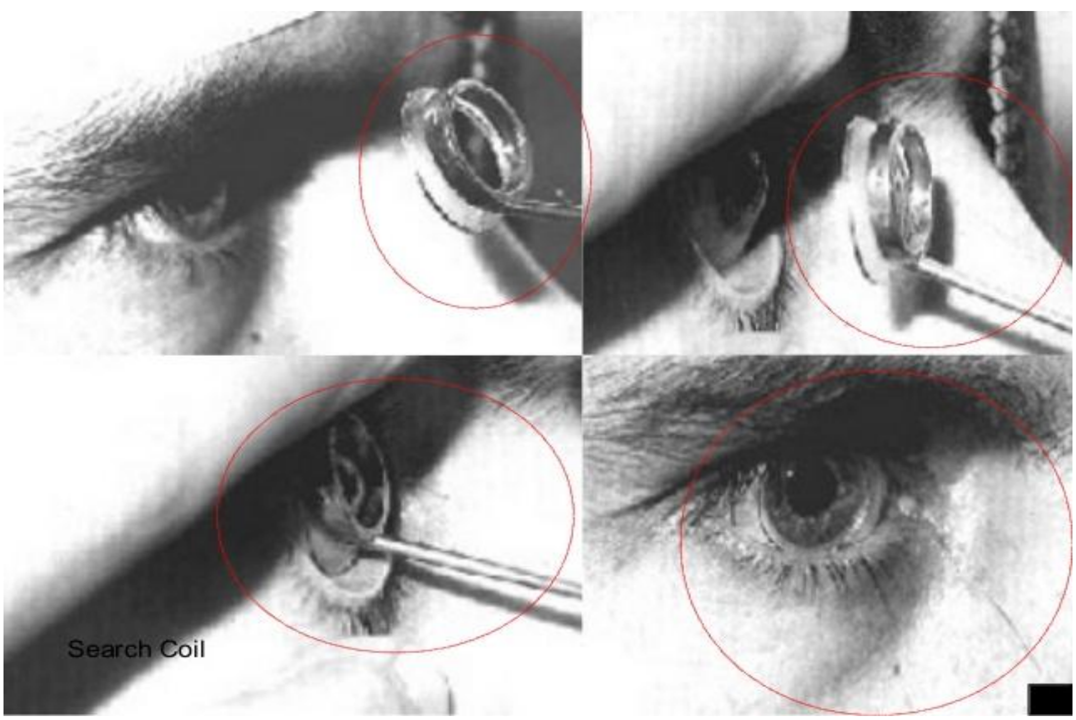

Figure 2. Search coil implanted as an eye lens

5. **Cornea reflex method,** which uses the reflection of one or more light sources (infrared or special lasers) on the cornea and the pupil position to each other as shown in Fig.3. The eye picture is taken with a suitable camera. The procedure is also referred to as "video based eye tracking".

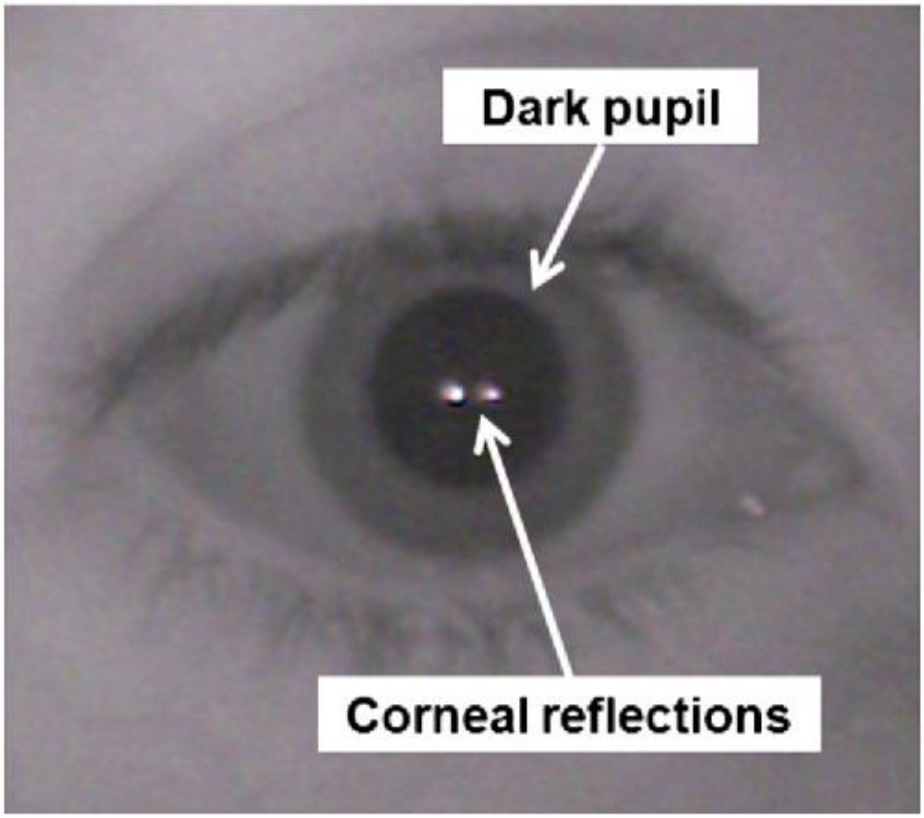

Figure 3. Corneal Reflection on pupil

1.1 Corneal Reflection

Among the techniques for recording the position of the gaze, the technique of corneal reflection is probably the most used in ergonomics because it allows a good accuracy of measurements while leaving the participant free of his movements (see Baccino & Colombi, 2001 for a description of other recording techniques, and Pottier & Neboit, 1995, for application examples).

The corneal reflection technique involves sending beams of infrared light emitted by a set of diodes to the center of the pupil. The infrared reflections returned by the cornea of the eye are then detected and allow, after calculation, to locate the center of the pupil and to know the fixing position of the eye on a target. Many systems are based on this principle. Figure 4 illustrates a test situation that uses a video camera with infrared diodes that films the eye.

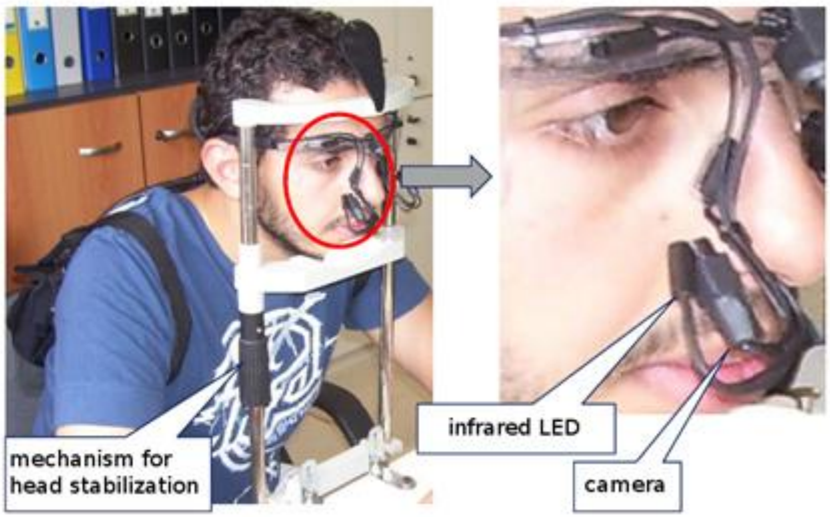

Figure 4. Infrared LED and camera to film corneal reflection

Depending on the area of application, experimental requirements and system properties, video-based eye movement (Corneal reflections) measurement systems are normally distinguish between following categories.

1.1.1 Recording Equipment

1. Simple direct recordings

In a Hunziker study on eye movements in problem solving (1970), an 8mm film camera was used to film the subjects through a glass. The visual puzzle was recorded on the glass. [4] Before the emergence of complex recording devices, simple film recordings were used, which enabled the analysis of the eye movements in certain cases by evaluating the individual images. They could be used where it was sufficient to capture the viewpoints top, bottom, right, left and center. In addition, these had the advantage that also the hand movements and the facial expression could be seen.

2. **Overhead systems**

Overhead systems, or head-mounted eye trackers primarily allow mobility. Evaluations and analyzes are essentially individual and user-centered. A scene camera also attached to the head system records a video as shown in Figure 5, which usually corresponds to the subject's field of vision. In this video, the gaze position of the subject is recorded and thus allows online, but also in the analysis offline, a tracking of the gaze path during a free operation of the subject. Generalization over several subjects and attention analysis to specific objects can be technically via 3D / 6D position detection systems (so-called head tracker)

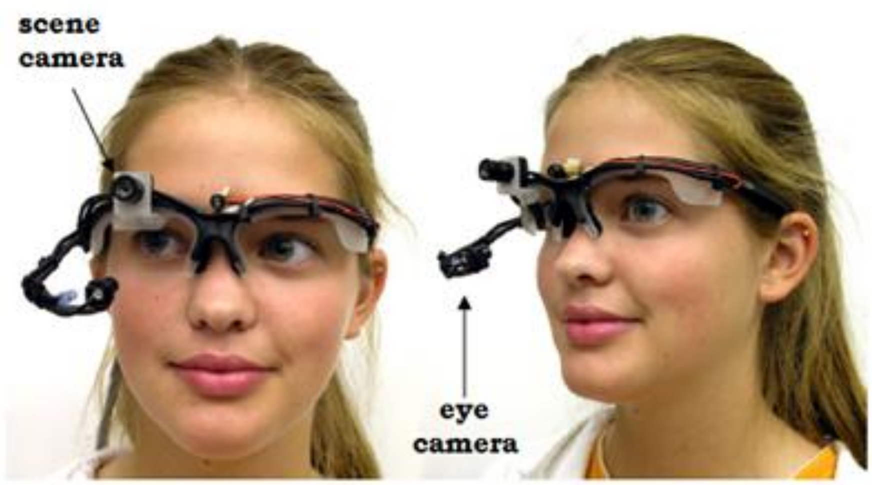

Figure 5. Overhead System

1.1.2 Remote controlled systems

Remote Eye Tracker systems enable non-contact measurements. This eliminates mechanical components such as transmission cables or chin restraints. The subject can move freely after successful calibration within a certain range of motion. An important aspect is the compensation of head movements. A person can fix a place while performing head movements without losing the fixation site. Different techniques are used:

1. **Pan-tilt systems:**

Mechanically movable components guide the camera with camera optics to the subject's head movements. Current systems achieve measurement rates of up to 120Hz.

2. **Tilting-mirror systems:**

While the camera and optics remain fixed in space, servo-driven mirrors allow tracking of the eye during head movements.

3. **Fixed Camera Systems:**

These systems dispense with any mechanically moving components and achieve the freedom of movement by means of image processing methods.

1.2 Application Areas

1.2.1 Medicine

The laser treatment for defective vision owe their success to. the use of eye trackers, which ensure that the laser treats the cornea at the intended location and monitors eye movements. Figure 6 shows the use of eye tracker in laser treatment of eyes. If necessary, the laser tracked - or switched off until the eye has the correct position again. Other systems are used in the fields of neurology, balance research, oculomotor and ocular malpositions.

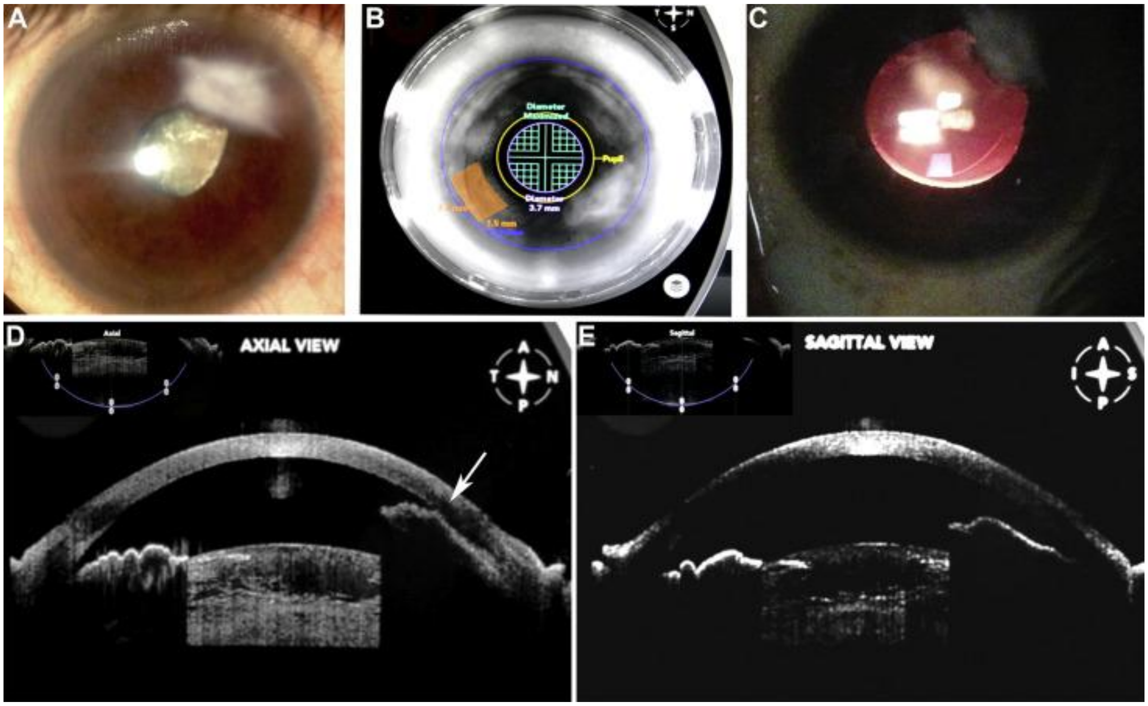

Figure 6. Use of Eye tracker in laser treatment

1.2.2 Neuroscience

Eye trackers are used as reference systems in functional imaging in magnetic resonance imaging (fMRI), magneto encephalography (MEG), or electroencephalography (EEG) systems.

1.2.3 Psychology

- Image perception
- Analysis of readings
- Analysis of learning progress (Figure 7 shows gaze during different learning tasks)

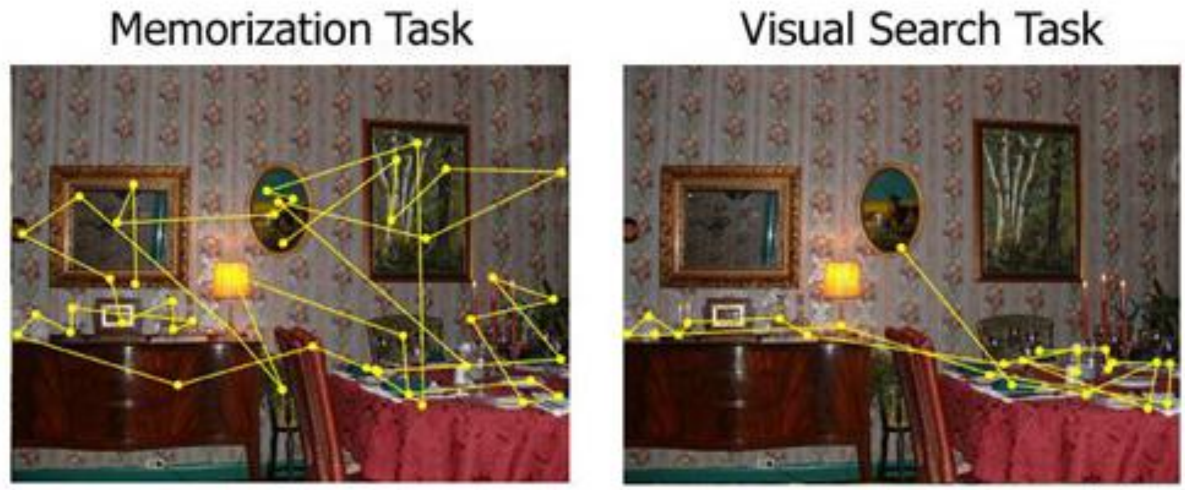

Figure 7. Gaze tracking in learning

1.2.4 Market research

- Package design (To meet the customer demand)
- Point-Of-Sale (To target the right audience)

Figure 8 shows the gaze of a customer on a McDonald's advertisement. Data can be further used by advertisement companies to improve their advertisement campaign and to meet the customer demands

Figure 8. Customer gaze on an advertisement

1.2.5 Media research

Visual attention of person using the social and other media can be observed to find out of the interests of different kind of audience as shows in Figure 9 to research on effectiveness of media on a specific audience.

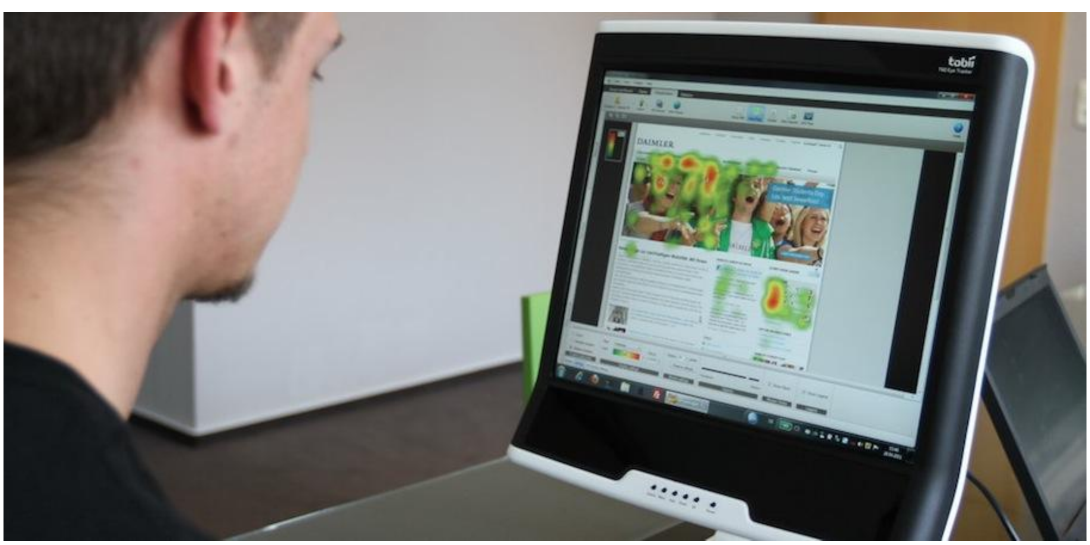

Figure 9. Visual attention of a person on media and social content

1.2.6 Human-Machine Interaction

Computer controls for physically impaired people through an eyeball. In most cases, the cursor point represents the viewpoint. An interaction is generally triggered by a fixation on an object. Powerful computer chips, efficient infrared LEDs and modern camera sensors make the previously complex eye-tracking system available to meet the needs of consumer electronics applications today. With these systems, the electronic device can detect the user's eye movement and thereby identify the user's next intention. Combined with the existing input methods, eye tracking opens up a wealth of new experience of human-computer interaction. Figure 10 represents a human-robot interaction scenario using gaze tracking.

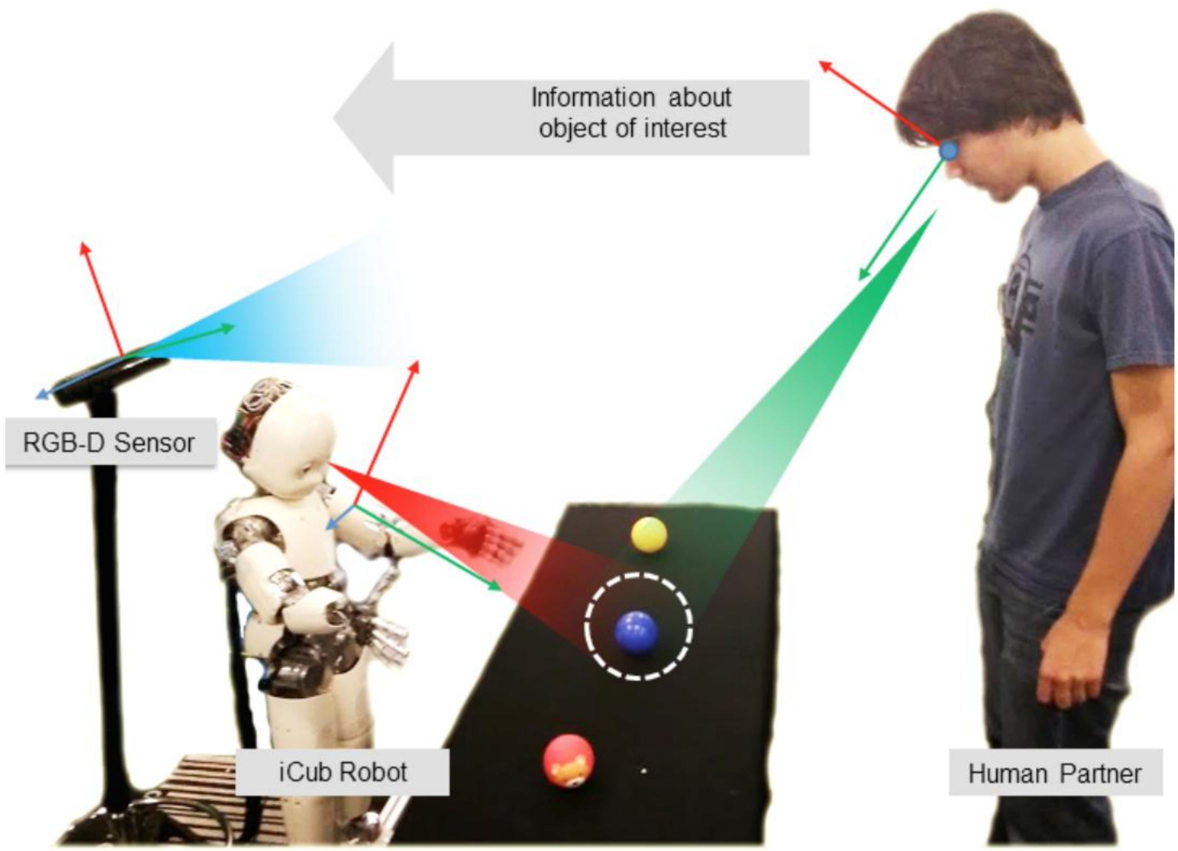

Figure 10. Eye tracking in Human-Robot interaction

Chapter 2: LITERATURE REVIEW

Eye tracking system can detect human eye movement and gaze direction. Initially, people developed eye tracking systems for the purposes of market research, behavioral analysis, and usability studies. Many of these systems use infrared light to illuminate the user's eyes and acquire images through the camera and then calculate eye movements based on the image data. As a result, these systems require specially crafted high-quality cameras, light sources and software, and sometimes require hardware accelerators to handle massive amounts of image data. Today, eye tracking is integrated into smartphones and other electronic devices with extremely powerful chips, compact camera sensors and modern high-power LEDs. In fact, many electronic devices, are equipped with camera sensors and infrared light source, but they are used to achieve facial recognition, iris recognition and other functions. So what is needed next is a suitable software that integrates eye tracking as an additional feature.

Eye tracking has also been used in different disciplines such psychology and neuroscience, industrial engineering and market research [1].

In neuroscience eye tracking can serve to diagnose neurological problems such as schizophrenia, dyslexia and Alzheimer’s disease, deficit hyperactivity disorder and autism [2]. It has also been used to study vestibular systems [3], auditory [4] and visual. To measure the position of eye (Eye Tracking) different techniques have been used over time which have their own set of advantages and disadvantages for example in electro-oculography (EOG) surface electrodes are placed around the eye to measure the small differences in potential on the skin which are due the retinal polarity, but this technique has poor accuracy and is unreliable for the recording of vertical eye movements [5]. To achieve the more resolution search coil technique is used [6].

In video-oculography (VOG) the eyes are tracked using a camera which is supported by some specific computer vision algorithms. The eye trackers which are available commercially are very expensive and can cost more than $10,000, another downside of these trackers is that they come with proprietary software which mostly dos not allow user to customize it. There are a several low cost hardware alternatives [7] [8] [9] [10] [11] [12] [13] and there are also pupil detection algorithms available [14] [15] [16] [17] [18] and there are also some VOG software which are open source [19] [20] [21]. Kalman filter is also used in recursive eye tracking [22] [23] [24] and has also been used a lot in computer vision [25] and related applications.

2.1 Background

In order to better understand the task and intention of the eye-tracking method, it is worth taking a closer look at the function of the human eye or its movements. For this purpose, different types of eye movement can be classified:

A fixation is when a certain point in the room - the so-called fixation point - focused, that is captured by the look to fix this point visually. However, it is not possible to force your eyes into an absolute rest position. Even with a fixation incessant light jitter movements of the eyeball occur, which lead to small changes in the positioning of the eye. Thus, the incoming image does not constantly fall on the same visual receptors on the retina, which are relieved thereby.

The jump from one fixation to the next, for example when switching from looking at the keyboard to looking at the monitor, is called a saccade. These are jerky and very fast eye movements, during which one does not perceive any optical information and is thus actually blind.

Most of the conscious eye movements consist of fixations and saccades, an alternating series of which make a tracking movement, with which moving objects are fixed, during the phases of fixation neurologically useful information is passed to the brain, which can also be further evaluated during the Saccade phase.

Generally, the shape and expression of eye movements, e.g. the duration of fixations or the length of saccades are greatly affected by biological factors such as fatigue, drug use, caffeine levels, sex and age or influenced by individual factors such as habits, abilities, intention of observation and interest in the observed object. But the quality of the visual object itself can also have a significant impact on eye movements, such as the complexity of a graphic representation or the difficulty level of a text. It is precisely these interactions between object and viewer, in which the seen and its cognitive processing influence each other, that make eye movement research and thus the method of eye tracking interesting from a scientific and economic perspective.

2.2 Eye Tracking Old Devices

In the early days of eye-tracking research, unwieldy and rather threatening devices were used to record eye movements that were strapped to the subject's head or face. Thanks to the technical development of the last decades, there is today eye-tracking hardware that is installed directly on the observation monitor and thanks to infrared technology works completely non-contact. The subjects are thus not affected in their natural movements and the test series is not falsified by a too artificial experimental environment in their results.

In infrared technology, the eye-tracking device is up to one meter away from the subject, while a weak infrared ray of light is directed at the eyes. After an initial calibration, a video camera records an image of the eyes that records both the pupils and a reflex point of the infrared light on the cornea - the so-called corneal reflex.

A software belonging to the eye-tracking system then calculates the eye movement from the relative distance between the pupils and the corneal reflex. These data can then be used to reconstruct and analyze exact fixation sequences, saccade movements or observation periods.

Examples

Several companies offer eye-tracking analysis as a service, in particular as an empirical basis for investigations into the effectiveness of advertising campaigns or the user-friendliness of websites, catalogs, magazines and field trials on site (shopping malls, trade fairs, etc.). But also from a scientific perspective, eye-tracking studies are being undertaken to learn more about the behavior of users in digital information or learning environments. Examples are the Poynter study or the MERIAN study of the University of Gottingen.

2.3 External systems

2.3.1. Eye-gaze system

Eye-gaze System Eye Follower: This eye-tracking system tracks eye movements with a dedicated video camera, sampling every 8.3ms (equivalent to a 120Hz sampling rate). The cameras are attached directly to the computer monitor, so that the subject does not need to be connected to other devices (glasses, helmets, etc.) and a non-contact data collection is possible. A disadvantage of the device is that pop-up windows cannot be evaluated. Further information on prices and features can be found on the manufacturer's website. With the eye-tracking provider Tobii, the recording function is integrated directly into the screen. Working of a Tobii Eyegaze system is shown in Figure 11.

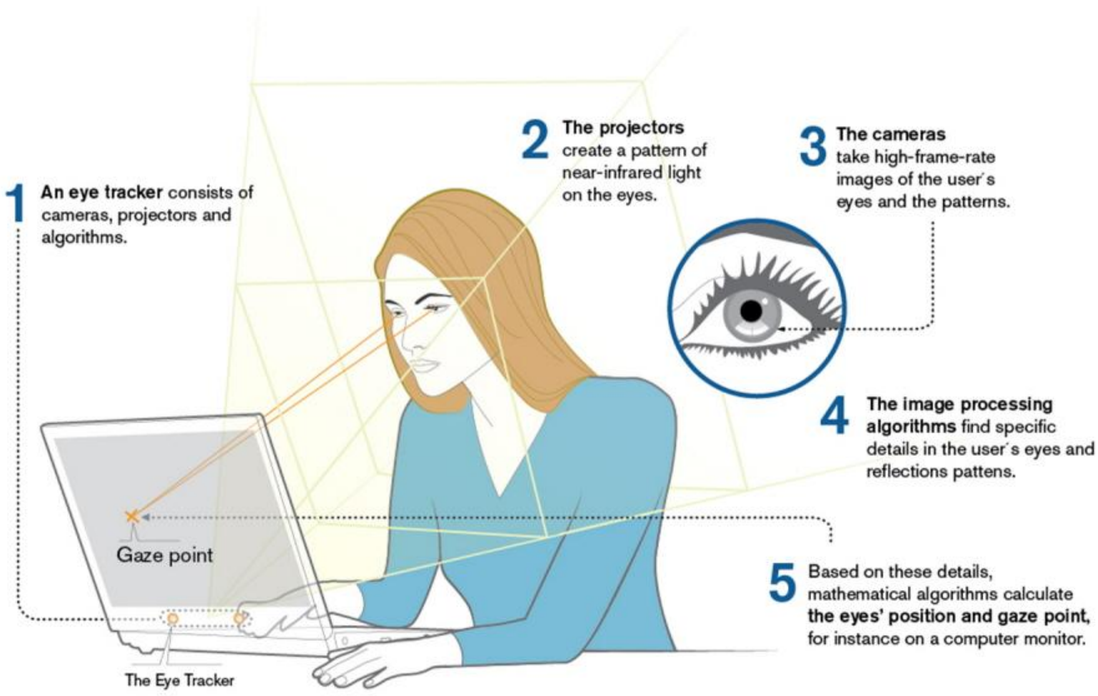

Figure 11. Working of Tobii Eyegaze System

ITU Gaze Tracker: The ITU Gaze Tracker is an open source product released by the Gaze Group, a Danish research group in April 2009. This should offer a cost-effective alternative to commercial systems. The Eyetracking is done here on your own computer with a webcam (which, however, must meet certain requirements). The GUI of ITU Gaze Tracker is shown in Figure 12 with setup and tracked eye.

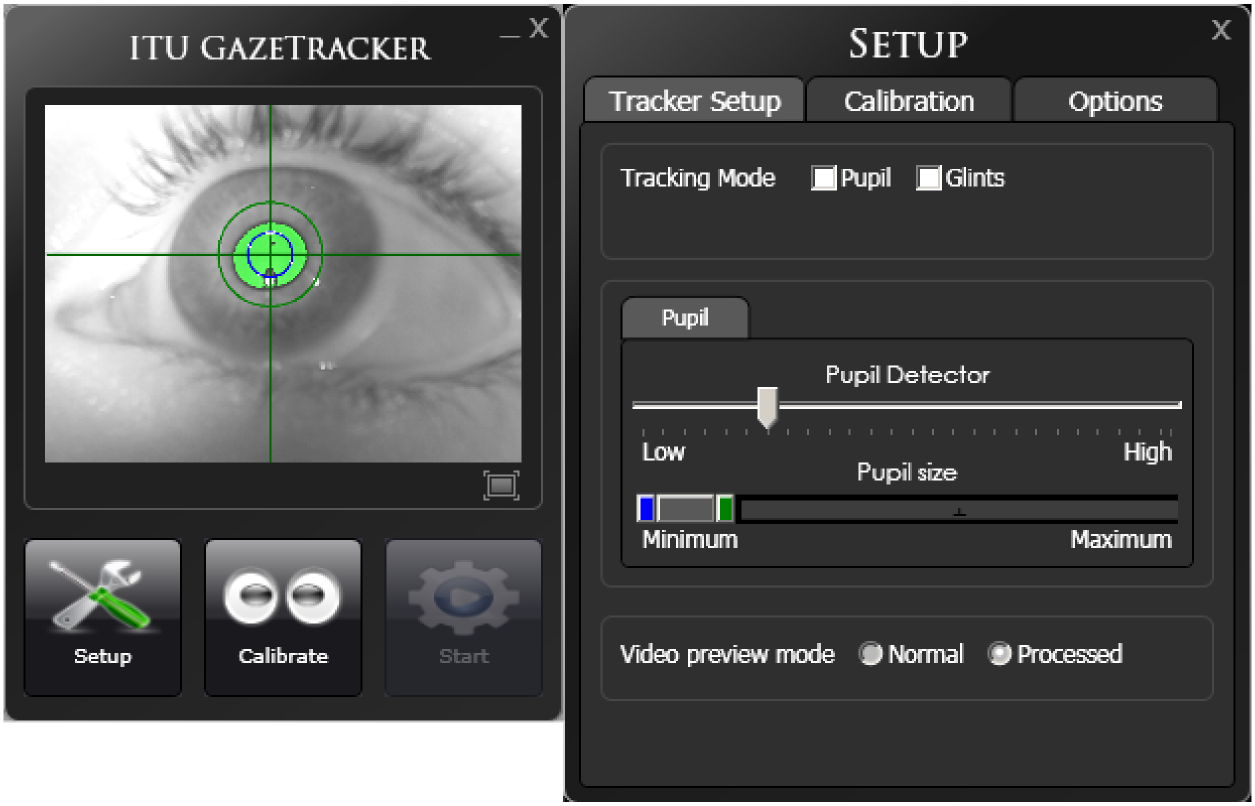

Figure 12. ITU Gaze Tracker System

2.3.2 Mobile Systems

Chronos 3D Binocular Eye Tracker: The Chronos model is a lightweight frame that is carried on the head and has two cameras attached to it to record eye movement. The sampling rate here is up to 400 Hz. These Mobile Systems can be very costly. Such mobile systems are mostly used in high end application. Usage of a Mobile System for gaze tracking in zero gravity is shown in Figure 13. As these Mobile systems are mounted on head so their position and orientation relative to eye always remain fixed rendering them easy to use in certain application which require special mobility.

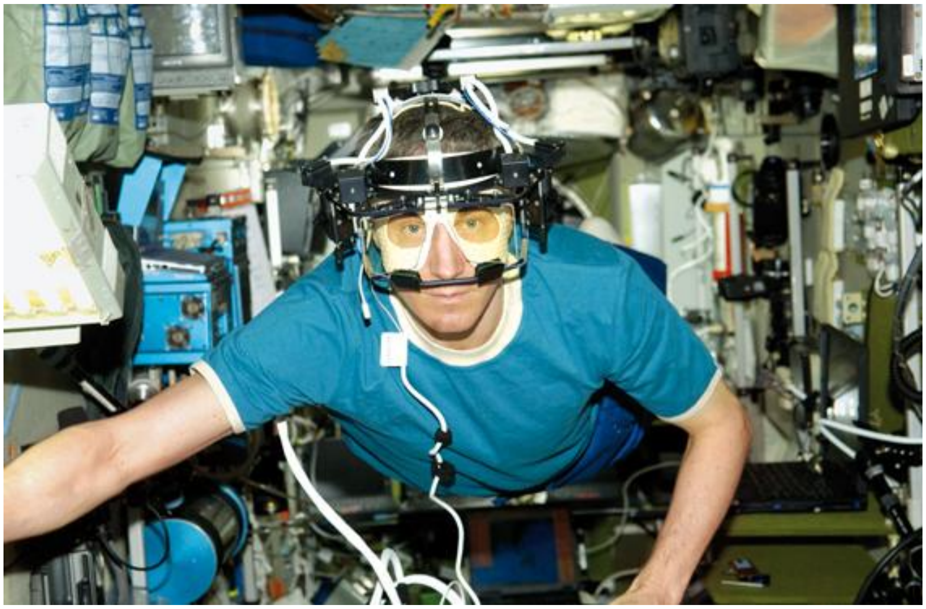

Figure 13. Mobile system for gaze tracking in zero gravity

Another mobile system is the SMI Eye LinkI Eye Tracker. Also, this system consists of a lightweight headset, which measures the eye movements by means of infrared rays and passes them on to a corresponding computer system. Further information can be found on the manufacturer's website and in the Eye Tracking research unit of the University of Bielefeld.

2.4 Evaluation options

Although the scientific findings on eye movement research are now very extensive and the eye-tracking method has a longer tradition, one must of course ask how the resulting results should be evaluated. Even if - or just because? - the technical possibilities of gaze tracking certainly have a certain fascination, the validity of this method of data collection and evaluation should be critically examined. So what can the eye tracking afford - and what can not afford?

Advantages

- Eye tracking can prove that the screen is being viewed. Without eye-tracking systems, it is often unclear whether users are ever looking at the screen, e.g. during the loading time of websites or programs.

- An analysis of the duration and number of fixations and saccades can clarify whether users are focusing on the content, e.g. read a text attentively or skim over a screen page only.
- Using eye tracking, it is possible to determine which areas of a screen page receive particular attention. Dividing the screen content into different observation areas, e.g. Header or "Top", Menu or "Left", Content or "Middle", "Right", so you can trace based on the recorded eye movements, which areas are fixed for how long.
- In particular, for web pages that are new to the user, it can be determined based on changes in the pupil diameter, whether unknown or irrelevant or expected terms and areas are covered, for which the user has been looking for.
- The data obtained can be used to compare the strategies of different users and the way they are used. For example, one and the same internet site - e.g. a sports information service - once on the topic interested test subjects show and once rather uninterested, and then to look for differences in visual perception and processing.

Disadvantage

- It is quite possible to fix things with the gaze without actually being grasped by the perception. Whether this is the case cannot be proven with eye-tracking systems.
- Also, through the periphery of the visual field, information enters the cognitive system and is processed (e.g., the scrollbars of a web page). In this regard, eye tracking cannot provide data or analysis results.
- The method of eye tracking is limited to a quantitative function. Just the fact that someone looks at the header on a screen first, does not allow any qualitative conclusions as to why this is the case.

- In addition, eye tracking cannot be used by all people with the same chance of success. Eyeglasses or contact lenses, for example, can falsify or prevent the capture of data.

2.5 Applications

Eye-tracking is a rather time-consuming method to investigate the use of learning environments by students. Although technology is becoming simpler and cheaper, evaluating the data presents researchers with challenges. If a research facility has an eye-tracking system, the analysis of eye movements offers interesting perspectives. For example, when eye-tracking is used to analyze user behavior on web pages or in digital learning environments, this method typically highlights the following aspects and questions:

- What is perceived on a screen page?
- What is the duration of individual pages and sections (e.g., menus, headlines, banners, graphic, and elements) within a page?
- How often and for how long are certain content areas covered?
- Which areas are intensively read or considered, which are only flown over?
- Do users only read headings or teasers or even entire articles?
- What is the attention relationship between graphics and text elements?

2.6 Technology

The modern eye tracking system is based on infrared LED (IRED) and high resolution camera sensors, which are used to illuminate the human eye and record the light reflected by the eyeball, respectively. Then, the image processing algorithm calculates the position of the pupil based on these original data, and then the dedicated software uses the position information of the reference object such as the screen to determine the specific position that the user is looking at. Infrared illumination ensures the desired contrast between the iris and the pupil, regardless of the color of the eye, especially in the dark or when the screen backlight is very bright.

The current detection range of these systems can reach one meter. The working distance for smartphones and tablets is typically around 30 cm, while desktops are typically around 60 cm. The screen resolution corresponds to the eye's raster size, which is about 1 cm for tablets and about 2 cm for computers. The number of infrared LEDs used, as well as the exact arrangement of the transmitter and the camera, depends on the type of application, ie the length of the working distance and the size of the area covered. In addition, but also because of the eye tracking software used by the different, this is because the geometric design depends on the algorithm can reliably detect the pupil's steering. In general, transmitters and camera sensors must be laid out at specific angles and at a distance from one another to avoid glare from the glasses or direct reflection from the eyeball to the sensor. The greater the distance, the better the signal quality, the more flexible the choice of the best distance between the user and the device.

For these applications, OSRAM has developed the Oslon Black line of products with SFH 4715A achieving a record 48% photoelectric efficiency. The 850 nm transmitter offers a typical output of 770 mW at 1 A operating current, making it the most efficient IRED for this operating current. Nano stacking can even be used to provide two launch centers on a single chip, stacked for higher output. The typical light output of the SFH 4715AS is 1340 mW at an operating current of 1 A. It offers two versions of 90 degrees and 150 degrees emission angle, covering many different designs. The Oslon Black version is 990 mW at 1 A operating current and is an ideal 940 nm light source. The component height as low as 2.3 mm is a unique feature of this IRED, making it suitable not only for today's smartphones but also for the next generation of more slim electronic devices.

A study of the learning management system Moodle by means of eye-tracking describes the publication "Cast your eyes on Moodle: An eye-tracking study investigating learning with moodle" by Gergely Rakoczi (2010). Judy Van Biljon and Marco Pretorius also used eye tracking as part of a usability study of a learning management system. The results are documented in the article "Usability of learning management systems: Do information and communication technology skills widen the gap?" (2009). At the Center for Empirical and Experimental Business Administration of DHBW, a lecture room was equipped with eye-tracking technology. A total of 20 devices and their analysis software now enable the ZEEB to train up to 20 students simultaneously and integrate the eye tracking technology as a research tool into the curriculum. ETIZ is a group of interests that formed in 2013 in Zurich. It consists mainly, but not only, of eye tracking interested students and academics of the ETH and the University of Zurich (UZH).

2.7 Importance of Gaze Tracking

Eye tracking systems are typically sold with data analysis software. These softwares make it possible to trace the ocular paths on a given surface and provide indications such as the frequencies of visual fixations on various zones of the screen. Imagine that we have registered a user's attachment points on this homepage. Once the recording is complete, the visual fixations are shown on the same page. Remember however that this is the visual journey of a user. But are we interested in the course of a single individual? Would not it be more interesting to have "patterns" of exploration? In other words, are there not courses that would be specific to certain categories of Internet users or courses that would be linked to particular exploration objectives?

Unfortunately, it is not possible to answer these questions with the software provided. To do this, you will need to process the data using other statistical analysis software and make "pattern" classifications (see, for example, Rousseau, Loslever, & Angué, 1995). These analyzes are longer and therefore more expensive.

The software provided allows you to define areas of "interest". In Figure 14, these areas are indicated by colored areas. Using these areas, you get setting frequencies for each zone. You can also calculate the percentage of users who have viewed each of these areas.

- But is the static presentation of a homepage representative of "natural" website exploration situations?
- How many times have you been confronted with web pages without being able to scroll the elevator when pages could not be displayed entirely in the browser?
- How many times did you end up on a homepage completely by chance with no objective of consultation?

Figure 14. Region of Interest on a Webpage

Website consultation is usually motivated. Looking for specific information (the address of a hotel less than 600 Francs in the 5th arrondissement of Paris), or if a user want to buy something (eg, the last CD of the band Coldplay you know). It's a safe bet that in these situations the visual behavior of Internet users will be different. The different areas of the page can then be explored differently.

The use of eye tracking can be an interesting tool for studying the visual behavior of Internet users. But like any tool he cannot answer all the questions. In addition, the interpretation of the results must be "prudent".

- Does not having an element fix mean that element has not been perceived?
- How long does it take to set a point for the information to be processed?

The visual exploration of a web page can be guided by the organization of the elements of the page but this exploration is also guided by objectives. The visual exploration behavior is not "passive". This is particularly shown by the results of the studies on the homepages.

To understand the behavior of Internet users and to design sites that are adapted to the objectives that they pursue, the studies will have to be more realistic and better controlled. This is to observe users when performing tasks representative of those that can be conducted on sites.

Chapter 3: METHODOLOGY

The project purpose was to design and develop a system that can allow users to perform gaze tracking experiments with their webcams.

The project was divided into parts namely

- Eye Tracker
- Eye Calibrator
- Gaze Tracker
- GUI Development and Subject Related Data Entry
- Experiment data saving on Disk for future use

The eye tracker purpose was to find the position of eye in the image captured by the low quality normal web cams. It computes the position of both eyes in the face image after performing some enhancements into the face image.

The purpose of the eye calibrator was to take recordings of different eye positions of the person while asking him to look at different places on the screen that will further be used in the gaze tracking.

The module takes the positions of the eye in the image, take the calibration data earlier recorded for that specific person and compute a gaze position on the screen for that user.

Subject related data entry was also provided to annotate the recorded data that asks user to enter data about user and then after the experiment is performed it saves the data on the disk for further future use.

3.1 Cascade Classifier

Cascading classifier is normally used to identify certain objects in digital image and it can also be used in statistics. In cascading classifier we study future observations with the help of previously obtained observations by combining different classifier. By using a lot of positive and negative samples we instruct cascade classifier. After the completion of this process, we move to our final goal. Our final goal is to observe an object onto the surface of an image. In order to observe an object onto the surface of an image, we explore each and every position of an image for the classifier. The first cascade classifier is proposed in 2001 by Michael Jones and Paul Viola. It resolves the problem of face detection from digital image.

3.2 Haar Cascading Classifier

Haar Cascade Classifier is an object detection technique. This technique is first suggested in 2001 by Michael Jones and Paul Viola in their work "Rapid Object Detection using a Boosted Cascade of Simple Features". Haar Cascade Classifier give the ability to the system to observe things on the image using algorithms, so that the system will prepare to detect an object. For this purpose, cascade algorithm required bunch of negative and positive images. Negative images imply the images without human faces and positive images imply the images with human faces. In Haar Cascade Classifier these positive and negative images is used to obtain the different characteristics from an object. Haar Cascade Classifier is similar to convolutional kernel that is used for embossing, blurring and edge detection from an object. Here we have three different types of features that are helpful to get attributes from an object. Haar like features is much fast as compared to other available features in image processing. First one is Edge features, second is Line features and third is Four-rectangle or center-surround features

.

These features represents particular characteristic of a certain area of an image. But before we use these features we used original pixel values of an object. The use of pixel values for object detection is much more complicated because in single image we have thousands of pixels, so that there computation is difficult task. Haar Cascade Classifier adds pixel intensities of each region of an image after examine the contiguous rectangular regions of an object. After this Haar Cascade Classifier find the differences of the sum of pixel intensities and these differences is used by the system for object detection. In this way Haar Cascade Classifier find out many characteristics of each kernel. If we have 24x24 windows then it means that cascade algorithm find 160000 different characteristics. For finding characteristic from an image we required to add white and black rectangles pixels. For this purpose we use the concept of integral images. This concept is first proposed by Frank Crow in 1984 for computer graphics. Using integral images algorithm we find out sum of all values efficiently that is present in the region of an image. Integral images sum up pixels of an image and it will do in constant time. From 160000 characteristics that we find using Cascade algorithm, a large number of characteristics is unrelated for object detection. Now the question is how Cascade algorithm identifies the relevant and irrelevant characteristics of an image. But first we explain with the help of figures, the relevant and irrelevant characteristics of an image.

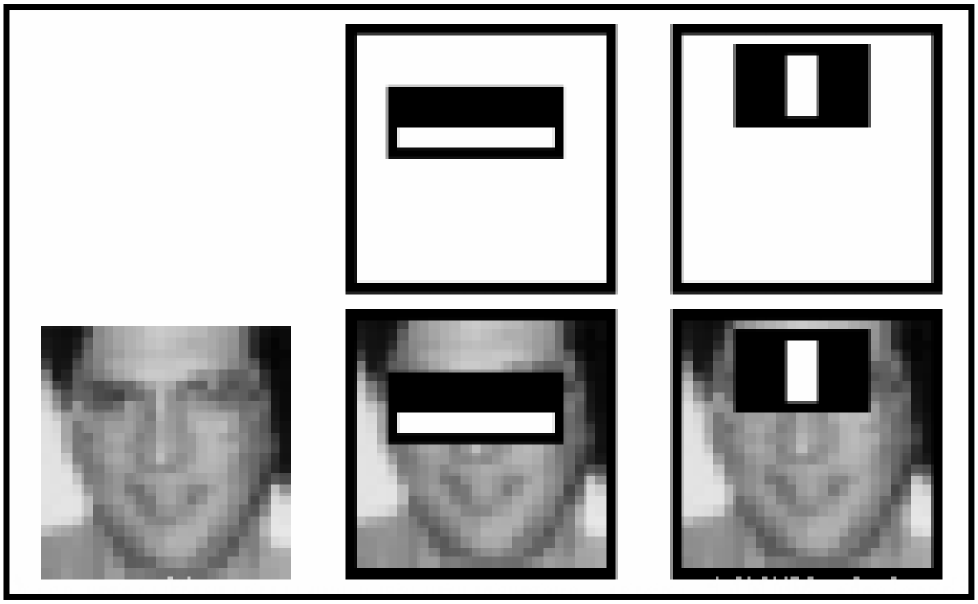

Figure 15: Haar Cascase features

In the above figure 15. Top row is representing 2 good features. The first characteristic which is helpful for us is the area of the eyes. Because the area around the eyes is dark as compared to the other area of face such as nose, lips and checks. The second characteristic which is helpful for us is that the color of eyes is dark as compared to the bridge of nose. So in this figure these two characteristics are relevant for object detection while other characteristics are irrelevant.

Cascade algorithm identifies the relevant and irrelevant characteristics of an image with the help of Adaboost. Adaboost also known as Adaptive Boosting is an algorithm, which plays vital role in machine learning. This algorithm is first proposed by Yoav Freund and Robert Schapire. Adaboost is used with many other learning algorithms to improve production. First we take some images for training purpose and apply all features on training images. For the avoidance of error we select only those features whose error rate is near to zero and they are easily identifies the faces and non faces images. For each features Adaboost identify the gateway because it will divide the faces into positive and negative category. In the start of this process each image has an equal weight. But the weight of misclassified images is increased after the identification of faces and non-faces images. This process is repeated again and again until the required numbers of features are not identified. If we have an image on the window of 24x24 and we apply 6000 features on it. After this we want to identify that in which location the face is present. In any image with the human face most of the area of that is non-face area. Thus first we identify a window for getting face area. If window contain face area, then we apply some more methods to identify required features on face area otherwise ignore this image. For this purpose we use the idea of Cascade Classifier. In Cascade Classifier, we don't apply all 6000 features on window at once. Before applying the features on window we create group of features and apply one by one. When we apply first group of features on window and we find required result then we move to next group of features, otherwise don't apply remaining group of features on this window.

The method works with a rectangular search window, which is gradually pushed over the entire image. This window initially has a minimum size, and is enlarged with a fixed scaling factor after each pass across the image. The image section under the search window is passed to a detector, which creates a hypothesis whether or not an object of the class sought is to be seen at this location. In this way, the objects can be recognized in different positions and scales.The detector uses a number of selected features with associated thresholds. The features are calculated within rectangular blocks that have a fixed position relative to the search window. The basic forms used in Haar cascade classifier are given in figure 15.

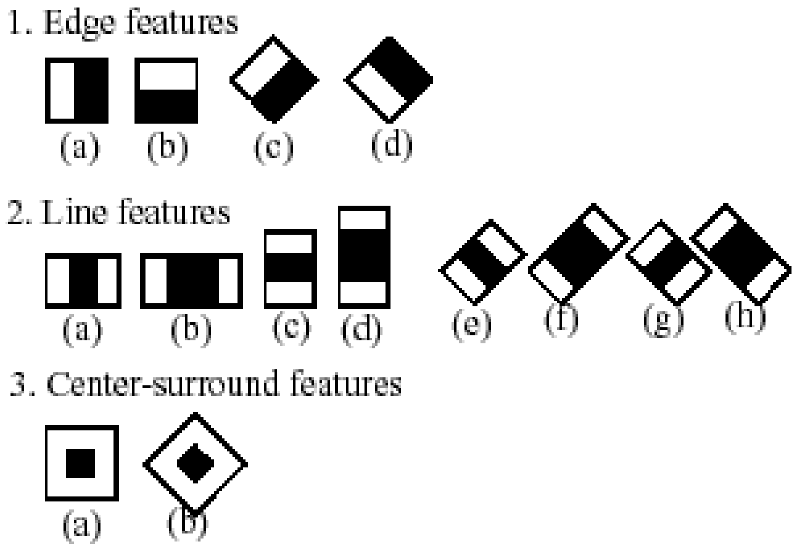

Figure 16: Haar cascade classifier basic forms

The value of a feature is calculated by subtracting the sum of all pixels in the black blocks from the sum of all the pixels in the white blocks, ie, it is a difference of integrals over rectangular image areas. Each of the basic shapes can be arbitrarily positioned and scaled within the search window as in figure 17. This results in a very large number of possible characteristics. (Within a 24x24 window there are already more than 117,000 possible features)

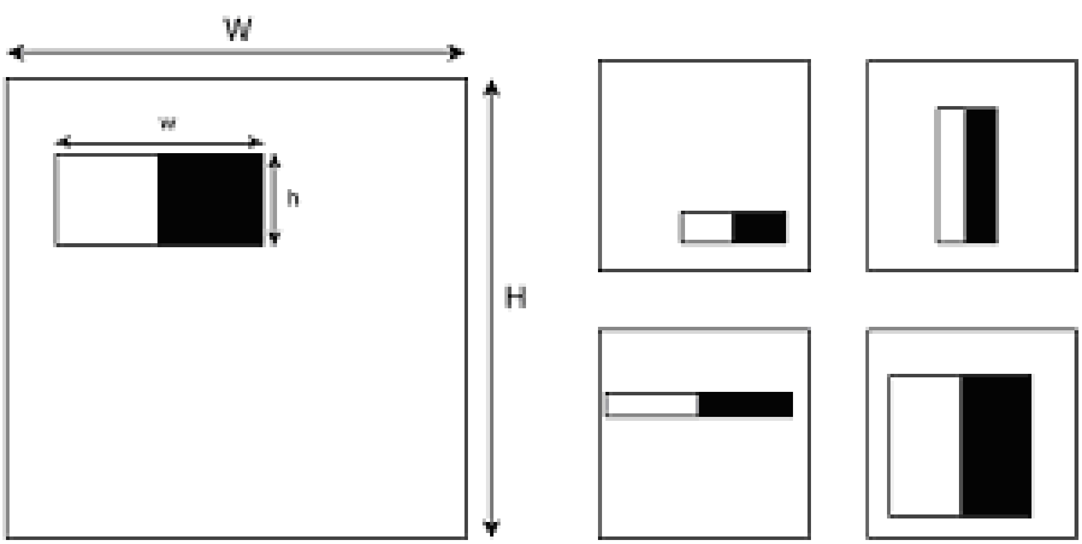

Figure 17: Basic forms arbitrary positioned and scaled

The special form of the features enables an efficient calculation in various positions and scaling stages by means of a corresponding preprocessing step, the generation of the so-called *integral* image. During the training phase, among the plurality of possible features and feature positions within the search window, those most significant to the discrimination of objects and non-objects are selected. The feature selection is done through the *AdaBoost* learning process. This method first requires a sufficiently extensive training data set of positive examples (ie image sections on which an object of the sought class can be seen) and negative examples. For a good result, a large data set with several hundred images is usually useful.

For example, if parking prohibition signs are to be detected in the image, they are characterized by a characteristic brightness distribution in the red channel. Figure 18 illustrates some positive examples that are useful for this application and that need to be specified for training.

Figure 18: Positive Examples

The positive samples should contain all possible variants in which the searched objects can occur. Parking prohibition signs can be twisted relative to the camera both in the image plane and relative to the viewer to some extent. The positive data set must reflect the variability of the objects, as otherwise restrictive thresholds or even unsuitable features could be determined. (The positive examples should include all "worst cases.") After the training, for example, the parking prohibition sign detection results in the features shown in Figure 19 in the first stage

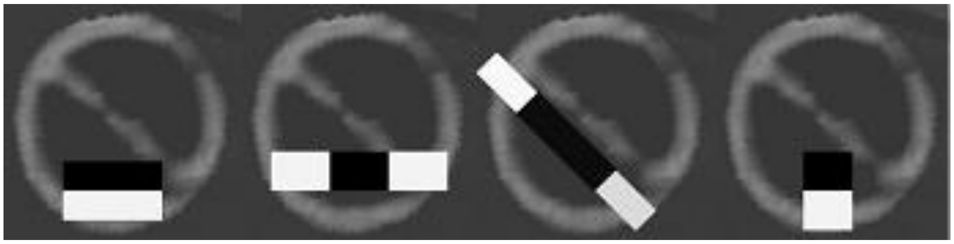

Figure 19: Parking prohibition sign detection features

3.3 Face detection

To detect objects, the feature set selected during the training must first be loaded. For detection, the entire image is scanned several times with a search window, the search window being enlarged after each iteration by a scaling factor. In this way, the image is searched at all possible positions and scaling levels for objects of the class to be detected. The face detection using haar cascade classifier implemented is given in figure 20.

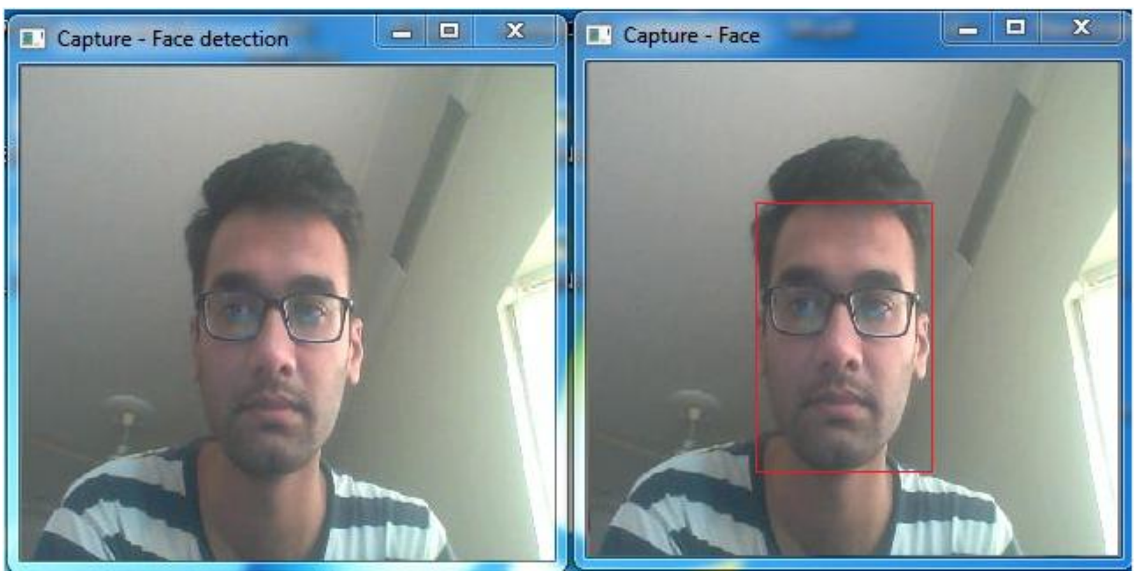

Figure 20: Face detection using Haar cascade classifier

3.4 Eye region detection

After the face detection, eye regions are extracted using the information of location of eyes on face. For this purpose let (a,b) be the upper left and (c,d) be the width and height of the detected face region respectively. The mean of the right eye center is thus calculated as (a+0.30, b+0.0) similarly the mean of the left eye center can be calculated as (a+0.7, b+0.4) [].

Gaussian filters are frequency filters which have no overshoot in the step response and at the same time maximum slew rate in the transition range. As a special feature in this filter, both the transfer function and the impulse response have the course of a Gaussian bell curve , as shown in the illustrations on the right, from which also derives the name of this filter type. Areas of application of this filter are in digital modulation methods and in the field of image processing.

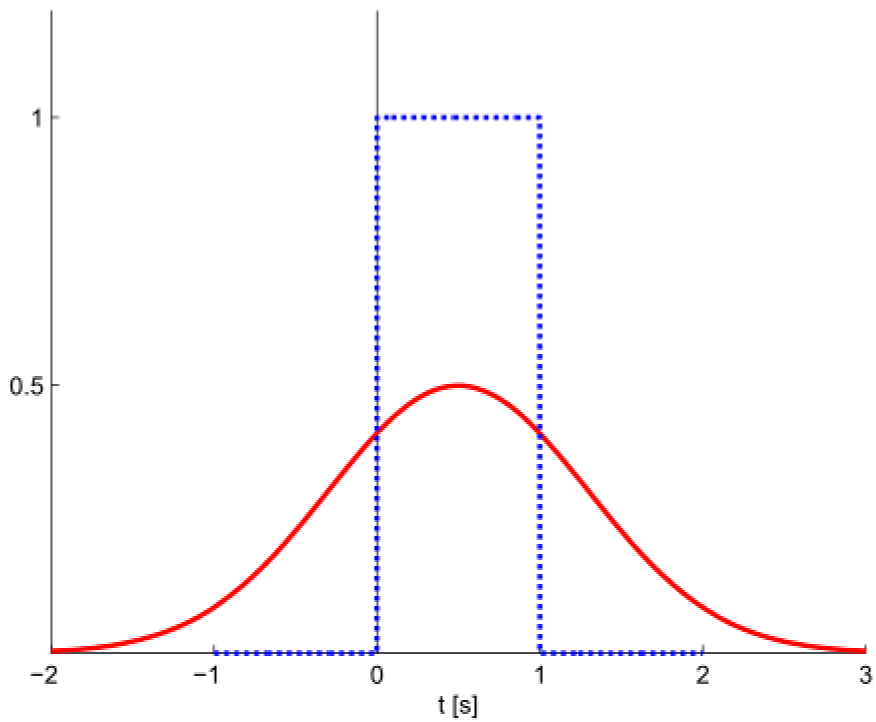

Figure 21: A rectangular pulse is converted into the signal curve

A rectangular pulse, shown in dotted blue, is converted by the pulse shaping of a Gaussian filter into the signal curve shown in red given in figure 21. In image processing Gaussian filters are used to smooth or blur the image content. It can thus reduce the image noise: smaller structures are lost, coarser structures, however, remain intact. Spectral smoothing is equivalent to a low-pass filter. Since an image has two dimensions, the impulse response must be extended to two dimensions for image processing. The impulse response has the two arguments and according to the spatial directions or practical implementations in the context of digital image processing, the discrete impulse response is usually used in the form of a two-dimensional matrix .Alternatively, in the literature, the description of Gaussian filters instead of the constants Equal to the variance used in the expression of the impulse response - which expresses the mathematical proximity of the impulse response of a Gaussian filter to the function of normal distribution . By taking advantage of the separability the computing time can be significantly reduced. Guassian blur applied to a half tone image is given in figure 22.

Figure 22: Gaussian Blur applied to a halftone image

In our algorithm we applied the Gaussian blur to the detected face region as preprocessing to smooth the noise.

3.5 Eye center localization

Vector field of image gradient can be used to find the center of a circular object, this technique is mostly used for effective eye center localization. In our work we used this technique to find the eye center. Others methods also exist for the eye center localization for example Mitchell and Kothari propose a methodology that uses the flow field charter which occurs because of significant contrast sclera and iris. But this methodology has certain drawbacks for instance it is define in discrete image space and it doesn't efficiently address the problems that occur because of eyelids, glasses and eyebrows.

In our work we implemented the eye center localization using the vector field of image gradients and introduce a mathematical method for vector field characteristics [26].

let

$c = possible\ center$

$g_i = gradient\ vector$

$d_i = normalised\ displament\ vector$

Here gradient vector and normalized displacement vector has same orientation.

$c^* = optimal\ center\ of\ a\ circular\ object$

With pixel position

$$x_i, i \in \{1, ..., N\},$$

Now

$$c^* = \arg\max\left\{\frac{1}{N}\sum_{i=1}^{N}\left(d_i^T g_i\right)^2\right\}$$

$$d_i = \frac{x_i - c}{\|x_i - c\|_2}, \quad \forall_i : \|g_i\|_2 = 1$$

Notice displacement vector is scaled to unit length to get the equal weights for all pixels. The robustness of the algorithm can be further increase by scaling the gradient vector to unit length this will allow better performance against contrast and changes in lighting.

3.6 Eye Calibrator

The eye calibrator asks user to record eye positions of the user on nine different points on the screen. During this a screen will appear to the user as shown below and the user will be asked to look at some positions on the screen as shown in figure 23.

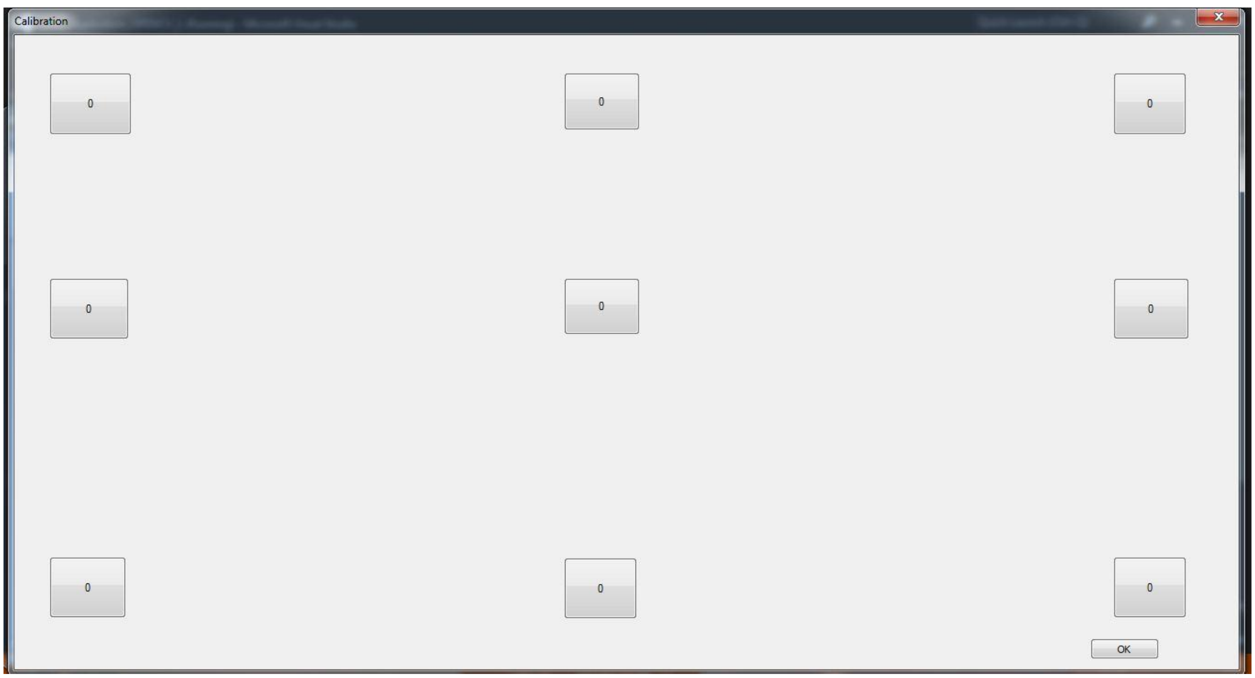

Figure 23: Eye calibrator user interface

In this image each box containing the ‘0’ letter represents a point on the screen and user is required to record the position of the eye for that position. To record the position of the eye user has to click on the respective box and keep looking at the respective box until a dialog box appears which says the position is recorded as shown in below image. Repeat the same process for all the boxes on the screen and once you have recorded positions for all the boxes click OK button on the screen.

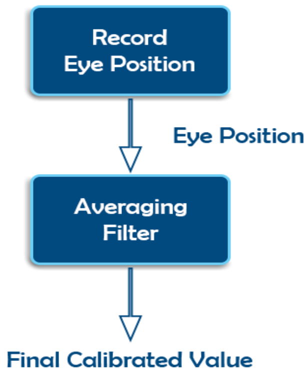

Figure 25: Eye calibrator block diagram

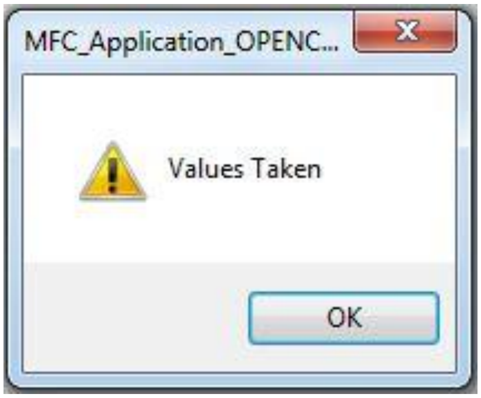

Figure 24: Dialogue box of eye calibrator

This takes twenty values of eye position for each point on the screen as in above picture, computes the average of those values and use that averaged value as the value of the eye position at that point on the screen.

The averaging filter was applied instead of the median filter because there were not many outliers either high or low, so average filter was applied in place of the mean filter.

$$\frac{1}{20}\sum_{i=1}^{20} x_i$$

3.7 Eye Gaze tracker

Gaze tracker takes the position of both the eyes in the current frame, maps them to the calibrated values obtained from the Eye Calibration and computes an approximate output of the user's gaze position on the screen for the respective calibration. This takes the position of both the eyes from the Eye Tracker module, and find the center point of the eyes in the image to get a position in the image, takes the calibration data from the calibration module and tries to map the position of the

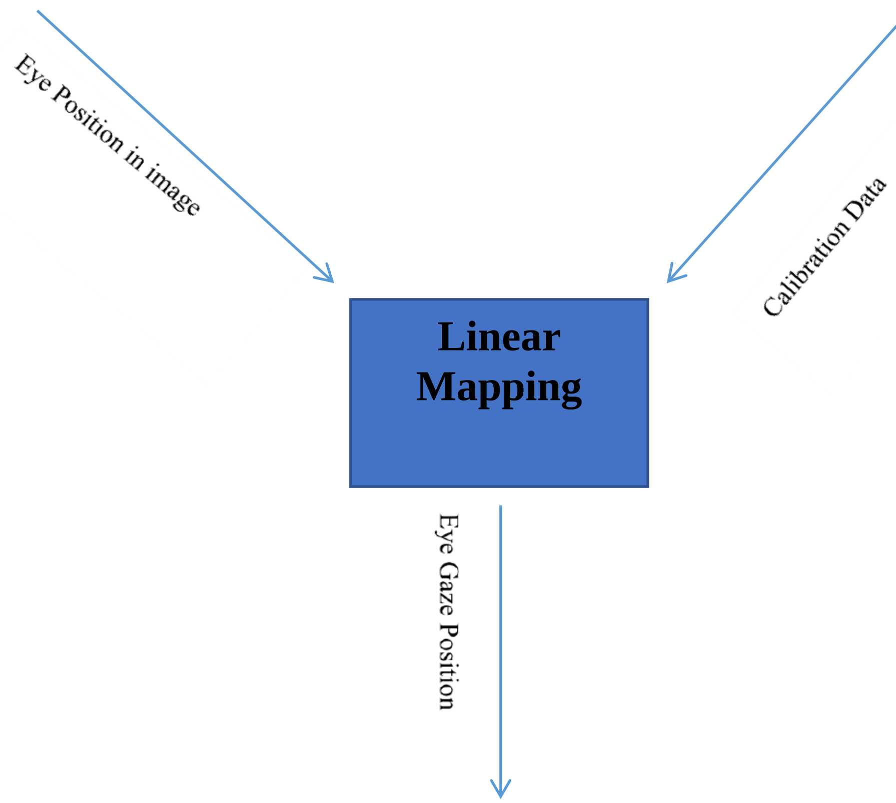

Figure 26: Flow diagram Eye gaze tracker

eyes in the image to the point in the screen by linearly mapping the recordings of positions of the eye taken by Eye Calibrator module. A flow diagram of eye gaze tracker is given in figure 26.

3.8 GUI Development

The GUI for this framework was developed using C++, IDE used was Microsoft Visual Studio and Microsoft Foundation Classes (MFC) framework was used for GUI. This provides a good enough platform to quickly develop the GUI based applications and provides one of the best debugging features to help programming.

- MFC can manage arbitrary number of windows.
- MFC can perform many functions with windows like moving, activating, clicking, resizing and with their contents also.
- MFC can handle file menu functions like Open, Save, Save As and Print with their corresponding dialogs.
- It can process input data of dialog controls.

3.8.1 MFC Environment Setting

The minimum set of tools required to run MFC application are a text editor, a C++ compiler and a linker. All these tools offered by a single Microsoft Visual Studio tool to create and run MFC application.

3.8.2 MFC's Cornerstones

This section will focus on principle mechanism and concepts used inside MFC Framework.

MFC contains building blocks for creating user interface (e.g. Buttons, Menus, and Forms etc.), basic data structures and collections like (CList, CStack and CString etc.) and also application components classes such as CWinApp, CWnd, CView etc.

Together all these user interface components, high level application components classes and helper classes so it is possible to create good looking MFC applications with any functionality.

Below Figure 27 shows relationships between MFC classes. Abstract classes shown with italic style and all classes names starts with 'C' letter as this is naming convention of MFC classes.

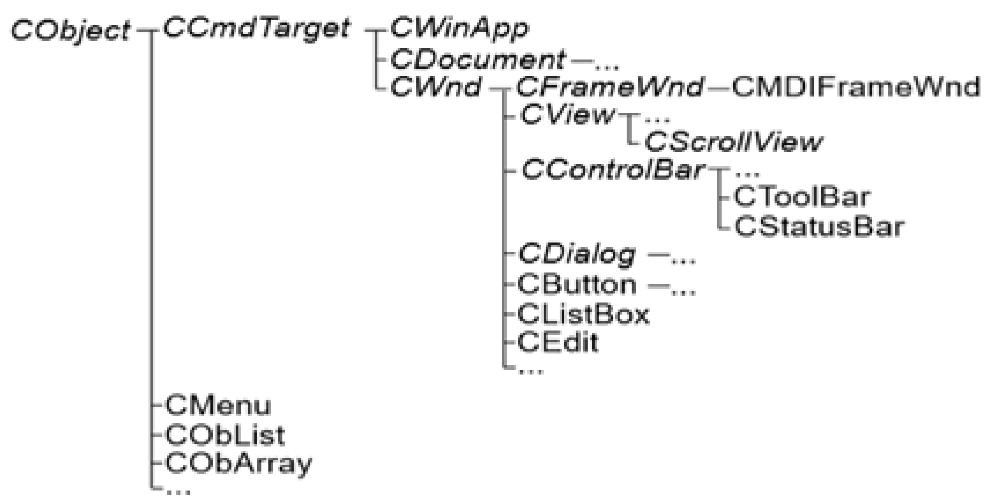

Figure 27 Core classes of the MFC framework

3.8.3 MFC Root Class (CObject):

In MFC Framework almost all classes derived from ***CObject*** abstract Class and inherits its functionality (behavior). And this ***CObject*** class defines/implement's the protocol for objects meta-information and object serialization.

This term Meta-information is contains information about objects and classes, C++ does not provide any information about any object as it's a class.

Due to this deficiency of C++ there are many libraries provide functionality to get meta-information about objects and classes. So like other libraries MFC also implements mechanism using Macro's to provide meta-information about objects and classes.

3.8.4 Data Structures in MFC

MFC Framework there are number of collections to implements different data structures like lists, arrays and dictionaries (they are able to store key value pairs sometimes called maps). We can use these data structures in any MFC application without any modification.

Classes Defining a Generic Windows Application

Generally, GUI applications are event driven. As users of event driven application can enter its commends via any input device such as keyboard, mouse etc. then an GUI application process coming input from user any input device. For example by clicking on exit button a GUI application should be closed so this closing function will process by a GUI application or user can click on title bar then he can move application by dragging mouse so this dragging function should be also implement in GUI application in such a way that user can drag this selected window.

In MFC CWinApp class object gathers all incoming events from any device and distribute/dispatches then various relative components of application. Every MFC application will have one CWinApp object for handling application events. As MFC windows application started WinMain function will called and sends messages **InitInstance** and **Run** to the particular CWinApp object. After invoking Run function start the event loop that loop will handle all incoming events and messages constantly. Below figure 28 shows the starting of the event loops in MFC application.

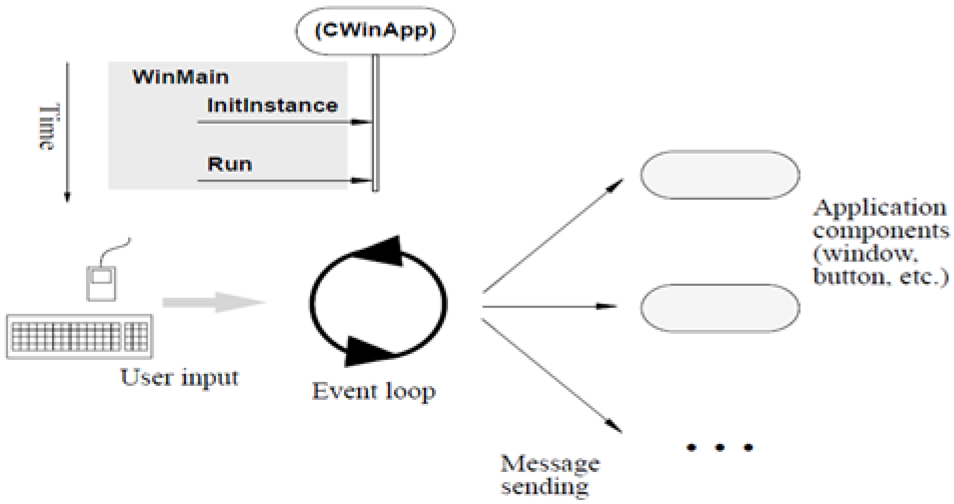

Figure 28: Starting the event loop in an MFC application

There are many GUI applications are document oriented applications these applications manages documents (view, edit and delete). MFC framework supports this document type applications directly. MFC document oriented application structure is architecture to MVC architecture.

In MFC CDocument and CView classes corresponds to MVC Model and View components. MFC also supports that CDocument can have several CView that is closely similar to concept of MVC. Where controller in MFC is will discuss in below section.

MFC high level class CWinApp manages all CDocument class objects. That is similar to controller component in MVC.

Figure 29 shows the structures of MFC class objects. In this figure there are two CDocument objects having their respective CView objects, in upper CDocument class has one CView class object that is responsible of edit and view its CDocument class object data/content. The content of other below CDocument class object is displayed and edit by two CView class objects. The main class CWinApp manages all CDocument objects. For example if user exit application then

CWinApp ask to all CDocument object that their changed contents should be saved or not before closing MFC application.

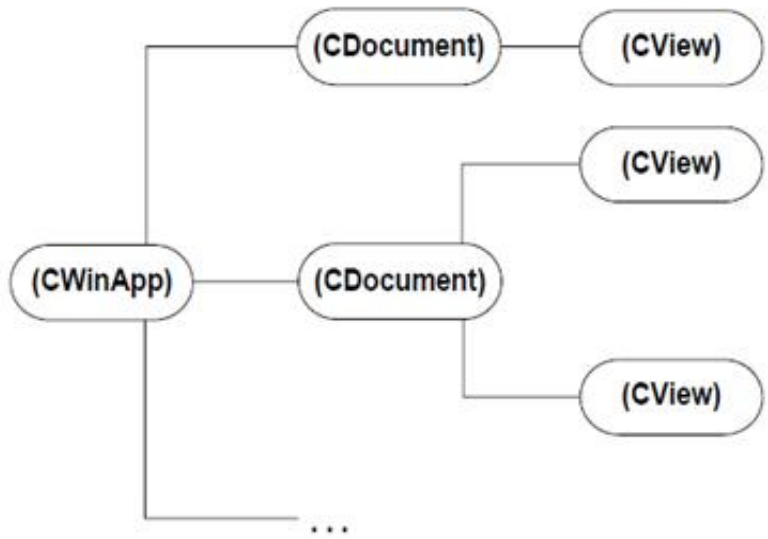

Figure 29: Principal object relationships in an MFC-based application

At the starting of MFC application during Application Wizard user can choose SDI (single document Interface) application that contain exactly one CDocument object and MDI (Multiple Document Interface) that manages arbitrary number of CDocument. MFC also support Dialog based application like Windows calculator this type of application also selectable from Application Wizard.

3.9 Experiment Data saving on Disk

After the experiment is performed the user is prompted with a message box that the experiment has been completed. Before that prompt message the developed framework takes all the data entered by the user as name, age, gender, profession and interests, takes all the recorded data of the Eye Gaze positions on the screen and create a file on the disk with name of the user as entered by the user and save it on the disk for further experiment use.

The generated file can be loaded into any other software and read into any other program for further experiment, a little Matlab program was written to show these values with user data on the Matlab plot as shown below where each green circle shows the gaze position of the user on the screen.

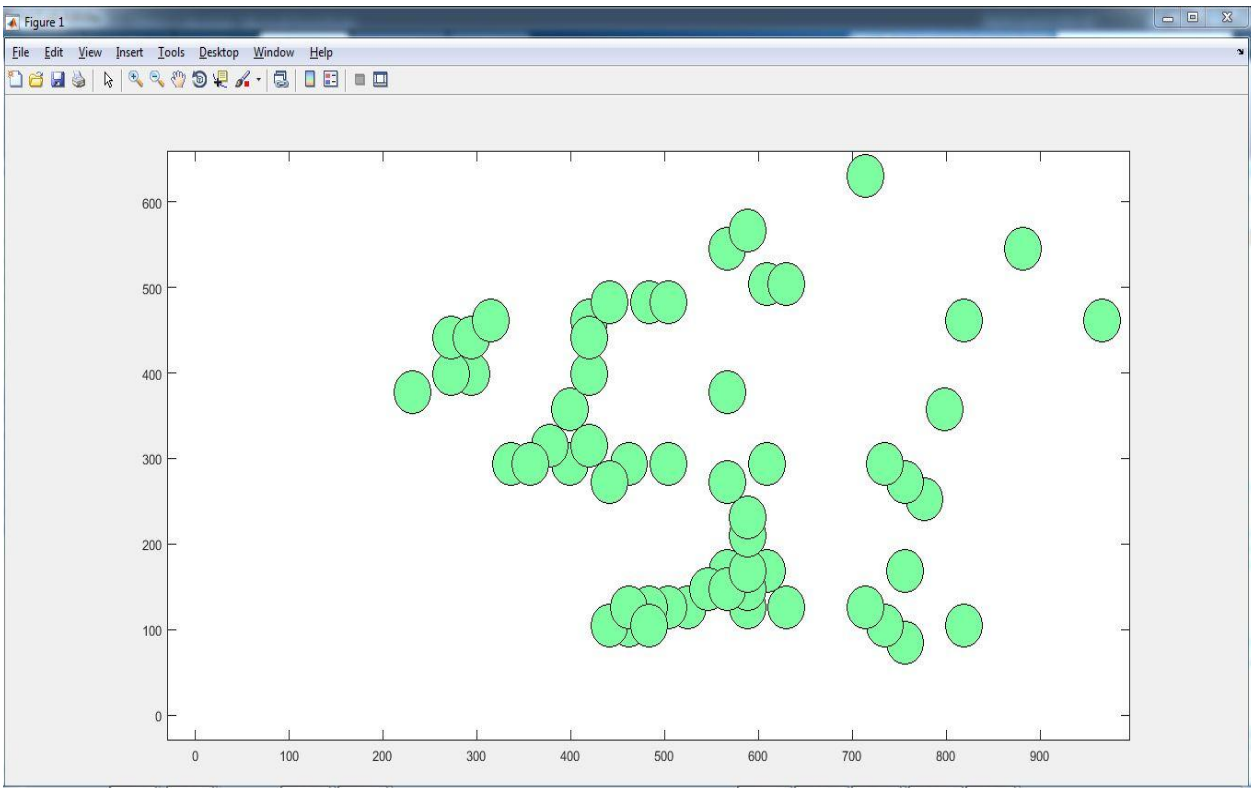

Figure 30: Gaze recordings of a user looking at screen

CHAPTER 4: RESULTS AND CONCLUSION

The program was tested with different users and results were found to be quite satisfactory. The results were taken with different light conditions, with user wearing glasses and having a small light source in the background with user with very less open eyes etc. The results were very satisfying as is shown below.

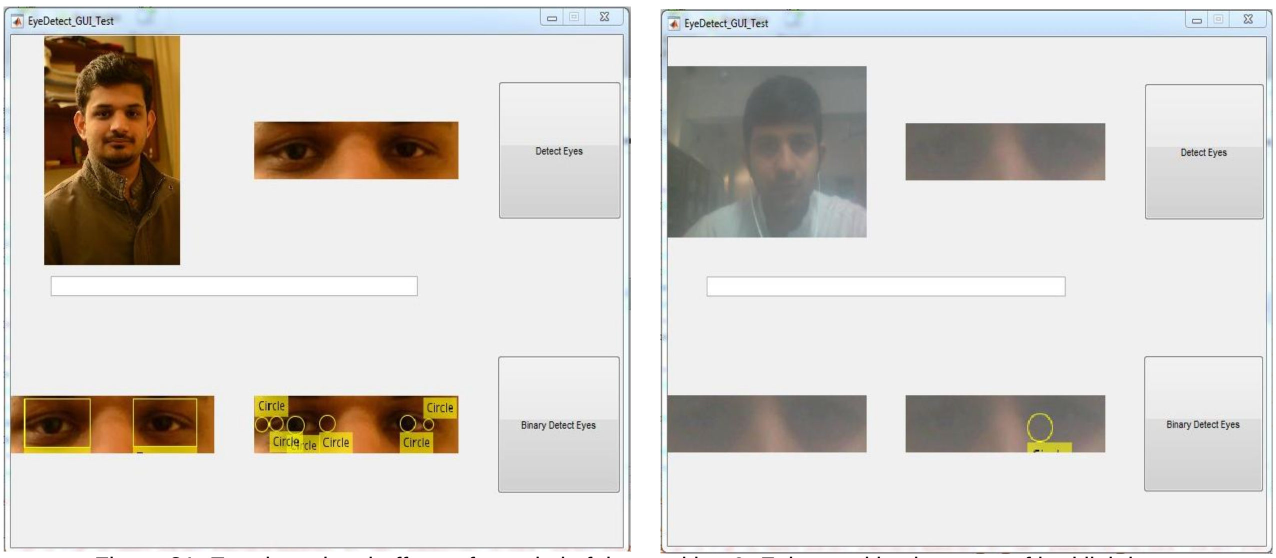

Figure 31: Eye detection, huff transform circle false positive & False positive because of bad lighting

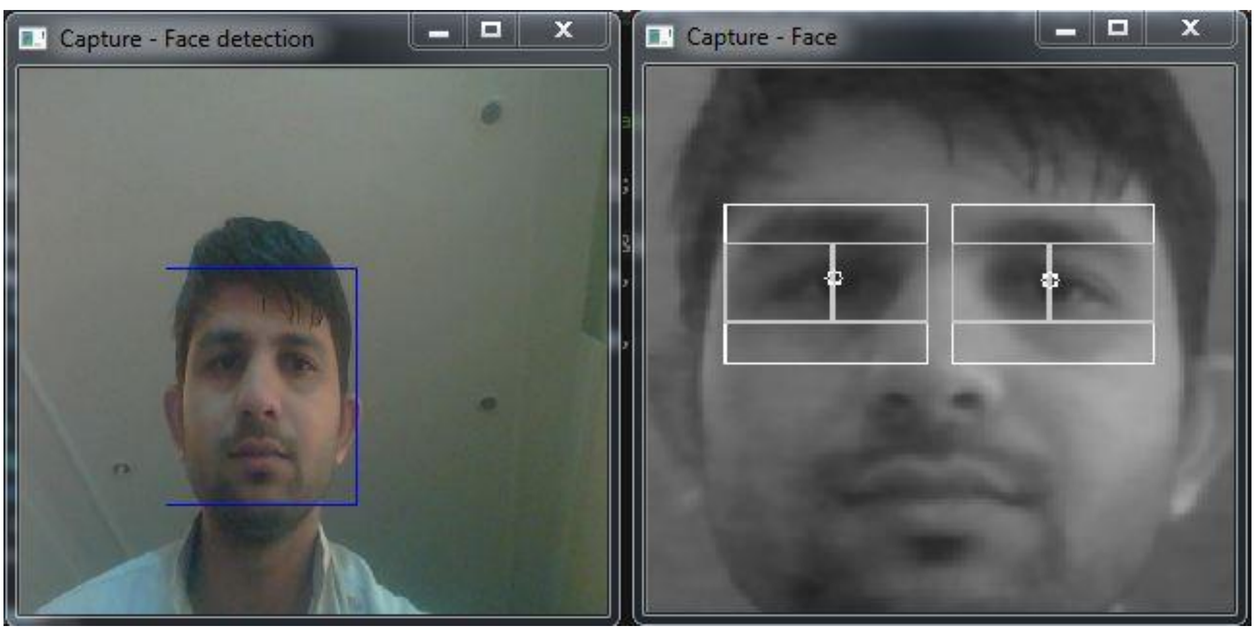

Figure 32: Face and eye detection using gradiant vector method

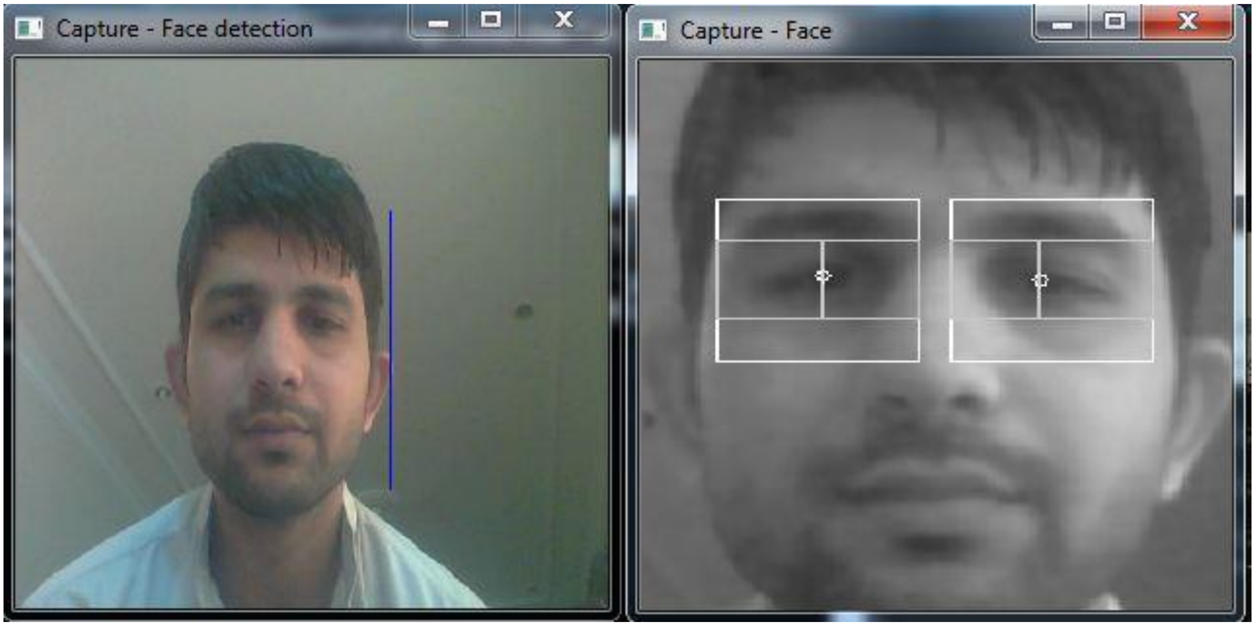

Figure 33: Results with lesser eyes open

Figure 34: Results with eyes with very less eyes open

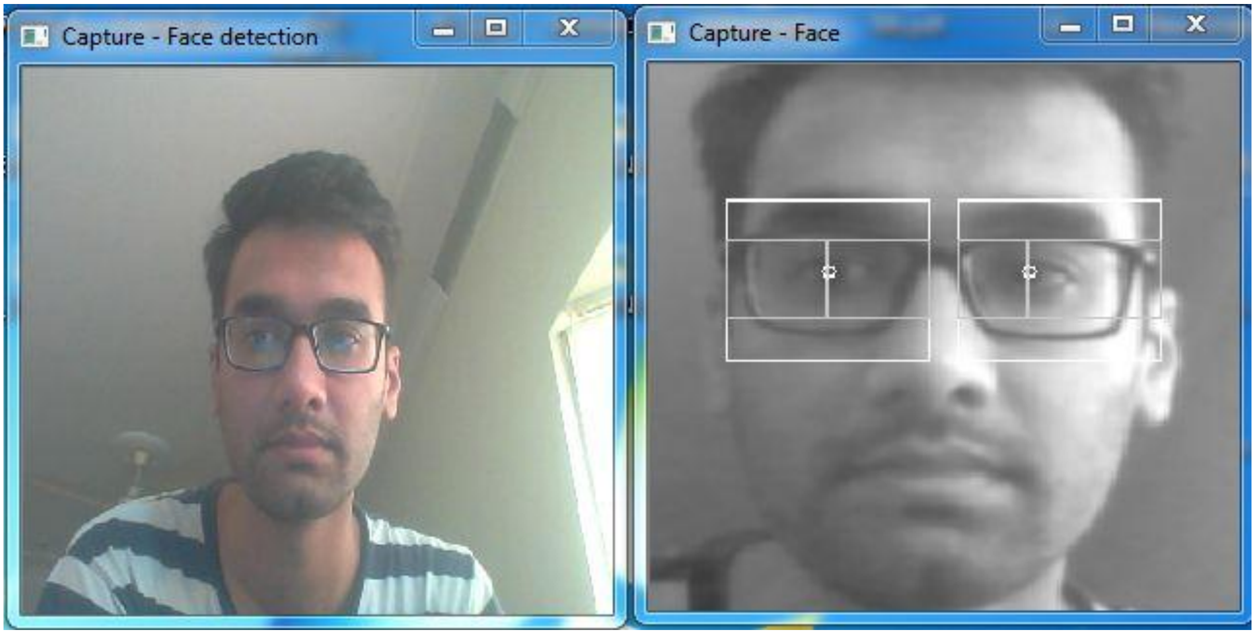

Figure 35: Results of proposed scheme with glasses

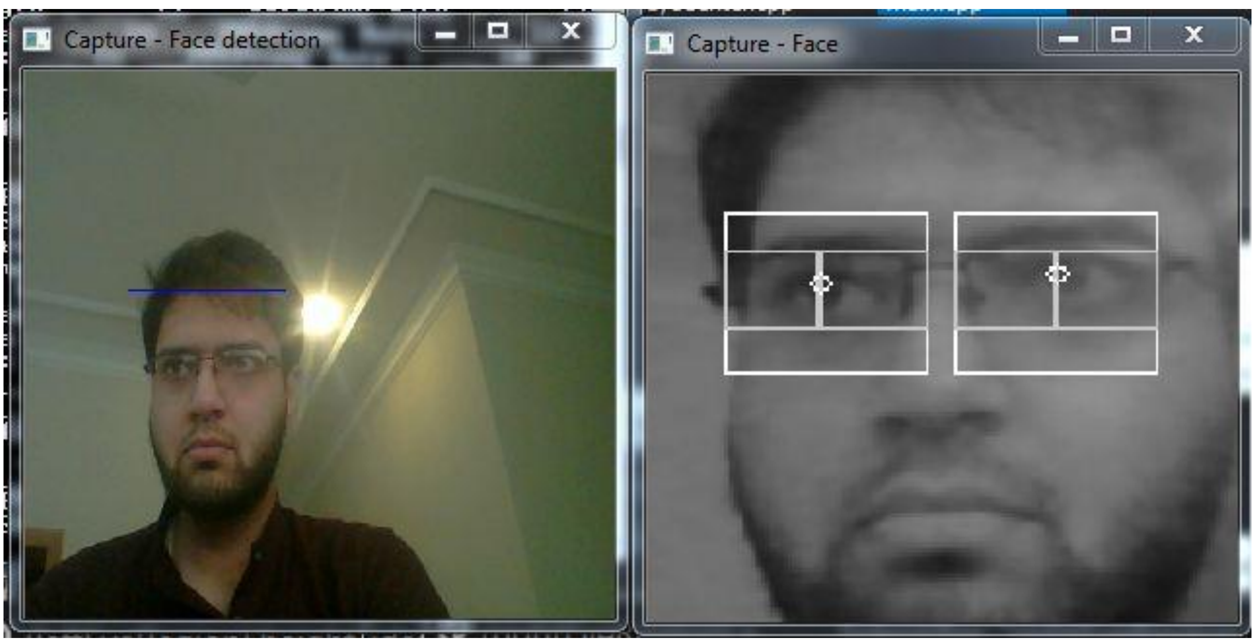

Figure 36: Results with glasses and light source in background

4.1 Conclusion

The proposed technique for eye gaze tracking gives satisfactory results as compare to other open source techniques available. It overcomes the deficiencies of the pupil identification where the user has opened his eyes partially and also overcomes the false positives which are generated by the eye brows and darker areas around the eyes. The proposed technique also works well while having light source in behind the user as is shown in the figures above. The proposed technique is also robust to different light conditions with 25 to 30 FPS throughput.

4.2 Future Work

The work can be improved by improving the eye calibrator part of the framework where a complex calibration technique can be used to better provide the calibration of the user eye recordings.

The work can be improved to get more FPS by improving the algorithm or by adding Dead Reckoning and/or interpolation techniques where the framework's eye tracking part is put into separate thread and returns values let say at 60 FPS or more and if a new value of gaze has not arrived it computes the gaze values by predicting the gaze position and later on correcting it along the experiment by using dead reckoning.

REFERENCES

[1] Andrew T Duchowski. (2002) A breadth-first survey of eye-tracking applications. Behavior Research Methods, Instruments, & Computers, 34(4):455–470. 27

[2] J. Doleˇzal and V. Fabian. (2015) 41. application of eye tracking in neuroscience. Clinical Neurophysiology, 126(3):e44

[3] Robert S Allison, Moshe Eizenman, and Bob SK Cheung (1996). Combined head and eye tracking system for dynamic testing of the vestibular system. IEEE Transactions on Biomedical Engineering, 43(11):1073–1082.

[4] Alexander C Volck, Roman D Laske, Ralph Litschel, Rudolf Probst, and Abel-Jan Tasman. (2015) Sound localization measured by eye-tracking. International journal of audiology, 54(12):976–983.

[5] W. Heide, E. Koenig, P. Trillenberg, D. K¨ompf, and D. S. Zee. (1999) Electrooculography: technical standards and applications. the international federation of clinical neurophysiology. Electroencephalography and clinical neurophysiology. Supplement, 52:223–240.

[6] JN Van der Geest and MA Frens. (2002). Recording eye movements with video-oculography and scleral search coils: a direct comparison of two methods. Journal of neuroscience methods, 114(2):185–195.

[7] W W Abbott and A A Faisal. (2012) Ultra-low-cost 3d gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces. Journal of Neural Engineering, 9(4):046016.

[8] A. I. Adiba, N. Tanaka, and J. Miyake. (2016) An adjustable gaze tracking system and its application for automatic discrimination of interest objects. IEEE/ASME Transactions on Mechatronics, 21(2):973–979, April 2016.

[9] Elizabeth S Kim, Adam Naples, Giuliana Vaccarino Gearty, Quan Wang, Seth Wallace, Carla Wall, Michael Perlmutter, Jennifer Kowitt, Linda Friedlaender, Brian Reichow, et al. (2014) Development of an untethered, mobile, low-cost head-mounted eye tracker. In Proceedings of the Symposium on Eye Tracking Research and Applications, pages 247–250. ACM.

[10] Dongheng Li, Jason Babcock, and Derrick J Parkhurst. openeyes: (2006) a low-cost head-mounted eye-tracking solution. In Proceedings of the 2006 symposium on Eye tracking research & applications, pages 95–100. ACM.

[11] Rados law Mantiuk, Micha l Kowalik, Adam Nowosielski, and Bartosz Bazyluk. (2012) Do-it-yourself eye tracker: Lowcost pupil-based eye tracker for computer graphics applications. In International Conference on Multimedia Modeling, pages 115–125. Springer.

[12] I Ketut Gede Darma Putra, Agung Cahyawan, and Yandi Perdana. (2013) Low-cost based eye tracking and eye gaze estimation. TELKOMNIKA (Telecommunication Computing Electronics and Control), 9(2):377–386.

[13] Nicolas Schneider, Peter Bex, Erhardt Barth, and Michael Dorr. (2011) An open-source low-cost eye-tracking system for portable real-time and offline tracking. In Proceedings of the 1st Conference on Novel gaze-controlled applications, page 8. ACM.

[14] Dongheng Li, D. Winfield, and D. J. Parkhurst. Starburst: (2005) A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In 2005

IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops, pages 79–79, June 2005.

[15] Lin Lin, Lin Pan, LiFang Wei, and Lun Yu. (2010) A robust and accurate detection of pupil images. In Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, volume 1, pages 70–74. IEEE.

[16] Xindian Long, Ozan K Tonguz, and Alex Kiderman. (2007) A high speed eye tracking system with robust pupil center estimation algorithm. In Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pages 3331–3334. IEEE.

[17] Lech Swirski, Andreas Bulling, and Neil A. Dodgson. (2012) Robust real-time pupil tracking in highly off-axis images. ´ In Proceedings of ETRA.

[18] Danjie Zhu, Steven T. Moore, and Theodore Raphan. (1999) Robust pupil center detection using a curvature algorithm. Computer Methods and Programs in Biomedicine, 59(3):145 – 157.

[19] Moritz Kassner, William Patera, and Andreas Bulling. (2014) Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp ’14 Adjunct, pages 1151–1160, New York, NY, USA. ACM.

[20] Wolfgang Fuhl, Thomas K¨ubler, Katrin Sippel, Wolfgang Rosenstiel, and Enkelejda Kasneci. (2015) ExCuSe: Robust Pupil Detection in Real-World Scenarios, pages 39–51. Springer International Publishing, Cham.

[21] Jan Zimmermann, Yuriria Vazquez, Paul W. Glimcher, Bijan Pesaran, and Kenway Louie. (2016) Oculomatic: High speed, reliable, and accurate open-source eye tracking for humans and non-human primates. Journal of Neuroscience Methods, 270:138 – 146, 2016.

[22] Wael Abd-Almageed, M Sami Fadali, and George Bebis. (2002) A non-intrusive kalman filter-based tracker for pursuit eye movement. In American Control Conference, 2002. Proceedings of the 2002, volume 2, pages 1443–1447. IEEE.

[23] Jian-nan Chi, Li-hua Xie, Peng-yun Zhang, Yi-fang Lu, and Guo-sheng Zhang. (2014) Hybrid particle and kalman filtering for pupil tracking in active ir illumination gaze tracking system. Mathematical Problems in Engineering.

[24] Jiashu Zhang and Zutao Zhang. (2006) Application of a strong tracking finite-difference extended kalman filter to eye tracking. In International Conference on Intelligent Computing, pages 1170–1179. Springer.

[25] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. (1997) Pfinder: Real-time tracking of the human body. IEEE Transactions on pattern analysis and machine intelligence, 19(7):780–785.

[26] F Timm, E Barth. (2011) Accurate Eye Centre Localisation by Means of Gradients. Visapp 11, 125-130